Exploring the Transferability of Supervised Pretraining with an MLP Projector

Yizhou Wang[1,3*†], Shixiang Tang[2†], Feng Zhu[3], Lei Bai[2‡], Rui Zhao[3,4], Donglian Qi[1], Wanli Ouyang[2]
[1]Zhejiang University, [2]The University of Sydney,[3]SenseTime Research,
[4]Qing Yuan Research Institute, Shanghai Jiao Tong University, Shanghai, China
{yizhouwang, qidl}@zju.edu.cn, stan3906@uni.sydney.edu.au, baisanshi@gmail.com,
{zhufeng, zhaorui}@sensetime.com, wanli.ouyang@sydney.edu.au

A. Visualization of Feature Mixtureness

We provide an intuitive understanding of the relation between Feature Mixtureness and the feature distribution distance by manually generating two sets of features with different distribution distance. We use red and blue to represent class centers from pre-D and eval-D, respectively. The visualization results are illustrated in Fig. 1. From (a) to (c), when the distribution distance between pre-D and eval-D increases, Feature Mixtureness decreases accordingly. When we fix the variance of features in pre-D and gradually enlarge the variance of features in eval-D (from (d) to (f)), Feature Mixtureness will decrease as well. Based on the observations above, we conclude that our Feature Mixtureness can empirically measure the feature distribution distance between pre-D and eval-D.

B. Visualization of Feature Distribution during Pretraining

In this section, we provide an illustration to establish an intuition about how intra-class variation and Feature Mixtureness evolve during different pretraining epochs.

B.1. Intra-class Variation on pre-D

We visualize the feature distribution using samples from 10 randomly selected classes in pre-D in Fig. 2 to illustrate the evaluation results of the intra-class variation on pre-D. Different colors represent different classes. In SL, the intra-class variation will continuously decrease to a small value with more training epochs. In contrast, the intra-class variance of SL-MLP and Byol retains even though we pretrain the networks at large pretraining epochs. This visualization graphically validates that the MLP projector can enlarge the intra-class variation of features in pre-D.

B.2. Feature Mixtureness between pre-D and eval-D

We randomly select features from 5 classes in pre-D and 5 classes in eval-D, and then visualize them by t-SNE in Fig. 3. Cold colors represent features from pre-D and warm colors represent features from eval-D. At the early pretraining stage, all methods show high Feature Mixtureness as they cannot well classify images in pre-D. When the training epoch is becoming larger, SL shows lower Feature Mixtureness, which indicates a larger feature distribution distance between pre-D and eval-D. Instead, SL-MLP and Byol remain large Feature Mixtureness when the training epoch is becoming larger, which shows that the feature distribution distance between pre-D and eval-D is not enlarged by Byol and SL-MLP.

C. Theoretical Analysis of Theory 1

C.1. Proof of Theory 1

Proof. Denote the pretrained feature extractor with the parameters θ as $f(\cdot;\theta)$. The softmax function is built upon the feature representation of the backbone $\mathbf{f}_i = f(\mathbf{x}_i;\theta) \in \mathbb{R}^D$, where $\mathbf{x}_i$ is an image, and D is the dimension of features. We compute

∗ The work was done during an internship at SenseTime.
† Equal Contribution.
‡ Corresponding author.

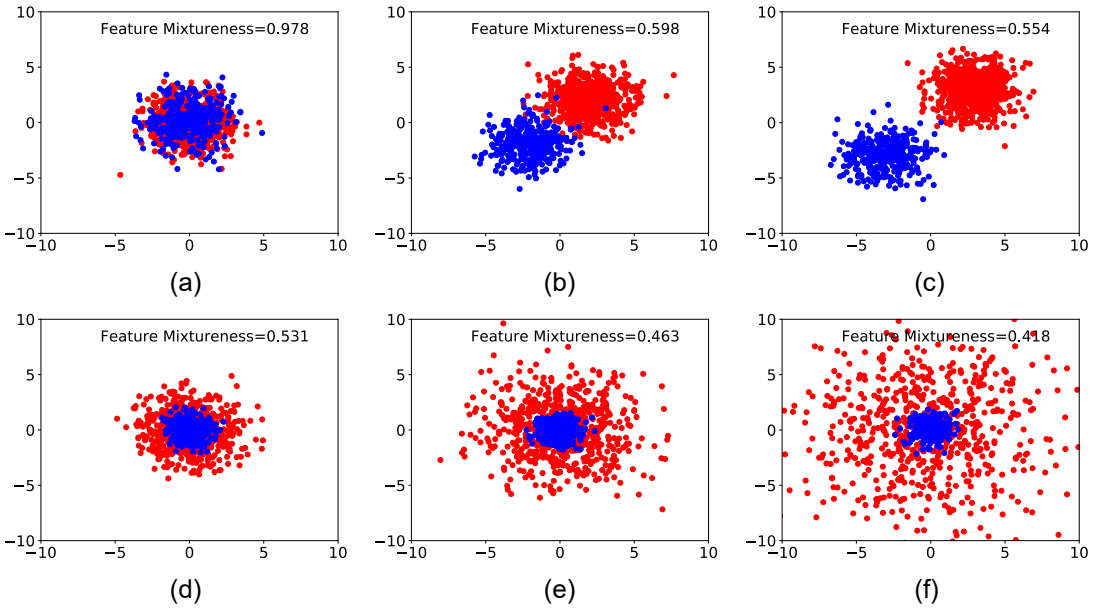


Figure 1. Visualization of Feature Mixtureness with different manually generated feature distribution. Red and blue represent pre-D and eval-D class centers, respectively.

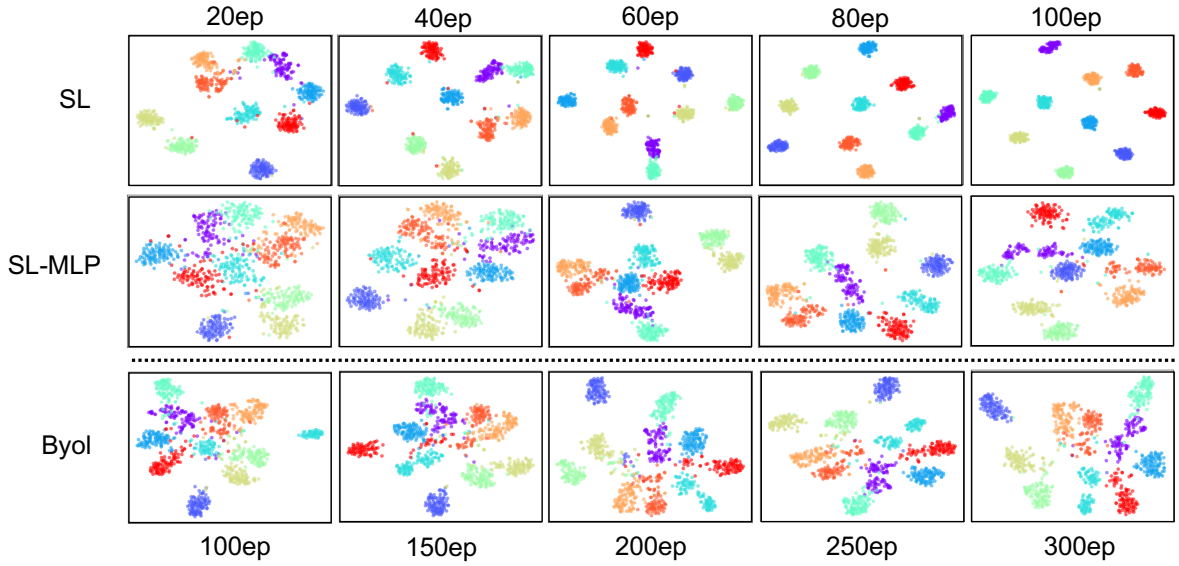


Figure 2. Evolution of intra-class variation of features in pre-D with different epochs. Different colors denote different classes. The intra-class variation of SL will be very small when the pretraining epoch is large enough. Instead, the intra-class variation of SL-MLP and Byol still retains even though the model is pretrained by large epochs.

the class center $\mu(I_j)$ for class j as the mean of the feature embeddings as

$$\mu(I_j) = \frac{1}{I_j} \sum_{(\mathbf{x}_i, y_i) \in I_j} \mathbf{f}_i, \tag{1}$$

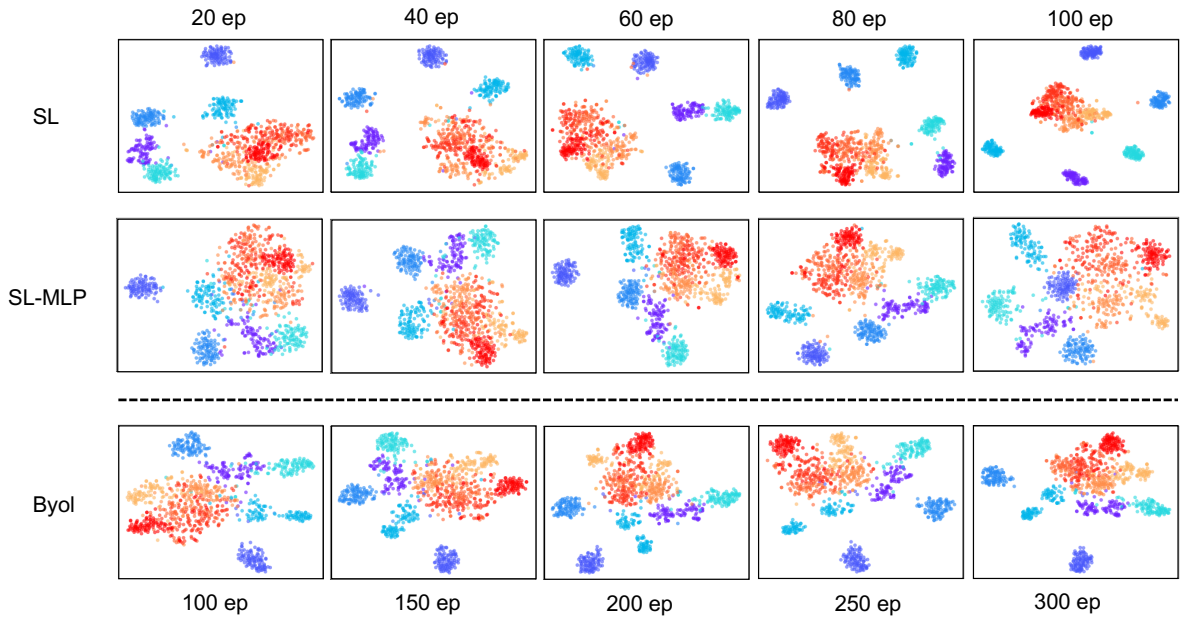


Figure 3. Evolution of Feature Mixtureness between features from pre-D and from eval-D. Cold colors denote features from 5 classes that are randomly selected from pre-D, and warm colors denote features from 5 classes that are randomly selected from eval-D. Feature Mixtureness of SL continuously decrease during pretraining. Alternatively, SL-MLP and Byol keeps a relatively high Feature Mixtureness at large pretraining epochs.

where I_j denotes the images in the j-th class. Then we define the inter-class distance $D_{inter}(I)$, and intra-class distance $D_{intra}(I)$ on a datatset with C classes as

$$D_{inter}(I) = \frac{1}{C(C-1)} \sum_{j=1}^{C} \sum_{k=1,k\neq j}^{C} ||\mu(I_j) - \mu(I_k)||^2, \tag{2}$$

$$D_{intra}(I) = \frac{1}{C} \sum_{j=1}^{C} (\frac{1}{|I_j|} \sum_{(\mathbf{x}_i, y_i)\in I_j} ||\mathbf{f}_i - \mu(I_j)||^2). \tag{3}$$

Substituting Eq. 1 into Eq. 2 and Eq. 3, we have

$$D_{inter}(I) = \frac{1}{C(C-1)} \sum_{j=1}^{C} \sum_{k=1,k\neq j}^{C} \left(\frac{1}{2|I_j||I_k|} \sum_{(\mathbf{x}_i, y_i)\in I_j} \sum_{(\mathbf{x}_l, y_l)\in I_k} ||\mathbf{f}_i - \mathbf{f}_l||^2 \right), \tag{4}$$

$$D_{intra}(I) = \frac{1}{C} \sum_{j=1}^{C} \left(\frac{1}{2|I_j|^2} \sum_{(\mathbf{x}_i, y_i)\in I_j} \sum_{(\mathbf{x}_l, y_l)\in I_j} ||\mathbf{f}_i - \mathbf{f}_l||^2 \right). \tag{5}$$

Taking expectation to Eq. 4 and Eq. 5, for any pair of data $(\mathbf{x}_i, y_i), (\mathbf{x}_l, y_l) \in I$, we have

$$\mathbb{E}(||\mathbf{f}_i - \mathbf{f}_l||^2) = \begin{cases} 2D_{intra}(I), y_i = y_l \\ 2D_{inter}(I), y_i \neq y_l \end{cases}. \tag{6}$$

For ease of analysis, we denote I^{pre}, I^{eval} as pre-D and eval-D, respectively. For any pair of data $(\mathbf{x}'_i, y'_i), (\mathbf{x}'_l, y'_l) \in I^{eval}$ in eval-D in the same class, *i.e.,* $y'_i = y'_l$, we have

$$\begin{aligned} D_{intra}(I^{eval}) &= \frac{1}{2}\mathbb{E}\left(||\mathbf{f}'_i - \mathbf{f}'_l||^2\right) \\ &= \frac{1}{2}\mathbb{E}\left[P(y_i = y_l)2D_{intra}(I^{pre}) + P(y_i \neq y_l)2D_{inter}(I^{pre})\right] \\ &= PD_{intra}(I^{pre}) + (1-P)D_{inter}(I^{pre}), \end{aligned} \tag{7}$$

where y_i is the label of an image $\mathbf{x}_i$ assigned by the classifier trained on pre-D, and $\mathbf{f}' = f(\mathbf{x}', \theta)$. Here, P represents the possibility that a pair of images in eval-D that belong to the same class is classified into the same classes in pre-D.

We denote $\psi(\phi^{-1}(I^{pre})) = D_{inter}(I^{eval})/D_{inter}(I^{pre})$ as the ratio of the model's inter-class distance on eval-D and the model's inter-class distance on pre-D. When the model is optimized on pre-D, its discriminative ratio on pre-D $\phi(I^{pre})$ becomes larger with the increase of $D_{inter}(I^{pre})$ and the decease of $D_{intra}(I^{pre})$. In most cases, $D_{inter}(I^{eval})/D_{inter}(I^{pre})$ is a monotonic decreasing function of $\phi(I^{pre})$, and is a monotonic increasing function of $\phi^{-1}(I^{pre})$, which has been empirically proven by [14]. Mathematically, it can be formulated as

$$\psi(\phi_2^{-1}(I^{pre})) > \psi(\phi_1^{-1}(I^{pre})), \text{ if } \phi_2^{-1}(I^{pre}) > \phi_1^{-1}(I^{pre}). \tag{8}$$

By substituting $D_{intra}(I^{eval}) = PD_{intra}(I^{pre}) + (1-P)D_{inter}(I^{pre})$ (Eq. 7) into the discriminative ratio inequality $\phi_2(I^{eval}) < \phi_1(I^{eval})$ given $\phi_2(I^{pre}) > \phi_1(I^{pre})$, we have

$$\phi_2(I^{eval}) < \phi_1(I^{eval}) \tag{9}$$

$$\iff \frac{D^2_{inter}(I^{eval})}{D^2_{intra}(I^{eval})} < \frac{D^1_{inter}(I^{eval})}{D^1_{intra}(I^{eval})} \tag{10}$$

$$\iff \frac{D^2_{inter}(I^{eval})}{PD^2_{intra}(I^{pre}) + (1-P)D^2_{inter}(I^{pre})} < \frac{D^1_{inter}(I^{eval})}{PD^1_{intra}(I^{pre}) + (1-P)D^1_{inter}(I^{pre})}, \tag{11}$$

$$\iff P < \frac{\frac{D^1_{inter}(I^{eval})}{D^1_{inter}(I^{pre})} - \frac{D^2_{inter}(I^{eval})}{D^2_{inter}(I^{pre})}}{\frac{D^1_{inter}(I^{eval})}{D^1_{inter}(I^{pre})} \cdot \left(1 - \frac{D^2_{intra}(I^{pre})}{D^2_{inter}(I^{pre})}\right) - \frac{D^2_{inter}(I^{eval})}{D^2_{inter}(I^{pre})} \cdot \left(1 - \frac{D^1_{intra}(I^{pre})}{D^1_{inter}(I^{pre})}\right)}, \tag{12}$$

$$\iff P < \frac{\psi(\phi_1^{-1}(I^{pre})) - \psi(\phi_2^{-1}(I^{pre}))}{\psi(\phi_1^{-1}(I^{pre}))\left(1 - \phi_2^{-1}(I^{pre})\right) - \psi(\phi_2^{-1}(I^{pre}))\left(1 - \phi_1^{-1}(I^{pre})\right)}, \tag{13}$$

$$\iff P < \frac{1}{1 - \phi_1^{-1}(I^{pre}) + \frac{\phi_2^{-1}(I^{pre}) - \phi_1^{-1}(I^{pre})}{\psi(\phi_2^{-1}(I^{pre})) - \psi(\phi_1^{-1}(I^{pre}))}\psi(\phi_1^{-1}(I^{pre}))}, \tag{14}$$

$$\iff P < \frac{1}{1 - \phi_1^{-1}(I^{pre}) + r\psi(\phi_1^{-1}(I^{pre}))}, \tag{15}$$

$$\iff r\psi(\phi_1^{-1}(I^{pre})) - \phi_1^{-1}(I^{pre}) < P^{-1} - 1, \tag{16}$$

$$\iff \frac{d\phi_1^{-1}(I^{pre})}{d\psi(\phi_1^{-1}(I^{pre}))}\psi(\phi_1^{-1}(I^{pre})) - \phi_1^{-1}(I^{pre}) < P^{-1} - 1, \tag{17}$$

$$\iff \frac{d\phi^{-1}(I^{pre})}{P^{-1} - 1 + \phi^{-1}(I^{pre})} < \frac{1}{\psi(\phi^{-1}(I^{pre}))} d\psi(\phi^{-1}(I^{pre})), \tag{18}$$

where

$$r = \frac{\phi_2^{-1}(I^{pre}) - \phi_1^{-1}(I^{pre})}{\psi(\phi_2^{-1}(I^{pre})) - \psi(\phi_1^{-1}(I^{pre}))} \tag{19}$$

$$\approx \frac{d\phi^{-1}(I^{pre})}{d\psi(\phi^{-1}(I^{pre}))}, \text{ when } \phi_2^{-1}(I^{pre}) - \phi_1^{-1}(I^{pre}) \to 0. \tag{20}$$

We take integration of Eq. 18 as

$$\iff \int_0^{\phi^{-1}(I^{pre})} \frac{d\phi^{-1}(I^{pre})}{P^{-1}-1+\phi^{-1}(I^{pre})} < \int_{\psi(0)}^{\psi(\phi^{-1}(I^{pre}))} \frac{1}{\psi(\phi^{-1}(I^{pre}))} d\psi(\phi^{-1}(I^{pre})), \tag{21}$$

$$\iff \ln\left[\phi^{-1}(I^{pre})+P^{-1}-1\right] < \ln\left[\psi(\phi^{-1}(I^{pre}))\right] + \ln\left(\frac{P^{-1}-1}{\psi(0)}\right), \tag{22}$$

$$\iff \phi^{-1}(I^{pre})+P^{-1}-1 < \psi(\phi^{-1}(I^{pre}))\frac{P^{-1}-1}{\psi(0)}, \tag{23}$$

$$\iff \phi^{-1}(I^{pre}) < 1-P^{-1}+\psi(\phi^{-1}(I^{pre}))\frac{P^{-1}-1}{\psi(0)}, \tag{24}$$

$$\iff \phi^{-1}(I^{pre}) < (\frac{\psi(\phi^{-1}(I^{pre}))}{\psi(0)}-1)(P^{-1}-1) \tag{25}$$

$$\iff \phi(I^{pre}) > t \tag{26}$$

where the threshold t is defined as

$$t = \left[(\frac{\psi(\phi^{-1}(I^{pre}))}{\psi(0)}-1)(P^{-1}-1)\right]^{-1}. \tag{27}$$

According to Formulation 8, $\psi(\phi^{-1}(I^{pre})) > \psi(0)$ because $\phi^{-1}(I^{pre}) > 0$. Therefore, $\frac{\psi(\phi^{-1}(I^{pre}))}{\psi(0)} - 1 > 0$, which means that increasing P will lead to increasing the threshold t. □

C.2. Analysis of P

In the following, we explain how P in Equation 7 can be theoretically computed, and how P negatively relates to the feature distribution distance briefly.

C.2.1 Computational Method of P

Given a fixed backbone pretrained $f(\cdot;\theta)$ on pre-D, we denote the classifier trained by pre-D as $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_{C^{pre}})$. The possibility of an image $\mathbf{x}$ of the class j in eval-D classified by the classifier $\mathbf{W}$ into the class k in pre-D can be defined as

$$P_{jk} = \frac{1}{|I_j^{eval}|} \sum_{(\mathbf{x}_i, y_i)\in I_j^{eval}} \frac{\exp(\mathbf{w}_k \cdot f(\mathbf{x};\theta))}{\sum_{k'=1}^{C^{pre}} \exp(\mathbf{w}_{k'} \cdot f(\mathbf{x};\theta))}, \tag{28}$$

where $|I_j^{eval}|$ denotes the number of images in the j-th class in eval-D. Then the probability of a pair of samples in the same class j in eval-D classified into the same class in eval-D is

$$P_j = \sum_{k=1}^{C^{pre}} P_{jk}^2. \tag{29}$$

The average probability of P_j is

$$P = \frac{1}{C^{eval}} \sum_{j=1}^{C^{eval}} P_j. \tag{30}$$

C.2.2 P is Negatively Related to the Feature Distribution Distance

In this part, we only use two extreme cases to briefly analyze the relation between P and the feature distribution distance.

Specifically, we first deduce the upper bound and the lower bound of P. We find that the upper bound is reached when the feature distribution distance between pre-D and eval-D is extremely small, and the lower bound is reached when the feature distribution distance between pre-D and eval-D is extremely large, which indicates P is negatively related to the feature distribution distance.

For the upper bound of P,

$$P = \frac{1}{C^{eval}} \sum_{j=1}^{C^{eval}} P_j \tag{31}$$

$$= \frac{1}{C^{eval}} \sum_{j=1}^{C^{eval}} \sum_{k=1}^{C^{pre}} P_{jk}^2 \tag{32}$$

$$\leq \frac{1}{C^{eval}} \sum_{j=1}^{C^{eval}} \left(\sum_{k=1}^{C^{pre}} P_{jk} \right)^2 \tag{33}$$

$$= \frac{1}{C^{eval}} \sum_{j=1}^{C^{eval}} 1 \tag{34}$$

$$= 1, \tag{35}$$

where Inequality 33 is derived by Cauchy Schwarz Inequality [27], and if and only if $P_{jk} = 1$ and $P_{jk'} = 0$ for $\forall k' \neq k$, P reaches its upper bound 1.

For the lower bound of P,

$$P = \frac{1}{C^{eval}} \sum_{j=1}^{C^{eval}} P_j \tag{36}$$

$$= \frac{1}{C^{eval}} \sum_{j=1}^{C^{eval}} \sum_{k=1}^{C^{pre}} P_{jk}^2 \tag{37}$$

$$\geq \frac{1}{C^{eval}} \sum_{j=1}^{C^{eval}} \frac{1}{C^{pre}} \left(\sum_{k=1}^{C^{pre}} P_{jk} \right)^2 \tag{38}$$

$$= \frac{1}{C^{eval}} \sum_{j=1}^{C^{eval}} \frac{1}{C^{pre}} \tag{39}$$

$$= \frac{1}{C^{pre}}, \tag{40}$$

where Inequality 38 is derived by Fundamental Inequality [2], and if and only if $P_{jk} = \frac{1}{C^{pre}}$ for $\forall k \in [1, C^{pre}]$, P reaches its lower bound $\frac{1}{C^{pre}}$.

Analysis on Small Feature Distribution Distance between pre-D and eval-D. When pre-D and eval-D have small feature distribution distance, a pair of two images $(\mathbf{x}_m, y'_m)$ and $(\mathbf{x}_n, y'_n)$ belong to the same class j in eval-D, *i.e.,* $y'_m = y'_n$ will be classified to the same class k in pre-D when classified by $\mathbf{W}$ with high confidence. That is, only P_{jk} will have high confidence close to 1 and $P_{jk'}, \forall k' \neq k$ will be close to 0, which is similar to the condition when P reaches its upper bound.

Analysis on Large Feature Distribution Distance between pre-D and eval-D. When pre-D and eval-D have large feature distribution distance, a pair of two images $(\mathbf{x}_m, y'_m)$ and $(\mathbf{x}_n, y'_n)$ belong to the same class in eval-D, *i.e.,* $y'_m = y'_n$ will be randomly classified to the classes in pre-D using $\mathbf{W}$. Mathematically, $P_{jk} \approx \frac{1}{C^{pre}}$, which is similar to the condition when P reaches its lower bound.

Based on the analysis above, we can conclude that P is negatively related to feature distribution distance, and larger P often means less feature distribution distance.

D. MLP components

In this section, we provide the detailed analysis about how each component of the MLP projector influences the intra-class variation (represented by discriminative ratio ϕ^{pre}) on pre-D, Feature Mixtureness Π between pre-D and eval-D, and feature redundancy $\mathcal{R}$. Based on SL which does not include MLP, we ablate the structure of the MLP projector by adding the input fully connected layer, the output fully connected layer, the batch normalization layer and the ReLU layer incrementally.

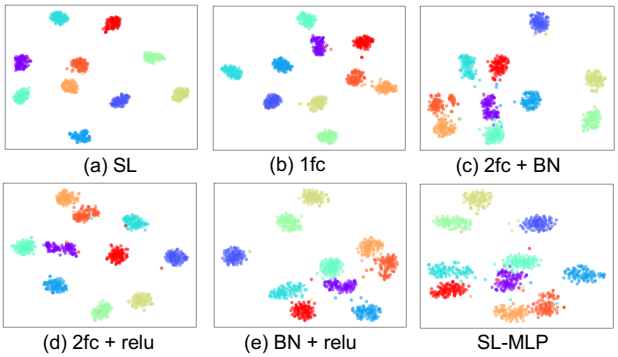


Figure 4. Visualization of intra-class variation by different components. We randomly select 10 classes in pre-D. Different colors denote different classes. Comparing (a) wth (b), we can see the fully-connected layer can slightly help enlarge the intra-class variation. Comparing (a-b) and (d-e), we can observe the batch normalization layer and the ReLU layer can significantly enlarge the intra-class variation in the feature space. In general, all components in the MLP layer is beneficial to enlarge intra-class variation, which proves their effectiveness in enhancing transferaiblity of pretraining models.

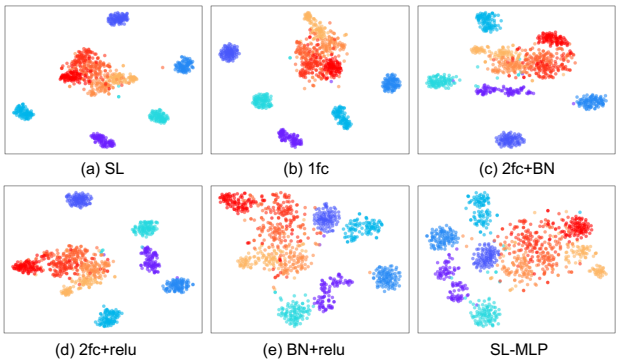


Figure 5. Visualization of Feature Mixtureness of features pretrained by different MLP components. Different colors denote different classes. Points with cold colors denote the features from pre-D, and points with warm colors denote the features from eval-D. Comparing (c-d) with (a-b), we can see that adding BN and ReLU can increase Feature Mixtureness between pre-D and eval-D. Comparing (e) with (a-d), we can conclude that BN and ReLU play the main roles in the MLP projector as (e) shows larger Feature Mixtureness. An MLP projector with all components achieves the largest Feature Mixtureness.

The input fully connected layer and the output fully connected layer are both set to have hidden units of 2048 and output dimensions of 2048 to keep same output feature dimensions as SL. All experiments are pretrained over 100 epochs. Testing results of the discriminative ratio on pre-D, Feature Mixtureness Π and feature redundancy $\mathcal{R}$ are illustrated in Tab. 1.

D.1. Visualization of intra-class variation

We randomly select features from 10 classes in pre-D and visualize their intra-class variation in Fig. 4. Different colors denote features from different classes. We specify the components in the MLP projector below each visualization image. Comparing (a) with (b), we can see that adding a fully connected layer can slightly enlarge intra-class variation, which indicates that linear transformation helps transferability marginally. Instead, comparing (a-b) with (c-e), we can observe that the batch normalization layer and the ReLU layer are important components in the MLP projector, which can significantly enlarge the intra-class variation in the feature space of pre-D. In general, comparing SL-MLP with (a-e), we can conclude that all components in MLP projector help enlarge the intra-class variation of features in pre-D while the batch normalization layer and the ReLU layer play the most important roles.

D.2. Visualization of Feature Mixtureness

We randomly select features from 5 classes in pre-D and 5 classes in eval-D to visualize Feature Mixtureness with different MLP components. The results are summarized in Fig. 5. The features with cold colors come from pre-D, the features with warm colors come from eval-D. Comparing (a) and (b), we can see adding a fully connected layer can hardly increase Feature Mixtureness between pre-D and eval-D. Comparing (c-d) with (b), we can conclude that the batch normalization layer and the ReLU layer can increase Feature Mixtureness between pre-D and eval-D. Comparing (b-d) with (e), we can summarize that the batch normalization and the ReLU layer are the most important components. A batch normalization layer with a ReLU layer can significantly increase Feature Mixtureness between pre-D and eval-D, which has already been similar to Feature Mixtureness when the MLP projector has the complete architectural.

D.3. Quantitative Analyse of MLP components

With the discriminative ratio ϕ^{pre}, Feature Mixtureness Π and feature redundancy $\mathcal{R}$ defined in main text Sec. 4.2, we quantitatively examine the effect of different components in the MLP projector. The results are presented in Tab. 1. Firstly, the fully connected layer has little influence on three metrics. Comparing (a) and (b), when adding a fully connected layer, the model shows slight improvement on Feature Mixtureness and feature redundancy, and slight decrease of the discriminative ratio on pre-D. Second, non-linear layer brings considerable improvements. Comparing (b) to (d), we can summarize that

Exp	Components				Top-1	$D^{pre}_{inter}/D^{pre}_{intra}$	$\Pi(\uparrow)$	$\mathcal{R}(\downarrow)$
	Input FC	BN	ReLU	Output FC				
(a)					55.9	2.034	0.515	0.0776
(b)	✓				56.6	1.505	0.679	0.0671
(c)	✓	✓		✓	61.0	1.269	0.870	0.0369
(d)	✓		✓	✓	60.1	1.362	0.804	0.0654
(e)		✓	✓		60.5	1.045	0.846	0.0369
SL-MLP	✓	✓	✓	✓	62.5	1.124	0.871	0.0351

Table 1. Quantitative analysis of structural design of inserted MLP, including discriminative ratio on pre-D, Feature Mixtureness Π and feature redundancy $\mathcal{R}$. (b-e) denote experiments in which different components are added on the SL baseline (a). When incrementally adding components of the MLP into SL, the distriminative ratio on pre-D and feature redundancy will decrease while the Feature Mixtureness will increase.

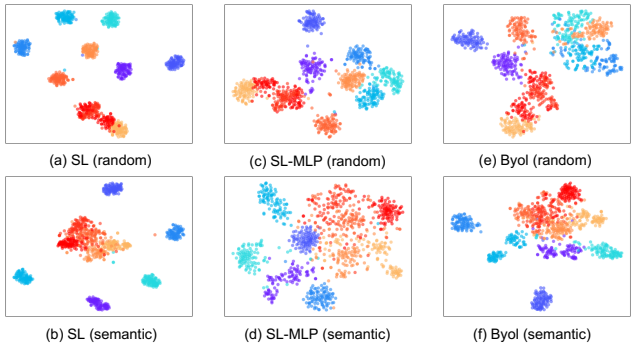


Figure 6. Visualization of Feature Mixtureness between pretraining dataset (pre-D) and evaluation dataset (eval-D). Different colors denote different classes. Classes in pre-D are denoted by cold colors, and classes in eval-D are denoted by warm colors. Comparing (a,c,e) and (b,d,f), we can conclude that large semantic gap between pre-D and eval-D will lead to small Feature Mixtureness between pre-D and eval-D. Comparing (b) and (d-f), we can observe that the MLP projector can increase Feature Mixtureness between pre-D and eval-D, and can bridge the semantic gap between pre-D and eval-D.

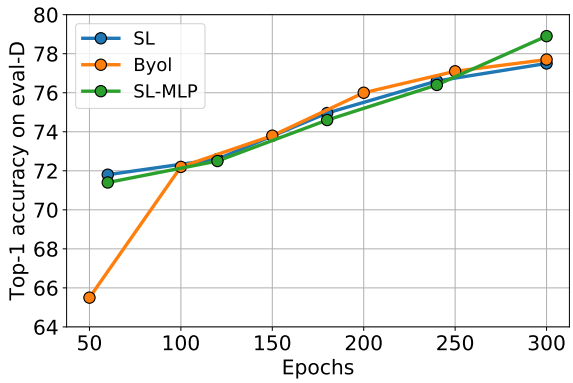


Figure 7. Linear evaluation accuracy on eval-D with small semantic gap. Following [7, 8], we pretrain SL, SL-MLP, and Byol on randomly chosen pre-D for 300 epochs. Compare the transfer performance on 300 epochs, SL shows a comparable transferability with Byol and SL-MLP when the semantic gap between pre-D and eval-D is small. In addition, unlike what we observe in main text Fig. 6(b), no performance drop during the last epochs appear. SL-MLP has a similar performance with SL from 60 to 240 epochs while has a consistently better performance on 300 epochs.

incrementally adding a ReLU, a batch normalization layer can increase Feature Mixtureness, reduce discriminative ratio, which could improve transferability of the pretrained model. Specifically, the ReLU layer brings a little improvement on feature redundancy. Comparing (a,b) with (c,e), we can conclude that BN not only reduces the discriminative ratio on pre-D, but also increases Feature Mixtureness. BN has a significant influence on future redundancy, which reduces feature redundancy by 50% (from 0.0671 to 0.0369). Last but not least, the combination of all components achieves the best transferability with the lowest feature redundancy, the highest Feature Mixtureness and a relatively large intra-class variation.

E. Concept Generalization Task with Small Semantic Gap

Following [22], to investigate how semantic difference between pre-D and eval-D can influence the transfer results on *Concept generalization task*, we randomly choose 652 classes as pre-D and 348 classes as eval-D from ImageNet-1K to establish a benchmark where pre-D and eval-D have small semantic gap. We denote the setting where pre-D and eval-D are constructed as Sec. 3.1 as large semantic gap setting (dubbed as *semantic*), and denote the setting where pre-D and eval-D are randomly selected as small semantic gap setting (dubbed as *random*).

epoch	cos	cos-mlp
20	**47.1**	45.0
40	47.8	**49.6**
60	50.9	**52.6**
80	53.5	**56.5**
100	53.7	**59.0**

Table 2. Top-1 linear evaluation accuracy on eval-D when pretraining the model on pre-D by cosine-softmax cross-entropy loss.

E.1. Visualization of Feature Mixtureness

We visualize features from pre-D and eval-D in small semantic gap setting and large semantic gap setting in Fig. 6. Specifically, SL (random), SL-MLP (random), and Byol (random) denote feature visualization of SL, SL-MLP, and Byol pretraining on the benchmark where pre-D and eval-D are randomly chosen. SL (semantic), SL-MLP (semantic), and Byol (semantic) denote feature visualization of SL, SL-MLP, and Byol pretraining on the benchmark where pre-D and eval-D are split according to semantic difference in WordNet, which is the same as main text Sec. 4.2. Our findings are two-fold. First, comparing with (a), (c), (e), pre-D features in (b), (d), (f) have large Feature Mixtureness, which indicates semantic difference influences the feature distribution distance between pre-D and eval-D in the feature space. Second, comparing (b) with (d), we find that Feature Mixtureness between pre-D and eval-D is enlarged by adding an MLP projector, which indicates that the MLP projector can significantly mitigate the feature distribution distance between pre-D and eval-D.

E.2. Quantitative Results

We first pretrain all the models on pre-D over 300 epochs, then examine linear evaluation results on eval-D. Our findings are three-fold. Firstly, compare the top-1 accuracy on 300 epochs, SL shows a comparable transferability with Byol and SL-MLP when the semantic gap between pre-D and eval-D is small. Second, unlike what we observe in main text Fig. 6(b), no performance drop during the last epochs appears, which indicates that the intra-class variation of SL is not above the threshold (defined in main text Sec. 4.3) when pre-D and eval-D have a small semantic gap. Third, SL-MLP has a similar performance with SL from 60 to 240 epochs while has a consistently *better* performance on 300 epochs, which verifies the effectiveness of the added MLP projector.

F. Replacing Softmax with Cosine-Softmax

In order to prove that our findings can be compatible with different loss functions, we replace the softmax cross-entropy loss with the cosine-softmax cross-entropy loss in the pretraining stage. Specifically, the cosine-softmax cross-entropy loss is defined as

$$\mathcal{L}_{\text{cos}}(\mathbf{x}_i, y_i) = -\log \frac{\exp(\beta \cdot \cos(\mathbf{w}_{y_i}, f(\mathbf{x}_i)))}{\sum_{i=j}^{C} \exp(\beta \cdot \cos(\mathbf{w}_j, f(\mathbf{x}_i)))}, \tag{41}$$

where $\mathbf{w}_i$ is the i-th class prototype, β is the scale factor. Accordingly, we add an MLP projector before the classifier to construct cosine-softmax-mlp cross-entropy loss, *i.e.,*

$$\mathcal{L}_{\text{cos-mlp}}(\mathbf{x}_i, y_i) = -\log \frac{\exp(\beta \cdot \cos(\mathbf{w}_{y_i}, g(f(\mathbf{x}_i))))}{\sum_{i=j}^{C} \exp(\beta \cdot \cos(\mathbf{w}_j, g(f(\mathbf{x}_i))))}, \tag{42}$$

where $\mathbf{w}_i$ is the i-th class prototype, $\beta = 30$ is the scale factor. We train for 100 epochs with a warm-up of 10 epochs and cosine decay learning schedule using the SGD optimizer. The base learning rate is set to 0.4. Weight decay of 10^{-4} is applied during pretraining. We report the top-1 accuracy on eval-D in Tab. 2. The results illustrate that when the model pretrained by cosine-softmax cross-entropy loss, adding an MLP projector can also facilitate transferability of supervised pretraining methods.

G. Visualize Convolution Channels by Optimization

According to [29] and [1], transfer performance is largely unaffected by the high-level semantic content of the pretraining data. To investigate that whether adding an MLP projector can influence what the convolution channels can learn. By using

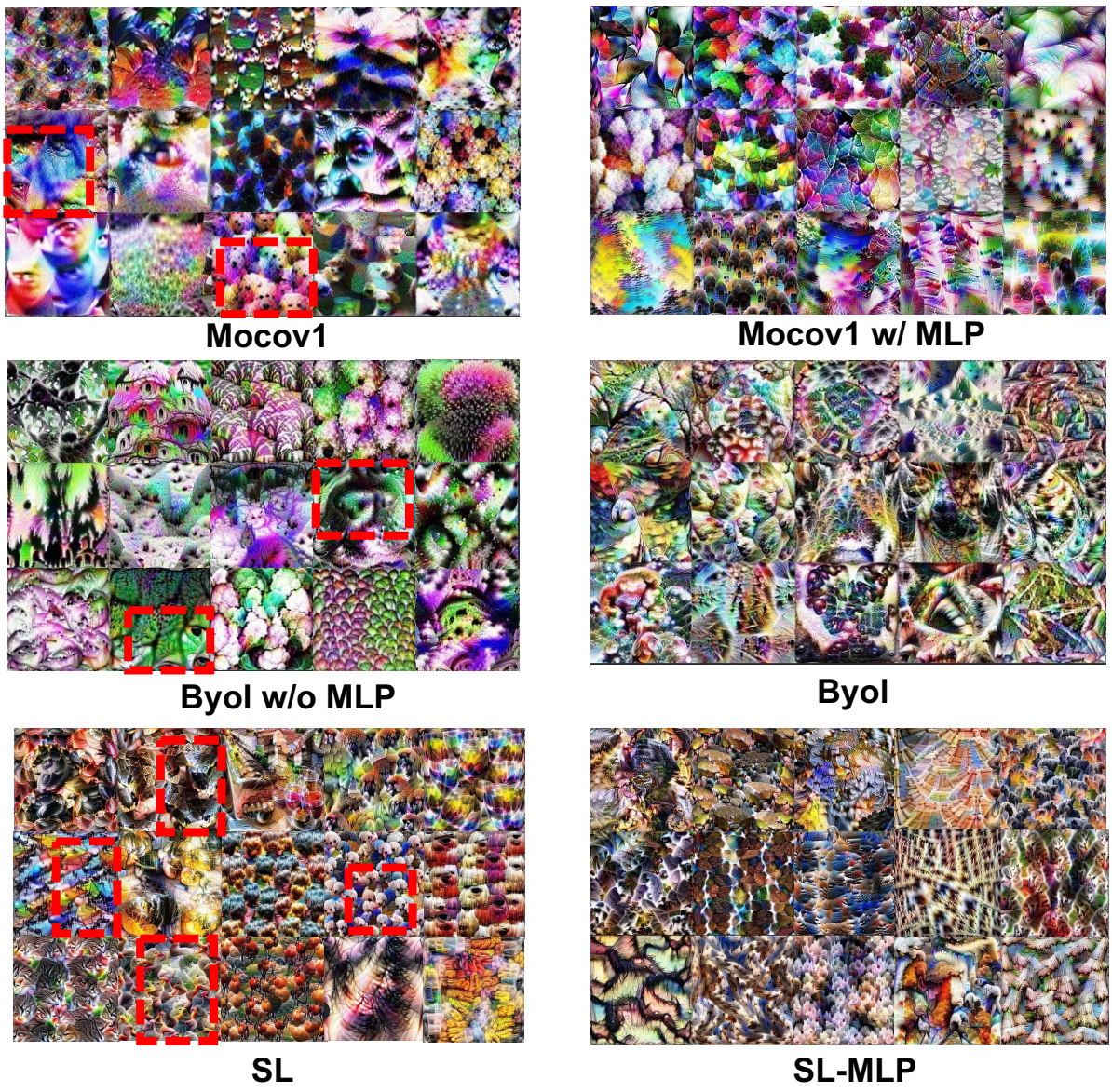


Figure 8. Convolution channels visualization of Mocov1, Mocov1 w/ MLP, Byol w/o MLP, Byol, SL and SL-MLP. Following the method proposed in [19], we visualize the maximum response of convolution channels in layer 4 of ResNet50 pretrained with different methods.

the method proposed in [19], we visualize the maximum response of convolution channels in layer 4 of ResNet50 (seen in main text Fig. 1) pretrained with methods without-MLP, *i.e.* SL, Mocov1, and Byol w/o MLP, and methods with-MLP, *i.e.* SL-MLP, Mocov1 w/ MLP, and Byol. Specifically, given a backbone with fixed parameters θ as $f(\cdot;\theta)$, we denote the parameters before the convolution channel j as $f(\cdot;\theta_j)$, we optimize the most representative sample $\mathbf{x}_i$ of the convolution channel j by maximizing the output logits $f(\mathbf{x};\theta_j)$, *i.e.*, $\mathbf{x}_i = argmax_{\mathbf{x}}(f(\mathbf{x};\theta_j))$, where $\mathbf{x}$ is optimized from a random initialized image $\mathbf{x}_0$.

As shown in Fig. 8, methods without-MLP (Mocov1, Byol w/o MLP, SL) learn more knowledge about animals from pre-D, highlighted by red rectangles. This is due to that we select classes of organisms to construct pre-D. Instead, we find that methods with-MLP (Mocov1 w/ MLP, Byol, SL-MLP) learn more texture information. According to [29], high-level semantic information is less critical to transfer learning, which explains effectiveness of the MLP.

H. Detailed Training Setup

H.1. Pretraining

For SL and SL-MLP, we use the SGD optimizer with a cosine decay learning rate of 0.4 with Nesterov momentum of 0.9 to optimize all the networks and set the batch size to 1024. A 3 epochs warm-up with a starting learninig rate of 0.1 is applied. The weight decay of ResNets, MobileNetv2, EfficientNetb2 is set to 1×10^{-4}, 5×10^{-5}, 1×10^{-5}, respectively. Data augmentations include random-crop (224x224), color-jitter, and random horizontal flip. For SupCon and SupCon w/o MLP pretraining, we set the temperature parameter to $\tau = 0.07$, and queue size to 65596. We use random-crop (224x224), color-jitter, random gray-scale, Gaussian blur, random horizontal flip for pretraining data augmentations.

H.2. Concept Generalization Task

In unseen generalization task, we divide ImageNet-1K into two class-exclusive datasets following the hierarchical structure built in WordNet [15] - one for pretraining (denoted as pre-D) and the other for evaluation (denoted as eval-D). Eval-D has 348 classes of instrumentality, and pre-D contains 652 classes mostly of organisms. All the networks are pretrained on pre-D, and then examined by linear evaluation protocal on eval-D. As in [3, 12, 21], we train a linear classifier with the frozen backbone for 100 epochs. During evaluation, images are resized to 256 pixels, after which 224×224 center crop is used. We optimize the cross-entropy loss with SGD optimizer with cosine decay scheduler with Nesterov momentum of 0.9 over 100 epochs, using a batch size of 4096. We finally sweep over 7 learning rate over $\{0.16, 0.48, 1.44, 4.8, 14.4, 48\}$ and report the best accuracy on the test set of eval-D.

H.3. Transfer to Other Classification Tasks

Follow the downstream image classification tasks and the evaluation methods mentioned in [11], we use 12 datasets from different domains to evaluate the transferability of different methods, including natural [16, 18, 20], satellite [4, 10], symbolic [13, 17], illustrative [23, 25], medical [6, 26], and texture [5]. The statistics of datasets are illustrated in Tab. 3.

Linear Evaluation. For fixed-feature linear evaluation, we add a linear layer on the frozen pretrained backbone to train the model on the downstream datasets. A batch normalization layer is added between the backbone and linear layer. All models are trained for 50 epochs with step learning scheduler which decreases the learning rate by 0.1 at epoch 25 and 37. 70% of the training set is used for training and the rest is used for validation, the models are then trained with

- learning rate: 0.001, 0.01, 0.1;
- batch size: 32, 128;
- weight decay: 0, $1 \times 10^{-4}, 1 \times 10^{-5}$.

The optimal hyperparameters are chosen based on the performance on the validation set. The top-1 accuracy is reported as the evaluation metric.

Full Network Finetuning. In full network finetuning, the whole pretrained backbone and a linear classifier are trained on the downstream dataset. All models are trained for 50 epochs with step learning scheduler which decreases the learning rate by 0.1 at epoch 25 and 37. A batch normalization layer is added between the backbone and linear layer to make the extracted features comparable among different models. The models are trained with

- learning rate: 0.001, 0.01, 0.1;
- batch size: 32, 128;
- weight decay: 0, $1 \times 10^{-4}, 1 \times 10^{-5}$.

The optimal hyperparameters are chosen based on the performance on the validation set.

Few-shot Learning. For few-shot learning, following [24], we use a logistic regression layer on the top of the features during meta-testing phase. The implementation from scikit-learn is used for logistic regression. Same as [11], we also provide the mean of 600 randomly sampled tasks as the accuracy.

H.4. Object Detection on COCO

For object detection, we train Mask-RCNN [9] (R50-FPN) on COCO 2017 train split and report results on the val split. We use a learning rate of 0.001 and keep the other parameters the same as in the $1\times$ schedule in detectron2 [28].

Category	Dataset	Train Size	Test Size	Classes
Satellite	EuroSAT	18900	8100	10
	Resisc45	22005	9495	45
Natural	CropDisease	43456	10849	38
	Flowers	1020	6149	102
	DeepWeeds	12252	5257	9
Symbolic	Omniglot	9226	3954	1623
	SVHN	73257	26032	10
Medical	ISIC	7007	3008	7
	ChestX	18090	7758	7
Illustrative	Kaokore	6568	821	8
	Sketch	35000	15889	1000
Texture	DTD	3760	1880	47

Table 3. Datasets used for downsteam classification tasks from different domains. Following [11], we divided these datasets into six categories, including satellite, natural, symbolic, medical, illustrative, and texture.

Method	ChestX	CropDisease	DeepWeeds	DTD	EuroSAT	Flowers102	Kaokore	Omniglot	Resisc45	Sketch	SVHN	ISIC	Average
5-ways 5-shots few-shot classification													
SL	25.64	89.07	54.32	78.58	82.96	93.14	46.14	92.82	84.17	87.06	38.03	41.22	67.76
SL-MLP	26.89	93.45	59.08	83.04	87.16	96.88	50.77	95.73	89.00	89.84	41.96	46.76	**71.71**
SupCon w/o MLP	23.62	75.64	49.34	73.04	73.90	82.16	38.10	67.87	75.18	81.01	34.92	35.16	59.16
SupCon	26.18	94.09	59.36	85.02	87.97	96.55	51.02	94.49	89.01	89.75	41.67	43.48	<u>71.55</u>
5-ways 20-shots few-shot classification													
SL	30.05	94.15	64.54	85.74	89.13	96.63	55.65	97.17	90.34	93.12	48.09	52.06	74.72
SL-MLP	32.57	97.27	70.11	89.46	92.39	98.79	61.32	98.60	94.19	93.68	54.62	58.29	**78.44**
SupCon w/o MLP	26.50	84.90	57.81	80.64	82.37	89.47	46.19	83.56	83.51	88.12	44.60	44.51	67.68
SupCon	31.20	97.06	69.48	90.24	92.62	98.65	61.35	98.03	93.82	95.38	54.16	54.67	<u>78.06</u>

Table 4. 5-ways 5-shots and 20-shots classification performance on 12 downstream datasets in terms of top-1 accuracy. Using the code in [11], we pretrain all models over 300 epochs on ImageNet-1K. The reported accuracy is the mean of 600 randomly sampled tasks. Average results style: **best**, <u>second best</u>.

Method	ChestX	CropDisease	DeepWeeds	DTD	EuroSAT	Flowers102	Kaokore	Omniglot	Resisc45	Sketch	SVHN	ISIC	Average
Full-data finetuning													
SL	57.71	99.87	96.88	73.78	98.60	94.31	88.80	86.37	89.55	95.90	78.46	97.07	88.11
SL-MLP	57.98	99.88	96.90	74.26	98.77	95.12	89.16	88.81	90.06	96.15	79.83	97.13	<u>88.67</u>
SupCon w/o MLP	57.70	99.86	96.04	74.04	98.38	94.60	87.03	85.10	90.24	95.68	80.85	97.15	88.06
SupCon	58.61	99.90	96.29	75.43	98.83	95.10	88.83	87.35	91.25	95.72	81.10	97.23	**88.80**

Table 5. Full-data finetuning classification performance on 12 downstream datasets in terms of top-1 accuracy. Using the code in [11], we pretrain all models over 300 epochs on ImageNet-1K. The reported accuracy is the mean of 600 randomly sampled tasks. Average results style: **best**, <u>second best</u>.

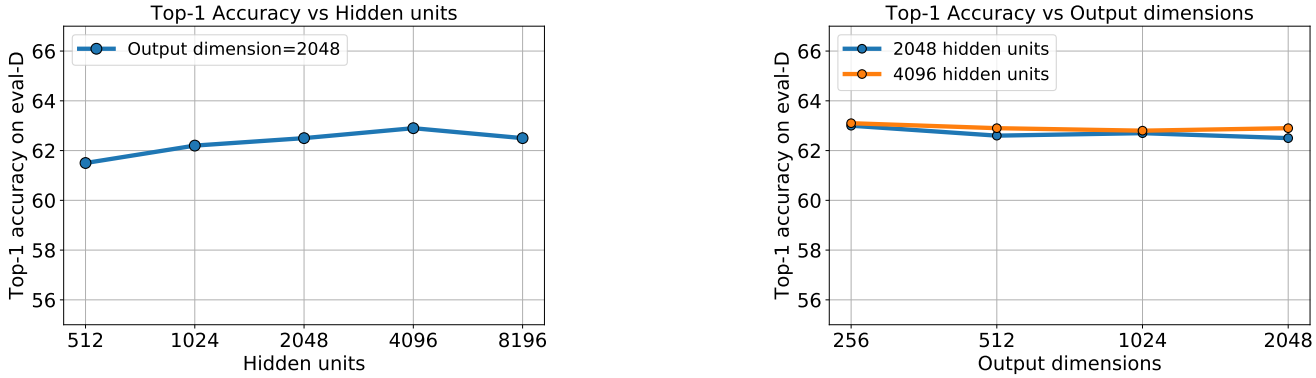
I. More Results

I.1. Few-shot Recognition Results

Using the code provided by [11], we pretrain all models over 300 epochs with a cosine decay learning scheduler on ImageNet-1K, and then testing on 12 downstream datasets (shown in Tab.3). We provide 5-ways 5-shots and 20-shots results in Tab. 4. All reported accuracy is the mean of 600 randomly sampled tasks. Comparing average results among different methods, we observe that supervised pretraining methods with the MLP projector, *i.e.* SL-MLP and SupCon, outperform their no MLP counterparts, *i.e.* SL and SupCon w/o MLP, on both 5-ways 5-shots and 20-shots few-shot classification tasks.

	Epoch	In-1K	L1	L2	L3	L4	L5
SL	300	**77.0**	66.2	60.1	56.1	54.7	48.3
Byol	300	71.7	68.2	64.4	60	58.7	52.9
SL-MLP	300	75.6	**70.4**	**66.2**	**61.8**	**60.8**	**54.4**

Table 6. Original concept generalization task [22] results. SL, SL-MLP, and Byol are all pretrained on ImageNet-1K with 300 epochs. Following [22], L1/L2/L3/L4/L5 represent five ImageNet-1K-sized datasets of increasing semantic distance from IN-1K as concept generalization levels.



(a) (b)

Figure 9. **(a)** Top-1 accuracy on eval-D as a function of the number of hidden units of the added MLP projector. **(b)** Top-1 accuracy on eval-D as a function of output dimension of the added MLP projector. We pretrain all the models on pre-D over 100 epochs and then evaluate on eval-D. Both hidden units and output dimensions show slight influence on the improved transferability.

I.2. Full-data Finetuning Results

We also provide full-sample results of 12-domains transfer task in Tab. 5, SL-MLP still gets a **+0.56%** accuracy gain. These consistent results show that adding an MLP on SL has large improvement on linear evaluation and observable improvement on fine-tuning (though relatively smaller).

I.3. Original Concept Generalization Task

We pretrain SL, SL-MLP, and Byol on ImageNet-1K with 300 epochs and use the code provided by [22] to evaluate their transferability on five ImageNet-1K-sized datasets of increasing semantic distance from IN-1K. Results are summarized in Tab. 6. SL-MLP is better than SL and Byol, and the improvement increases when the semantic distance increases from L1 (**+4.2**%) to L5 (**+6.1**%).

I.4. Ablation on Hidden Units and Output Dimensions

On concept generalization task, we also explore whether hidden units and output dimensions of the added MLP projector influence the final transferability. We pretrain SL-MLP on pre-D over 100 epochs using various hidden units and output dimensions of the added MLP projector, and report the evaluation results on eval-D (illustrated in Fig.9). We observe that, different from other unsupervised pretraining methods, *e.g*. BYOL and SimCLR, where the output dimension of the MLP projector have considerable impacts on transferability, the hidden units and output dimensions of the added MLP projector has little influence on the performance of SL-MLP.

J. Pretrain Results on pre-D

We also provide the top-1 accuracy of SL-MLP on pre-D in Tab. 7. We remove the MLP in SL-MLP for linear evaluation on pre-D, only the fixed backbones of SL and SL-MLP are used to train new classifiers over 100 epochs. We also report top-1 accuracy during pretraining in which accuracy of the whole SL-MLP is reported. Which features are used to evaluate these two metrics are illustrated in Fig. 10. As backbones and classifiers are jointly trained during pretraining, classifiers

Epochs	Top-1 accuracy during pretraining		Linear evaluation accuracy of fixed backbones	
	SL	SL-MLP	SL	SL-MLP
20	59.1	51.5	70.0	66.0
40	64.0	61.2	71.6	69.1
60	69.4	69.2	74.8	72.8
80	76.6	76.7	78.5	75.8
100	80.8	80.2	80.8	78.2

Table 7. Linear evaluation results and top-1 accuracy during pretraining on SL and SL-MLP. We remove the MLP in SL-MLP for linear evaluation, only the fixed backbones of SL and SL-MLP are used. For top-1 accuracy during pretraining, accuracy of the whole SL-MLP is reported.

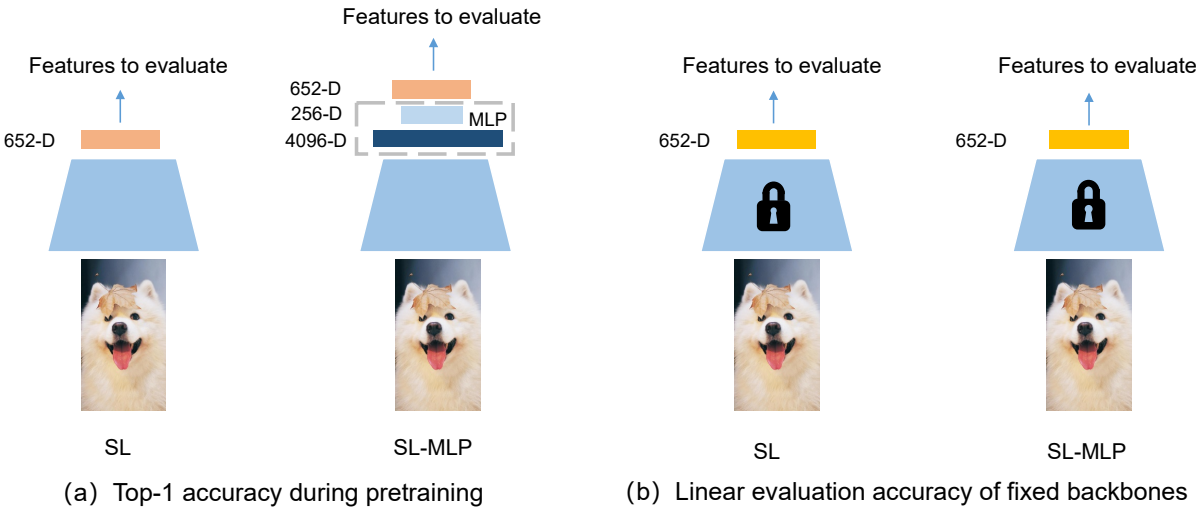


Figure 10. Evaluation of features extracted by SL and SL-MLP. **(a)**: During pretraining, features after the classifier is used to evaluate the accuracy on pre-D. **(b)**: After pretraining, we use the fixed backbones from different epochs to evaluate the performance of SL and SL-MLP.

are not well optimized at small pretraining epochs. Thus, models always achieve better performance on linear evaluation at small pretraining epochs because linear evaluation provides more epochs for networks to optimize better classifiers on fixed backbones. For SL, two evaluation methods display the same result at epoch 100, as they have all trained well-optimized classifiers.

Note that SL-MLP shows slight -2.6% performance drop (80.8% to 78.2%) on linear evaluation when SL and SL-MLP have all been pretrained over 100 epochs, which achieves closer performance gap than Exemplar-v2 [29] when compared with SL. Besides, as SL-MLP only adds an MLP projector before the classifier, the whole SL-MLP shows almost the same performance of SL on top-1 accuracy during pretraining at epoch 100.

References

[1] YM Asano, C Rupprecht, and A Vedaldi. A critical analysis of self-supervision, or what we can learn from a single image. In *International Conference on Learning Representations*, 2019. 9

[2] Edwin F Beckenbach, Richard Bellman, and Richard Ernest Bellman. An introduction to inequalities. Technical report, Mathematical Association of America Washington, DC, 1961. 6

[3] 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

[4] Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017. 11

[5] 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. 11

[6] Noel Codella, Veronica Rotemberg, Philipp Tschandl, M Emre Celebi, Stephen Dusza, David Gutman, Brian Helba, Aadi Kalloo, Konstantinos Liopyris, Michael Marchetti, et al. Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic). *arXiv preprint arXiv:1902.03368*, 2019. 11

[7] 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. *arXiv preprint arXiv:2006.07733*, 2020. 8

[8] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4918–4927, 2019. 8

[9] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 11

[10] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019. 11

[11] Ashraful Islam, Chun-Fu Chen, Rameswar Panda, Leonid Karlinsky, Richard Radke, and Rogerio Feris. A broad study on the transferability of visual representations with contrastive learning. *arXiv preprint arXiv:2103.13517*, 2021. 11, 12

[12] Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. Revisiting self-supervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1920–1929, 2019. 11

[13] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015. 11

[14] Bin Liu, Yue Cao, Yutong Lin, Qi Li, Zheng Zhang, Mingsheng Long, and Han Hu. Negative margin matters: Understanding margin in few-shot classification. In *European Conference on Computer Vision*, pages 438–455. Springer, 2020. 4

[15] George A Miller. *WordNet: An electronic lexical database*. MIT press, 1998. 11

[16] Sharada P Mohanty, David P Hughes, and Marcel Salathé. Using deep learning for image-based plant disease detection. *Frontiers in plant science*, 7:1419, 2016. 11

[17] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, pages 722–729, 2011. 11

[18] 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. 11

[19] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2(11):e7, 2017. 10

[20] Alex Olsen, Dmitry A Konovalov, Bronson Philippa, Peter Ridd, Jake C Wood, Jamie Johns, Wesley Banks, Benjamin Girgenti, Owen Kenny, James Whinney, et al. Deepweeds: A multiclass weed species image dataset for deep learning. *Scientific reports*, 9(1):1–12, 2019. 11

[21] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 11

[22] Mert Bulent Sariyildiz, Yannis Kalantidis, Diane Larlus, and Karteek Alahari. Concept generalization in visual representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9629–9639, 2021. 8, 13

[23] Yingtao Tian, Chikahiko Suzuki, Tarin Clanuwat, Mikel Bober-Irizar, Alex Lamb, and Asanobu Kitamoto. Kaokore: A pre-modern japanese art facial expression dataset. *arXiv preprint arXiv:2002.08595*, 2020. 11

[24] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pages 266–282. Springer, 2020. 11

[25] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. *Advances in Neural Information Processing Systems*, 32:10506–10518, 2019. 11

[26] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2097–2106, 2017. 11

[27] Hui-Hua Wu and Shanhe Wu. Various proofs of the cauchy-schwarz inequality. *Octogon mathematical magazine*, 17(1):221–229, 2009. 6

[28] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 11

[29] Nanxuan Zhao, Zhirong Wu, Rynson WH Lau, and Stephen Lin. What makes instance discrimination good for transfer learning? *arXiv preprint arXiv:2006.06606*, 2020. 9, 10, 14