IMPROVING THE TRANSFERABILITY OF SUPERVISED PRETRAINING WITH AN MLP PROJECTOR

Anonymous authors
Paper under double-blind review

ABSTRACT

The pretrain-finetune paradigm is a classical pipeline in visual learning. Recent progress on unsupervised pretraining methods showed superior transfer performance to their supervised counterparts. While a few works attempted to explore the underlying mechanisms, the reasons behind the transferability gaps still have not been fully explored. This paper reveals that the multilayer perceptron (MLP) projector is a key factor for the better transferability of unsupervised pretraining. Based on this observation, we attempt to close the transferability gap between supervised pretraining and unsupervised pretraining by adding an MLP projector before the classifier of supervised pretraining. Our analysis indicates that the MLP projector can help retain intra-class variation of visual features, decrease the feature distribution distance between pretraining dataset and evaluation dataset, and reduce feature redundancy for effective adaptation to new tasks. Extensive experiments demonstrate that the added MLP projector significantly boosts the transferability of supervised pretraining, *e.g.,* **+7.2%** top-1 accuracy on the unseen class generalization task and **+5.7%** top-1 accuracy on 12-domain classification tasks, making supervised pretraining even better than unsupervised pretraining.

1 INTRODUCTION

Transferring the knowledge learned from one task to another is a prominent ability of humans and an essential component for researchers to develop more general and robust artificial intelligence systems. Empowered by its automatic representation learning ability, deep neural networks can achieve transferability easily by reusing the pretrained models, *e.g.,* on the ImageNet-1K dataset, which helps neural networks achieve remarkable performance on various tasks, from computer vision (Ren et al., 2015; Song et al., 2018) to natural language understanding (Brown et al., 2020).

While Supervised Learning (SL) methods were the de facto pretraining paradigm in computer vision for a long period, recent Un-Supervised Learning (USL) methods have shown better transfer learning performance on various visual tasks. For example, the network pretrained with the unsupervised learning method Byol (Grill et al., 2020) has established a milestone on VOC2012 object detection (AP_{50} 77.5%) by significantly outperforming the supervised counterpart (AP_{50} 74.4%). This raised the question of why the unsupervised pretraining surpasses the supervised pretraining even though the supervised pretraining has used the collected annotations with rich semantic information.

Several works have attempted to explain the transferability discrepancy between the supervised pretraining and the unsupervised pretraining. These works mainly attributed the better transferability of the unsupervised pretraining to the following two reasons: (1) *Learning without semantic information in annotations,* which makes the backbone less-overfitted to semantic labels and keep the capacity to deal with instance-specific information that may be useful in transfer tasks (Ericsson et al., 2020; Wei et al., 2020; Zhao et al., 2021), and (2) *Special design of the contrastive loss,* which helps the learned features to contain more low/mid-level information that can be effectively transferred to downstream tasks (Zhao et al., 2021; Islam et al., 2021; Khosla et al., 2020). In this paper, we shed new light on understanding transferability by considering an MLP projector. With this new viewpoint, the transferability of supervised pretraining methods with the basic cross-entropy loss can be comparable or even better than representative unsupervised pretraining methods.

Specifically, we identify the multilayer perception (MLP) projector as a core factor for the transferability gap between existing SL and USL pretrining methods, and attempt to close the gap by

inserting the MLP projector in SL-based pretraining. Towards a better comparison of the models' transferability, we construct an *unseen class generalization task*, where pretrained models are evaluated to classify samples whose categories do not appear in the pretraining dataset with the fixed backbones. The experimental results and the corresponding analysis indicate that the MLP projector deployed in USL is important for better transferability. Motivated by this observation, we propose a simple yet effective supervised pretraining method SL-MLP, which inserts an MLP projector before the classifier in SL. SL-MLP can improve the transferability of the supervised pretraining pipeline and make it even better than the unsupervised pretraining. Experimental results on SL-MLP show three interesting findings: 1) SL-MLP preserves the intra-class variation on the pretraining dataset, which can be theoretically analyzed to improve the transferability on the evaluation dataset. 2) SL-MLP decreases the features' distribution distance between the pretraining set and the evaluation set; 3) SL-MLP decreases the redundancy of features in the pretraining dataset.

Large-scale experiments are conducted to evaluate the transferability of our SL-MLP on various visual tasks. The experimental results confirm that adding an MLP projector into the supervised pretraining can consistently improve transferability with various backbones and on different tasks. Specifically, in the *unseen class generalization task*, SL-MLP significantly boosts the top-1 accuracy of ResNet-50 (He et al., 2016) compared to the original SL (55.9%→63.1%). It also achieves **1.8**% higher top-1 accuracy than recent state-of-the-art USL methods Byol (64.1% vs. 62.3%) when both pretrained over 300 epochs. Besides, SL-MLP also improves the transfer performance on 12 downstream classification datasets from various domains by **5.7**% on average over the original SL pretraining and **1.3**% over the supervised contrastive learning method (Khosla et al., 2020). Moreover, SL-MLP shows consistent improvements on different downstream tasks, including object detection and instance segmentation on COCO dataset (Lin et al., 2014).

The main contributions of our paper are three-fold. (1) We reveal that the MLP projector is a main factor for the transferability disparity between existing unsupervised learning and supervised learning methods. (2) We empirically demonstrate that, by adding an MLP projector, supervised pretraining methods (SL-MLP) can match or even outperform unsupervised pretraining methods regarding transferability. (3) We theoretically prove that the MLP projector can improve pretrained networks' transferability by preserving the pretraining dataset's intra-class variation.

2 MLP MATTERS: TRANSFERABILITY ANALYSES OF USL AND SL

2.1 A NEW UNSEEN CLASS GENERALIZATION TASK FOR EVALUATING TRANSFERABILITY

Motivation: Typical downstream tasks to evaluate the transferability of pretrained models include classification on 12 datasets from different domains and detection/segmentation. However, the AP on detection/segmentation using Mask-RCNN (R50-FPN) by different pretraining methods varies slightly, *e.g.,* 37.4 to 38.9 in (Xie et al., 2021), which is not clearly distinguishable for comparing the transferability. Evaluating 12 datasets from different domains requires tuning hyperparameters, training the classifier, and assessing each dataset's accuracy, which is not convenient.

Implementation: We introduce an unseen class generalization task based on the ImageNet-1K dataset to avoid the above issues about transferability comparison. We divide ImageNet-1K into two semantically exclusive datasets following the hierarchical structure built in WordNet (Fellbaum, 2000) - one for pretraining (denoted as pre-D) and the other for evaluation (denoted as eval-D), which enlarges the transferability gaps among different methods and avoids evaluating on multiple datasets. Eval-D has 348 classes of instrumentality, and pre-D contains the other 652 classes in ImageNet-1K. To assess the transferability of a method under the proposed setting, we first pretrained the network on pre-D. Then, we freeze all parameters in the backbone [1], and finetune the classifier on eval-D for linear evaluation (He et al., 2020).

2.2 STAGE-WISE EVALUATION: MLP INFLUENCES TRANSFERABILITY

Motivated by the recent works (Zhao et al., 2021; Islam et al., 2021), which attribute the better transferability of unsupervised pretraining to the low/mid-level information preserved in the final

[1] All experiments in Section 2 and 3 are conducted with ResNet50, more backbones are tested in Section 4.

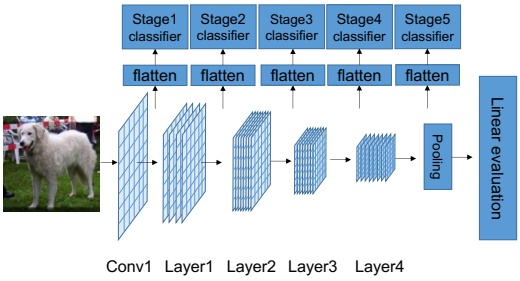

Figure 1: Schematic illustration of stage-wise evaluation. We flatten intermediate feature maps from different stages, and then use them to train stage-wise classifiers. Top-1 accuracy is reported by evaluating images in eval-D with the stage-wise classifiers.

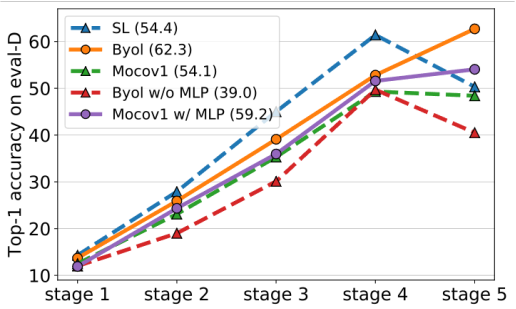

Figure 2: Top-1 accuracy of stage-wise evaluation. All methods use ResNet50 as their backbones and are trained by 300 epochs with the setting in original papers. The results of linear evaluation (after pooling of layer 4 in Figure 1) are reported in the legend.

features, we make a more thorough stage-wise investigation of supervised and unsupervised pretraining to evaluate the transferability of intermediate feature maps on eval-D. Specifically, we choose the basic SL with the cross-entropy loss, Byol, and MoCov1 for pretraining because of their representativeness in SL and USL, respectively. Then, we freeze all parameters of the pretrained model and use the extracted intermediate feature maps of images in eval-D to finetune a stage-wise classifier for linear evaluation (as shown Figure 1). Their stage-wise evaluation results are depicted in Figure 2. It can be observed that the linear evaluation accuracy of features after pooling (depicted in the legend) trained by SL is lower than those trained by Byol and similar to those trained by Mocov1, which is consistent with the observation in existing works (Grill et al., 2020; He et al., 2020). Owing to our stage-wise evaluation, we further get two new findings not reported by existing works. First, the linear evaluation accuracy of SL pretraining is consistently higher than Byol and Mocov1 from stage 1 to stage 4, which suggests that the semantic information in annotations can also benefit the transferability of low/middle-level feature maps. Second, Byol keeps an upward trend from stage 4 to stage 5 while SL and Mocov1 suffer performance drop from stage 4 to stage 5. By carefully inspecting three methods, we notice an architectural difference between SL, Mocov1 and Byol after stage 5: An MLP projector is inserted after stage 5 in Byol, which does not exist in SL and Mocov1. Such difference, together with the experimental results in Figure 2, motivates us to explore the influence of the MLP projector on transferability.

2.3 MLP IMPROVES THE TRANSFERABILITY OF USL PRETRAINING METHODS

To explore the contribution of MLP projectors to the transferability of USL pretraining methods, we ablate the MLP projectors on Byol (Grill et al., 2020) and Mocov1 (He et al., 2020) under our *unseen classes generalization task*. Specifically, we remove the MLP projector in Byol as Byol w/o MLP, and add an MLP projector in Mocov1 as Mocov1 w/ MLP. The stage-wise evaluation results are summarized in Figure 2, from which we obtain two observations. First, Byol and Moco w/ MLP achieve higher accuracy on eval-D than Byol w/o MLP and Mocov1 by $+23.3\%$ and $+5.1\%$ when evaluating the features after pooling with linear evaluation, respectively. Second, the MLP projector can avoid the transferability drop of Mocov1 from stage 4 to stage 5. These consistent improvements by MLP projectors empirically show that the MLP projector plays an important role in the transferability of the unsupervised pretraining.

3 SL-MLP: AN IMPROVED SUPERVISED PRETRAINING METHOD

3.1 MLP ENHANCED SUPERVISED PRETRAINING

Motivated by the empirical results in Section 2, an interesting question is whether MLP projector can also promote the transferability of supervised pretraining? We attempt to insert an MLP projector before the classifier on the SL pretraining for better transferability. We denote this supervised pretraining method as SL-MLP (see Figure 3 for their comparison). Specifically, SL-MLP includes

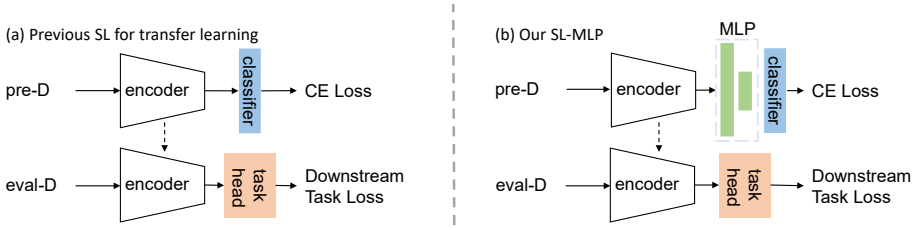

Figure 3: The difference between SL and SL-MLP. Our SL-MLP adds an MLP before the classifier compared to SL. Only the encoders in both methods are utilized for downstream tasks.

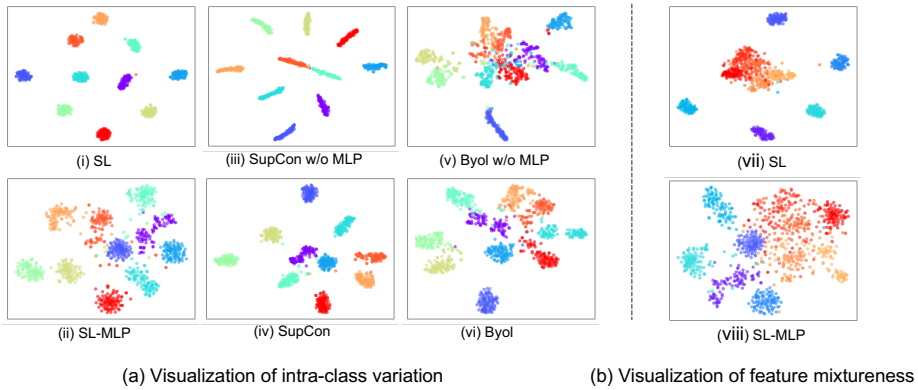

Figure 4: **(a)** Visualization of different methods with 10 randomly selected classes on pre-D. Different colors denote different classes. Features pretrained by methods without an MLP projector (top row) have less intra-class variation than those pretrained by methods with an MLP projector (bottom row). **(b)** Visualization of Feature Mixtureness between pre-D and 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.

a feature extractor $f(\cdot)$, an MLP projector $g(\cdot)$, and a classifier $\mathbf{W}$. Given an input image $\boldsymbol{x}$, the feature extractor outputs a feature $\boldsymbol{f} = f(\boldsymbol{x})$. For example, $f(\boldsymbol{x})$ transforms an image $\boldsymbol{x}$ to a 2048 dimensional feature $\boldsymbol{f}$ when using the ResNet-50 backbone. The MLP projector maps $\boldsymbol{f}$ into a projection vector $\boldsymbol{g} = g(\boldsymbol{f})$. Following Byol, the MLP projector consists of two fully connected layers, a batch normalization layer, and a ReLU layer, which can be mathematically formulated as $g(\boldsymbol{f}) = fc_2(ReLU(BN(fc_1(\boldsymbol{f})))) \in \mathbb{R}^{D_g}$, where fc_1 and fc_2 are fully connected layers, the hidden feature dimension in the MLP projector is set to 4096, and D_g is set to 256. Given the label denoted by y for image $\boldsymbol{x}$, the objective function for SL-MLP can be formulated as

$$\mathcal{L}(\boldsymbol{x}) = \text{CE}(\mathbf{W} \cdot g(f(\boldsymbol{x})), y), \tag{1}$$

where $\text{CE}(\cdot)$ is the cross-entropy loss in supervised pretraining methods. Same as SL, the learned feature extractor $f(\cdot)$ is utilized in downstream transfer tasks after the supervised pretraining.

3.2 Empirical findings for SL-MLP

MLP projector avoids transferability drop at stage 5 in supervised pretraining. We conduct the stage-wise evaluation as Section 2.2 again to see whether the transferability drop from stage 4 to stage 5 exists in SL-MLP. In Figure 5(a), the transferability of SL-MLP continuously increases from stage 1 to 5, avoiding a decrease at stage 5 as SL. Besides, when comparing SL-MLP to Byol, we observe that the transferability of SL-MLP is higher than that of Byol from stage 1 to stage 4, which indicates that annotations are helpful to the transferability of intermediate feature maps.

MLP projector enlarges the intra-class variation of $\boldsymbol{f}$. According to Zhao et al. (2021), features with large intra-class variations can preserve more instance discriminative information, which is beneficial for transfer learning. We reveal that adding MLP projector can enlarge the intra-class variation. We compare two supervised pretraining methods, *i.e.,* SL, SupCon (Khosla et al., 2020), and one unsupervised pretraining method, *i.e.,* Byol, with their variants with/without MLP. Qualita-

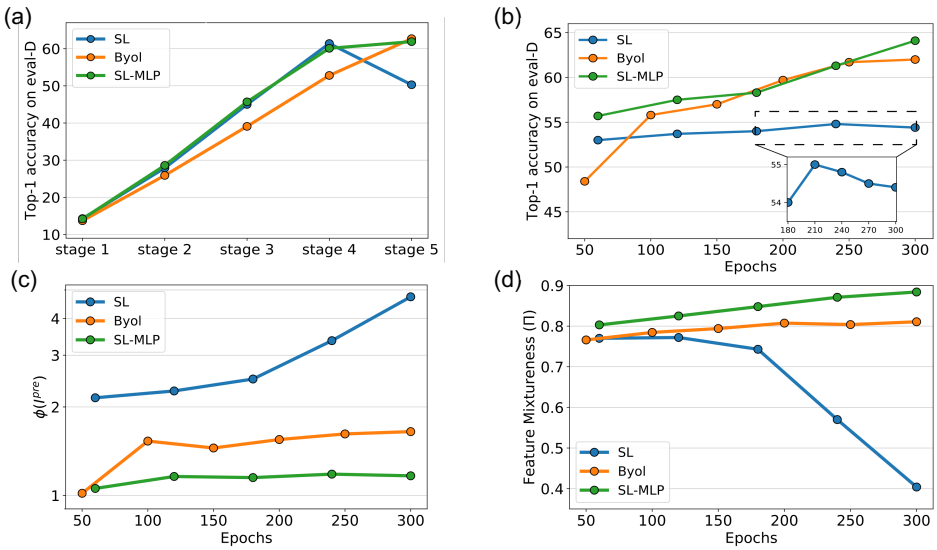

Figure 5: **(a)** Stage-wise evaluation on eval-D. **(b)** Linear evaluation accuracy on eval-D. **(c)** discriminative ratio of features on pre-D. **(d)**: Feature Mixtureness between pre-D and eval-D. Following He et al. (2019); Grill et al. (2020), we pretrain SL, SL-MLP and Byol for 300 epochs.

tively, we visualize the features learned on pre-D by t-SNE. As shown in Figure 4(a), the intra-class variations of features from SL-MLP, SupCon, and Byol are larger than that from SL, SupCon w/o MLP, and Byol w/o MLP, respectively. Quantitatively, following LDA (Balakrishnama & Ganapathiraju, 1998), we utilize a discriminative ratio $\phi(I^{pre}) = D_{inter}(I^{pre})/D_{intra}(I^{pre})$ to measure intra-class variation on pre-D, where I^{pre} denotes pre-D, D_{inter} and D_{intra} are inter-class distance and intra-class distance (mathematically defined in Section 3.3). Smaller discriminative ratio ϕ usually means larger intra-class variation[2]. Comparing Figure 5(c) with Figure 5(b), we can conclude that the smaller discriminative ratio ϕ, which means larger intra-class variations, can benefit transferability, with an exception for SL when the discriminative ratio on pre-D is too large in epochs 180-300. This exceptional phenomenon can be theoretically explained in Section 3.3. We additionally provide the visualization of intra-class variations on different pretraining epochs in Appendix B.1.

MLP projector reduces distribution distance of f between pre-D and eval-D. According to Blitzer et al. (2008); Liu et al. (2019), decreasing the feature distribution distance between pre-D and eval-D in the representation space can benefit transfer learning. Intuitively, the distribution distance between two sets of features is small when they are well mixed, and the distribution distance between two sets of features is large when they are separated. Therefore, we compare the mixtureness of features from pre-D and eval-D to indicate the feature distribution distance between SL and SL-MLP. Graphically, we visualize features from pre-D and eval-D by t-SNE in Figure 4(b). We find pre-D and eval-D features are better mixed by adding an MLP projector to SL, which indicates that MLP projector can mitigate the distribution distance between pre-D and eval-D. Quantitatively, we define the **Feature Mixtureness** Π in the feature space as

$$\Pi = 1 - \frac{1}{C}\sum_{i=1}^{C}\left|\frac{top_k^{eval}(i)}{k} - \frac{C^{eval}}{C}\right|, \tag{2}$$

where $C = 1000$ is total number of classes in ImageNet-1K, C^{eval} represents the number of classes in eval-D, and $top_k^{eval}(i)$ represents the number of classes in eval-D found by top k neighbors search of any class $i \in C$. Since the percentage of finding a sample from eval-D in k nearest neighbors is C^{eval}/C when pre-D and eval-D are uniformly mixed, Feature Mixtureness measures the similarity of the current and the uniformly mixed distribution between pre-D and eval-D in the feature space. We examine Feature Mixtureness of SL, SL-MLP and Byol during different pretraining epochs in Figure 5(d). Feature Mixtureness of SL gradually decreases, which indicates that SL will enlarge the

[2]Strictly speaking, larger intra-class variance is relative to inter-class distance, which is theoretically defined as discriminative ratio. We use "intra-class variation" to be consistent with previous work (Islam et al., 2021).

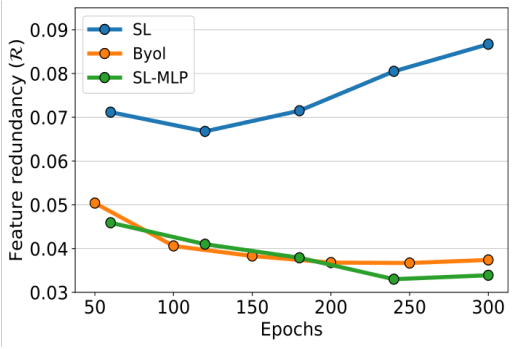

Figure 6: Redundancy $\mathcal{R}$ of pretrained features during different epochs. During large epochs, $\mathcal{R}$ increases in SL, but decreases in SL-MLP and Byol.

Method	Epoches	Top-1(↑)	$\mathcal{R}$(↓)
SL	100	55.9	0.078
SL-MLP	100	**63.1**	**0.035**
SL	300	54.4	0.087
SL-MLP	300	**64.1**	**0.034**
Byol w/o MLP	300	39.0	0.247
Byol	300	**62.3**	**0.037**
Mocov1	300	54.1	0.069
Mocov1 w/ MLP	300	**59.2**	**0.058**

Table 1: Redundancy $\mathcal{R}$ of pretrained features. Methods with an MLP projector obtain lower channel redundancy and better transferability.

distribution difference between pre-D and eval-D. In contrast, SL-MLP and Byol show consistently high Feature Mixtureness, indicating that SL-MLP can reduce the distribution distance between pre-D and eval-D. We visualize the relation between Feature Mixtureness and the feature distribution distance in Appendix A, and evolution of Feature Mixtureness in Appendix B.2.

MLP projector reduces the redundancy of $\boldsymbol{f}$. Inspired by Zbontar et al. (2021), high channel redundancy limits the capability of feature expression in deep learning. Mathematically, we compute Pearson correlation coefficient among feature channels to evaluate feature redundancy $\mathcal{R}$, *i.e,*

$$\mathcal{R} = \frac{1}{d^2}\sum_{i=1}^{d}\sum_{j=1}^{d}|\rho(i,j)|, \quad \rho(i,j) = \frac{\sum_{n=1}^{N}\boldsymbol{f}_{n,i}\cdot\boldsymbol{f}_{n,j}}{\sqrt{\sum_{n=1}^{N}||\boldsymbol{f}_{n,i}||^2}\sqrt{\sum_{n=1}^{N}||\boldsymbol{f}_{n,j}||^2}} \tag{3}$$

where $d = 2048$ is the feature dimension, $\rho(i,j)$ is Pearson correlation coefficient of feature channel i and j. As shown in Figure 6, SL-MLP achieves lower feature redundancy than SL, which validates that the MLP projector can reduce feature redundancy. With more pretraining epochs, the feature redundancy of SL-MLP even becomes smaller than unsupervised pretraining methods (*i.e.,* Byol in Figure 6). We further confirm the relationship between feature redundancy and transferability by conducting linear evaluation on the eval-D set with both supervised pretraining and unsupervised pretraining methods and the empirical results are reported in Table 1.

3.3 Theoretical analysis for SL-MLP

We provide a theoretical analysis about the empirical results in Figure 5(b) and Figure 5(c) on intra-class variation on pre-D and the feature distribution distance in Section 3.2. Specifically, we define pre-D and eval-D as I^{pre} and I^{eval}.We compute the j-th class center as $\mu(I_j) = \frac{1}{I_j}\sum_{(x_i,y_i)\in I_j}\boldsymbol{f}_i$, where $\boldsymbol{f}$ is the feature in Section 3.1. We define the inter-class distance $D_{inter}(I) = \frac{1}{C(C-1)}\sum_{j=1}^{C}\sum_{k=1,k\neq j}^{C}||\mu(I_j)-\mu(I_k)||^2$, and intra-class distance $D_{intra}(I)$ as $D_{intra}(I) = \frac{1}{C}\sum_{j=1}^{C}(\frac{1}{|I_j|}\sum_{(x_i,y_i)\in I_j}||\boldsymbol{f}_i-\mu(I_j)||^2)$, where C is the number of classes. According to LDA (Balakrishnama & Ganapathiraju, 1998), the discriminative ratio is $\phi(I) = D_{inter}(I)/D_{intra}(I)$, and higher discriminative ratio ϕ indicates higher classification accuracy. Following Liu et al. (2020), we analyze the relation between $\phi(I^{pre})$ and $\phi(I^{eval})$ in the Theory 1 below, which is detailed in Appendix G.

Theory 1: Given $\phi_1(I^{pre}) < \phi_2(I^{pre})$, $\phi_1(I^{eval}) > \phi_2(I^{eval})$ when $\phi_1(I^{pre}) > t$, where t is a threshold that is negatively related to the feature distribution distance.

Explanation of intra-class variations. In previous practices, researchers optimize the models on pre-D to achieve a better transferability to eval-D. However, comparing Figure 5(b) with Figure 5(c), we observe that too large discriminative ratio on pre-D will lead to transferability decrease on eval-D. Theory 1 explains this phenomenon because if the discriminative ratio $\phi(I^{pre})$ on pre-D is larger than a threshold t, further optimizing $\phi(I^{pre})$ will lead to decreasing $\phi(I^{eval})$.

Table 2: Unseen class generalization tasks. We report Top-1 accuracy on eval-D to compare the transferability of SL-MLP, Byol and SL on various backbones. SL-MLP and Byol share the same MLP projector.

Method	Architecture	Labels	MLP	Epochs	Top-1 on eval-D (↑)
SL	ResNet50	✓		100	55.9
SL-MLP	ResNet50	✓	✓	100	63.1
Byol	ResNet50		✓	300	62.3
SL	ResNet50	✓		300	54.4
SL-MLP	ResNet50	✓	✓	300	**64.1**
SL	ResNet34	✓		100	50.1
SL-MLP	ResNet34	✓	✓	100	55.0
Byol	ResNet34		✓	300	54.8
SL	ResNet34	✓		300	50.2
SL-MLP	ResNet34	✓	✓	300	**55.8**
SL	ResNet101	✓		100	56.0
SL-MLP	ResNet101	✓	✓	100	63.6
SL	ResNet101	✓		300	53.9
SL-MLP	ResNet101	✓	✓	300	**64.7**
SL	MobileNetv2(s=1.4)	✓		200	54.5
SL-MLP	MobileNetv2(s=1.4)	✓	✓	200	**61.5**
SL	EfficientNetb2	✓		100	57.6
SL-MLP	EfficientNetb2	✓	✓	100	**64.2**

Explanation of the relation between the feature distribution distance and threshold t. When the feature distribution distance (smaller feature distribution distance corresponds to more mixed distribution between data from pre-D and data from eval-D) is large (*e.g.,* when the pretraining data and the evaluation data have large discrepancy), the threshold t is small, in which case it is easier to have the undesirable effect of increasing the discriminative ratio $\phi(I^{pre})$ on pre-D leading to decreasing the discriminative ratio $\phi(I^{eval})$ on eval-D (and thus the accuracy on the evaluation data).

4 Experiment

4.1 Experimental setup

Datasets. For backbone analysis, we keep using the unseen class generalization setting described in section 2.1. For generalization to other classification tasks, we follow the setup in Islam et al. (2021), which pretrains the models on the whole ImageNet-1K dataset and then evaluate the transferability on 12 classification datasets from different domains (detailed in Appendix H.3). Furthermore, the COCO (Lin et al., 2014) dataset is used to evaluate the performance of SL-MLP pretrained by ImageNet-1K on object detection and instance segmentation tasks.

Pretraining details. For SL and SL-MLP pretraining, the cross-entropy is deployed as the loss function. The MLP projector deployed in SL-MLP is described in Section 3.1. Following He et al. (2016), we use the SGD optimizer with a cosine decay learning rate of 0.4 to optimize SL and SL-MLP, and set the batch size to 1024. Byol is used as a representative method for comparisons in the backbone analysis and object detection/instance segmentation. Following Grill et al. (2020), we use LARS optimizer (You et al., 2017) with a cosine decay learning rate schedule and a warm-up of 10 epochs to pretrain the network. The initial learning rate is set to 4.8. We set the batch size to 4096 and the initial exponential moving average parameter τ to 0.99. Except for the backbone analysis, we use ResNet50 as the default backbone. More detailed pretraining setups of different backbones and different methods are provided in Appendix H.1.

4.2 Experimental results

Generalize to unseen classes with diverse backbones. We analyze the transferability of SL-MLP on *unseen class generalization task* with different backbones. Following van den Oord et al. (2018); Kolesnikov et al. (2019); Chen et al. (2020), we train a linear classifer with the frozen backbone

Table 3: Linear evaluation performance of different supervised learning methods on 12 classification datasets in terms of top-1 accuracy. All models are pretrained for 300 epochs with the same code base.

Method	ChestX	CropDisease	DeepWeeds	DTD	EuroSAT	Flowers102	Kaokore	Omniglot	Resisc45	Sketch	SVHN	Average
SL	45.25	95.64	83.59	94.53	94.21	98.66	77.99	80.79	87.73	85.72	60.31	82.22
SL-MLP	**48.82**	**98.98**	**90.81**	**98.84**	**96.76**	**99.9**	**86.83**	**88.8**	**95.38**	**95.14**	**66.7**	**87.91**
SupCon w/o MLP	41.05	88.66	72.13	84.16	88.22	88.84	67.97	48.98	77.3	66.41	53.74	70.68
SupCon	48.39	98.61	88.16	97.31	95.97	99.8	82.7	87.96	94.33	94.48	65.15	86.62

Table 4: Object detection and instance segmentation fine-tuned on COCO using Mask-RCNN (R50-FPN).

Method	Epoch	Object detection			Instance segmentation		
		AP	AP50	AP75	AP	AP50	AP75
SL	100	38.9	59.6	42.7	35.4	56.5	38.1
SL-MLP	100	39.7	60.4	43.1	35.2	57.1	37.6
SL	300	40.1	61.1	43.8	35.7	57.7	38.0
Byol	300	39.4	60.4	43.2	34.9	55.3	37.5
SL-MLP	300	**40.7**	**61.8**	**44.2**	**36.1**	**58.4**	**38.5**

for 100 epochs, and report the top-1 accuracy on eval-D in Table 2. Firstly, SL-MLP obtains better performance than SL among different backbones. Specifically, with ResNet50, SL-MLP improves SL to 63.1 (+7.2%) when we pretrain the model by only 100 epochs, which bridges the performance gap between supervised pretraining and Byol. In 300 epochs setting, SL suffers from a transferability drop compared to 100 epochs setting (55.9%→54.4%), but the transferability of SL-MLP continue to increase (63.1%→64.1%). Secondly, SL-MLP achieves a better performance (64.1%) than Byol (62.3%) when both are trained by 300 epochs for a fair comparison. More experimental results in Table 2 also confirm that SL-MLP can consistently improve the transferability of supervised pretraining on various backbones, *e.g.,* ResNet101 (He et al., 2016), MobileNetv2(Sandler et al., 2018), and EfficientNetb2 (Tan & Le, 2019).

Generalize to other classification tasks. To evaluate if SL-MLP transfers well when it meets cross-domain tasks, following Islam et al. (2021), we pretrain the model on ImageNet-1K, and use their source code to evaluate the transferability on 12 classification datasets from different domains. As illustrated in Table 3, 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, by at least 5.7% on the average Top-1 accuracy. Besides, by comparing SupCon, SL-MLP and SupCon w/o MLP, SL, we conclude that the MLP projector instead of the contrastive loss plays the key role in increasing transferability. Our conclusion contrasts with previous works (Zhao et al., 2021; Islam et al., 2021) because they ignore the MLP projector before contrastive loss.

Generalize to object detection and instance segmentation tasks. We assess the transferability of SL-MLP beyond classification by object detection and instance segmentation tasks. In Table 4, we report results on COCO using Mask-RCNN (He et al., 2017) (R50-FPN), as detailed in Appendix H. In 100 epochs setting, SL-MLP consistently improves SL on detection and segmentation tasks, and achieves comparable performance with Byol (300 epochs). In 300 epochs setting, the performance of SL-MLP is *better* than SL and Byol. Concretely, SL-MLP outperforms SL (**+0.6/+0.4 AP**) and Byol (**+1.3/+1.2 AP**) on object detection and instance segmentation, respectively.

4.3 Ablation study

Effectiveness of different components in the MLP Projector. In this part, we investigate the influence of different components in the MLP projector, including an input fully connected layer, a batch normalization layer, a ReLU layer, and an output fully connected layer. We set the hidden units and output dimension of MLP to be 2048 to retain the dimension of output features the same as SL. Variants are constructed by adding the components incrementally: (a) no MLP projector; (b) only Input FC; (c) Input FC+BN+output FC; (d) Input FC+ReLU+output FC; (e) BN+ReLU. All experiments are pretrained on pre-D over 100 epochs. Their testing accuracies on eval-D are reported in Table 5, showing that SL-MLP achieves the best accuracy among all variants. We analyze the influence of different components on discriminative ratio ϕ on pre-D, Feature Mixtureness Π, feature redundancy $\mathcal{R}$ qualitatively and quantitatively in Appendix D.3. Besides, we also observe

Table 5: Empirical analysis of structural design of the MLP projector. We incrementally add different component to the MLP projector. We pretrain models over 100 epochs and set the output dimension to 2048.

Exp	Components				Additional params	Top-1	Gain
	Input FC	BN	ReLU	Output FC			
(a)					/	55.9	/
(b)	✓				4.196M	56.6	+0.7
(c)	✓	✓		✓	8.395M	61.0	+5.1
(d)	✓		✓	✓	8.391M	60.1	+4.2
(e)		✓	✓		0.004M	60.5	+4.6
SL-MLP	✓	✓	✓	✓	8.395M	62.5	+6.6

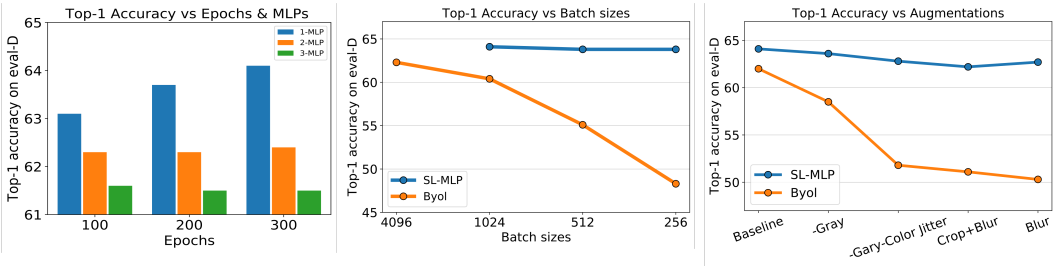

Figure 7: (Left to right) **(a)** Top-1 accuracy with different pretraining epochs and number of MLP projectors. **(b)** Top-1 accuracy with different batch sizes shows that SL-MLP has more robust transferability to small batch size. **(c)** Top-1 accuracy with different pretraining augmentations shows SL-MLP is robust to augmentations.

an interesting phenomenon on Table 5(e) that only inserting a lightweight BN-ReLU also achieves good transfer performance. As this is not our main focus, we will investigate it in future works.

Epochs and layers. Figure 7(a) shows that adding more MLP projectors in SL-MLP leads to a worse transferability. In addition, larger pretraining epochs benefit the transferability of SL-MLP when one MLP projector is inserted, but it has little influence when more MLP projectors are used.

Batch size. Most unsupervised methods depend on big mini-batches to train a representation with strong transferability. To investigate the sensitivity of SL-MLP to batch size, we do experiments with batch size from 256 to 4096 for Byol (following Grill et al. (2020)) and to 1024 for SL-MLP over 300 epochs. As shown in Figure 7(b), the transferability of Byol drops when the batch size decreases. In contrast, the transferability of SL-MLP retains over batch size from 128 to 1024.

Augmentation. Unsupervised methods benefit from a boarder space of colors and more intensive augmentations during pretraining, which always lead to undesirable degradation when some augmentations are missing. Supervised models trained merely with horizontal flipping may perform well (Zhao et al., 2021). We set Byol's augmentations as a baseline setting for both SL-MLP and Byol. We then compare the robustness on augmentation between SL-MLP and Byol by removing augmentation step by step. Experiments of SL-MLP and Byol are all constructed on their default condition with only augmentations changed. The results are illustrated on Figure 7(c) . We find that SL-MLP inherits the robustness of SL and shows a little accuracy drop with simple augmentations.

5 Conclusion

In this paper, we study the transferability gaps between supervised and unsupervised pretraining. Based on empirical results, we identify and argue that the MLP projector is a key factor for the good transferability of unsupervised pretraining methods. By incorporating an MLP projector into supervised pretraining methods, we close the gap between supervised and unsupervised pretraining and even make supervised pretraining better. Our finding is confirmed with large-scale experiments on diverse backbone networks and various downstream tasks, including unseen classes generalization tasks, cross-domain image classifications, objection detection, and instance segmentation.

REFERENCES

Yuki M Asano, Christian Rupprecht, and Andrea Vedaldi. A critical analysis of self-supervision, or what we can learn from a single image. *arXiv preprint arXiv:1904.13132*, 2019.

Suresh Balakrishnama and Aravind Ganapathiraju. Linear discriminant analysis-a brief tutorial. In *ISIP*, 1998.

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

John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman. Learning bounds for domain adaptation. In *NIPS*, 2008.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *P IEEE*, 2017.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *CVPR*, 2014.

Noel C. F. Codella, Veronica Rotemberg, Philipp Tschandl, M. Emre Celebi, Stephen W. Dusza, David A. Gutman, Brian Helba, Aadi Kalloo, Konstantinos Liopyris, Michael A. Marchetti, Harald Kittler, and Allan Halpern. Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic). *arXiv preprint arXiv:1902.03368*, 2019.

Linus Ericsson, Henry Gouk, and Timothy M. Hospedales. How well do self-supervised models transfer? In *CVPR*, 2020.

Christiane Fellbaum. Wordnet : an electronic lexical database. *Language*, 2000.

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

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *CVPR*, 2017.

Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *CVPR*, 2019.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

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 J-STARS*, 2019.

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

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *NeurIPS*, 2020.

Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. Revisiting self-supervised visual representation learning. In *CVPR*, 2019.

Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 2015.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

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 *ECCV*, 2020.

Hong Liu, Mingsheng Long, Jianmin Wang, and Michael I Jordan. Towards understanding the transferability of deep representations. *arXiv preprint arXiv:1909.12031*, 2019.

Sharada Prasanna Mohanty, David Peter Hughes, and Marcel Salathe. Using deep learning for image-based plant disease detection. *FRONT PLANT SCI*, 2016.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS*, 2011.

M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *ICVGIP*, 2008.

Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. 2017.

Alex Olsen, Dmitry A. Konovalov, Bronson Philippa, Peter Ridd, Jake C. Wood, Jamie Johns, Wesley Banks, Benjamin Girgenti, Owen Kenny, James Whinney, Brendan Calvert, Mostafa Rahimi Azghadi, and Ronald D. White. Deepweeds: A multiclass weed species image dataset for deep learning. *SCI REP-UK*, 2019.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018.

Yuhang Song, Chao Yang, Zhe Lin, Xiaofeng Liu, Qin Huang, Hao Li, and C-C Jay Kuo. Contextual-based image inpainting: Infer, match, and translate. In *ECCV*, 2018.

Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019.

Yingtao Tian, Chikahiko Suzuki, Tarin Clanuwat, Mikel Bober-Irizar, Alex Lamb, and Asanobu Kitamoto. Kaokore: A pre-modern japanese art facial expression dataset. *ICCV*, 2020.

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

Haohan Wang, Songwei Ge, Zachary C. Lipton, and Eric P. Xing. Learning robust global representations by penalizing local predictive power. In *NeurIPS*, 2019.

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 *CVPR*, 2017.

Longhui Wei, Lingxi Xie, Jianzhong He, Jianlong Chang, Xiaopeng Zhang, Wengang Zhou, Houqiang Li, and Qi Tian. Can semantic labels assist self-supervised visual representation learning? *arXiv preprint arXiv:2011.08621*, 2020.

Hui-Hua Wu and Shanhe Wu. Various proofs of the cauchy-schwarz inequality. *Octogon mathematical magazine*, 2009.

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

Enze Xie, Jian Ding, Wenhai Wang, Xiaohang Zhan, Hang Xu, Zhenguo Li, and Ping Luo. Detco: Unsupervised contrastive learning for object detection. *arXiv preprint arXiv:2102.04803*, 2021.

Yang You, Igor Gitman, and Boris Ginsburg. Scaling sgd batch size to 32k for imagenet training. *arXiv preprint arXiv:1708.03888*, 2017.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. *arXiv preprint arXiv:2103.03230*, 2021.

Nanxuan Zhao, Zhirong Wu, Rynson W. H. Lau, and Stephen Lin. What makes instance discrimination good for transfer learning. In *ICLR*, 2021.

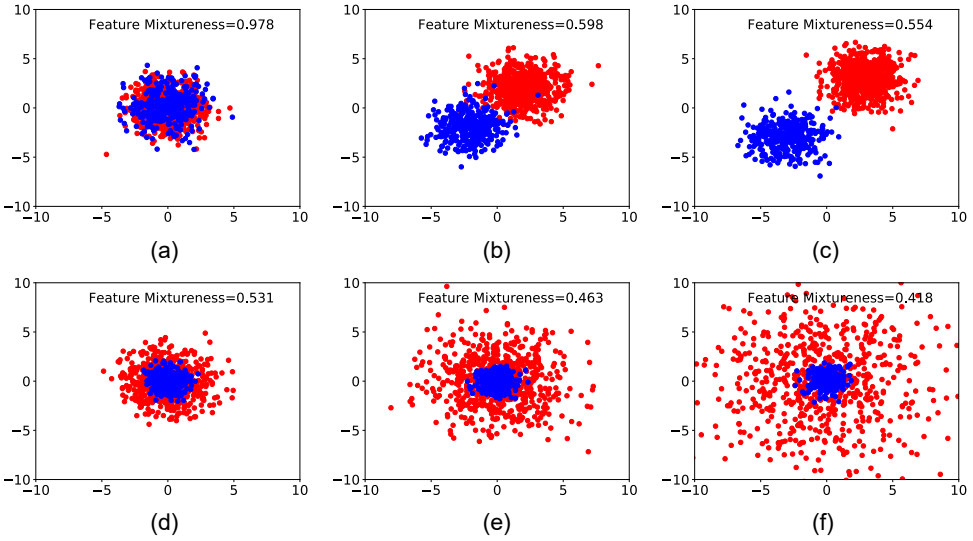

Figure 8: Visualization of Feature Mixtureness with different manually generated feature distribution. Red and blue represent pre-D and eval-D class centers, respectively.

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 Figure 8. 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 Figure 9 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 Figure 10. 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 remains higher Feature Mixtureness when the training epoch

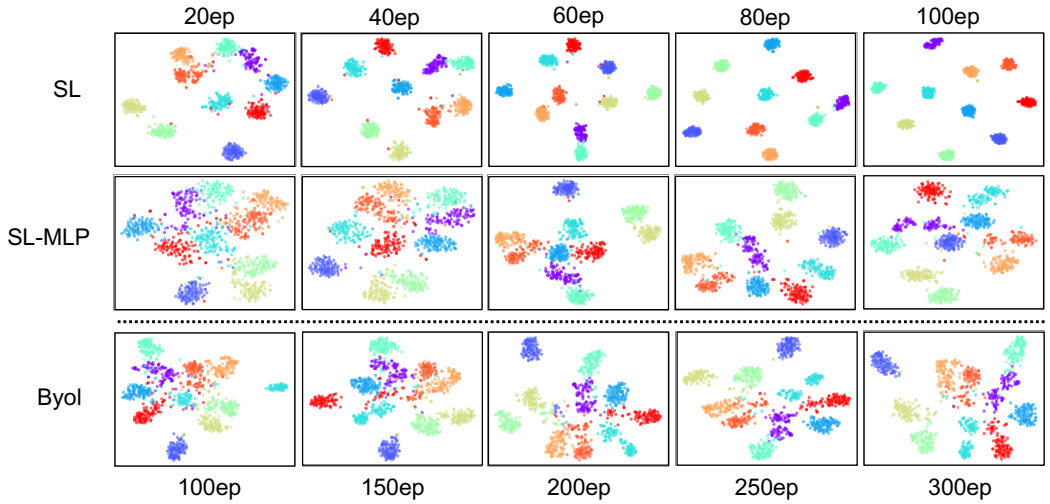

Figure 9: Evolution of intra-class variation of features in pre-D with different epochs. Different colors denote different classes. The intra-class variation of SL will be very small when the pretraining epoch is large enough. Instead, the intra-class variation of SL-MLP and Byol still retains even though the model is pretrained by large epochs.

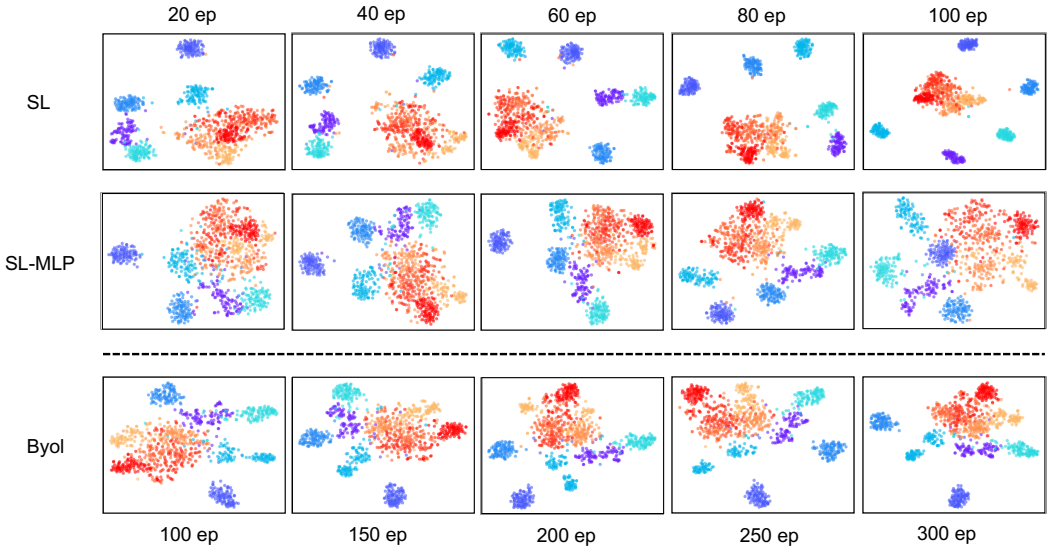

Figure 10: Evolution of Feature Mixtureness between features from pre-D and from eval-D. Cold colors denote features from 5 classes that are randomly selected from pre-D, and warm colors denote features from 5 classes that are randomly selected from eval-D. Feature Mixtureness of SL continuously decrease during pretraining. Alternatively, SL-MLP and Byol keeps a relatively high Feature Mixtureness at large pretraining epochs.

is becoming larger, which shows that the feature distribution distance is not enlarged by Byol and SL-MLP.

C VISUALIZE CONVOLUTION CHANNELS BY OPTIMIZATION

According to Zhao et al. (2021) and Asano et al. (2019), 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 the method pro-

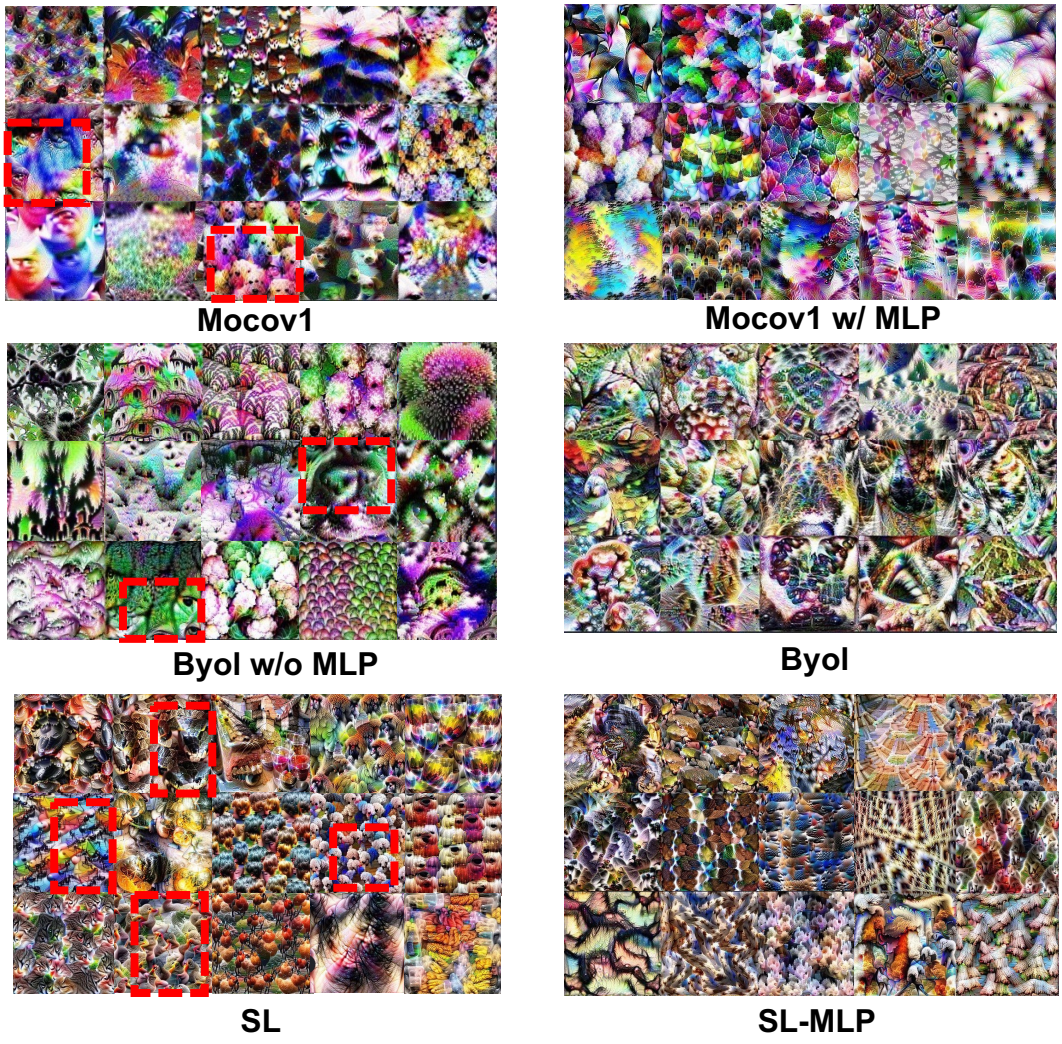

Figure 11: Convolution channels visualization of Mocov1, Mocov1 w/ MLP, Byol w/o MLP, Byol, SL and SL-MLP. Following the method proposed in Olah et al. (2017), we visualize the maximum response of convolution channels in layer 4 of ResNet50 pretrained with different methods.

posed in Olah et al. (2017), we visualize the maximum response of convolution channels in layer 4 of ResNet50 pretrained with methods with-/without-MLP (see in Figure 1) . Specifically, given a backbone with fixed parameters $\boldsymbol{\theta}$ as $f(\cdot;\boldsymbol{\theta})$, we denote the parameters before the convolution channel i as $f(\cdot;\boldsymbol{\theta}_i)$, we optimize the most representative sample $\boldsymbol{x}_i$ of the convolution channel i by maximizing the output logits $f(\boldsymbol{x};\boldsymbol{\theta}_i)$, *i.e.,* $\boldsymbol{x}_i = \arg\max_{\boldsymbol{x}}(f(\boldsymbol{x};\boldsymbol{\theta}))$, where $\boldsymbol{x}$ is optimized from a random initialized image $\boldsymbol{x}_0$.

As shown in Figure 11, 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 organism to construct pre-D. Instead, we find that methods with MLP (Mocov1 w/ MLP, Byol, SL-MLP) learn more texture information. According to Zhao et al. (2021), high-level semantic information is less critical to transfer learning, which explains the better performance of methods with MLP.

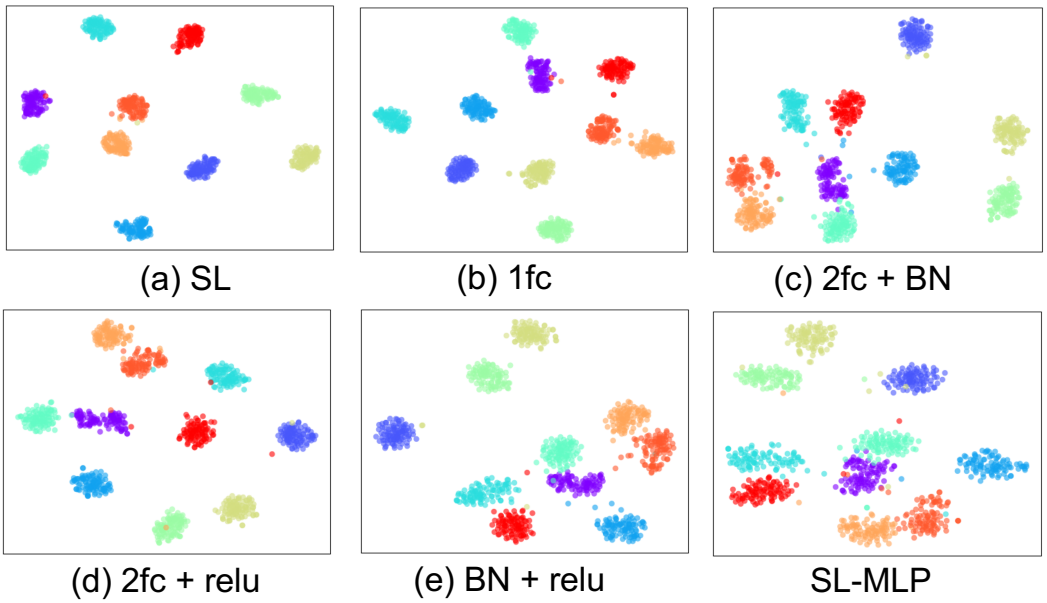

Figure 12: Visualization of intra-class variation by different components. We randomly select 10 classes in pre-D. Different colors denote different classes. Comparing (a) wth (b), we can see the fully-connected layer can slightly help enlarge the intra-class variation. Comparing (a-b) and (d-e), we 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.

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. 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 discriminative ratio on pre-D, Feature Mixtureness Π and feature redundancy $\mathcal{R}$ are illustrated in Table 6.

D.1 VISUALIZATION OF INTRA-CLASS VARIATION

We randomly select features from 10 classes in pre-D and visualize their intra-class variation in Figure 12. 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 Figure 13. The features

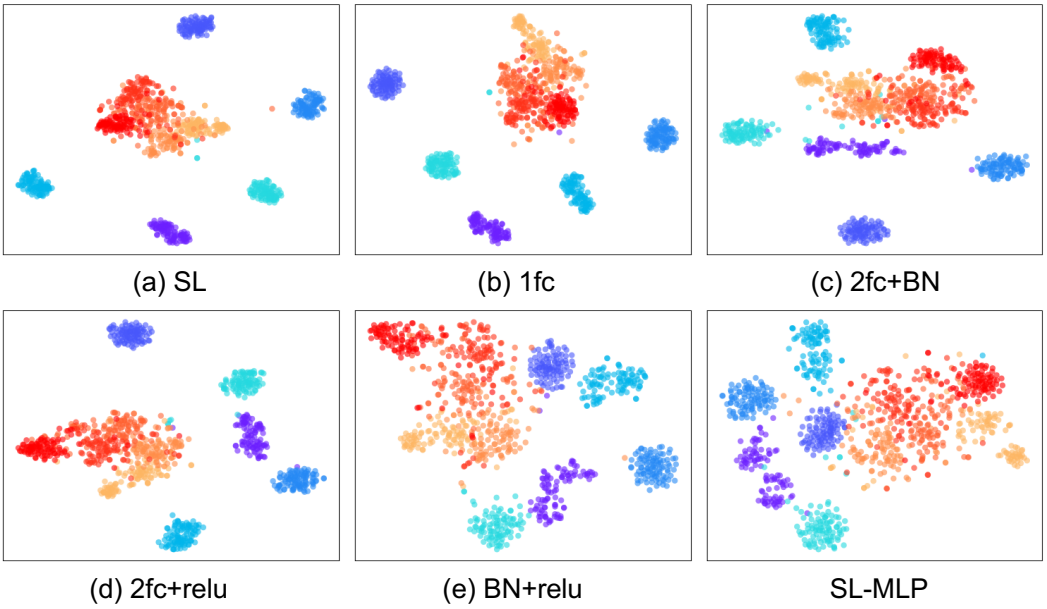

Figure 13: Visualization of Feature Mixtureness of features pretrained by different MLP components. Different colors denote different classes. Points with cold colors denote the features from pre-D, and points with warm colors denote the features from eval-D. Comparing (c-d) with (a-b), we can see that 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.

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 structure.

D.3 Quantitative analyse of MLP components

With the discriminative ratio ϕ^{pre}, Feature Mixtureness Π and feature redundancy $\mathcal{R}$ defined in Section 3.2, we quantitatively examine the effect of different components in the MLP projector. The results are presented in Table 6. 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 discriminative ratio on pre-D. Second, non-linear layer brings considerable improvements. Comparing (b) to (d), we can summarize that 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 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-variance.

Table 6: 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.

Exp	Components				Top-1	$D_{inter}^{pre}/D_{intra}^{pre}$	$\Pi(\uparrow)$	$\mathcal{R}(\downarrow)$
	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

E UNSEEN CLASS GENERALIZATION TASK WITH SMALL SEMANTIC GAP

To investigate how semantic difference between pre-D and eval-D can influence the transfer results in *unseen classes generalizatino tasks*, 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 Section 2.1 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*). We visualize features from pre-D and eval-D in small semantic gap setting and large semantic gap setting in Figure 14. Specifically, SL (random), SL-MLP (random), 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), 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 Section 3.2. 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.

F REPLACING SOFTMAX WITH COSINE-SOFTMAX

In order to prove that our finding 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

$$\mathcal{L}_{\text{cos}}(\boldsymbol{x}_i, y_i) = -\log \frac{\exp(\beta \cdot \cos(\boldsymbol{w}_{y_i}, f(\boldsymbol{x}_i)))}{\sum_{i=j}^{C} \exp(\beta \cdot \cos(\boldsymbol{w}_j, f(\boldsymbol{x}_i)))}, \tag{4}$$

where $\boldsymbol{w}_i$ 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.,***

$$\mathcal{L}_{\text{cos-mlp}}(\boldsymbol{x}_i, y_i) = -\log \frac{\exp(\beta \cdot \cos(\boldsymbol{w}_{y_i}, g(f(\boldsymbol{x}_i))))}{\sum_{i=j}^{C} \exp(\beta \cdot \cos(\boldsymbol{w}_j, g(f(\boldsymbol{x}_i))))}, \tag{5}$$

where $\boldsymbol{w}_i$ 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 Table 7. 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.

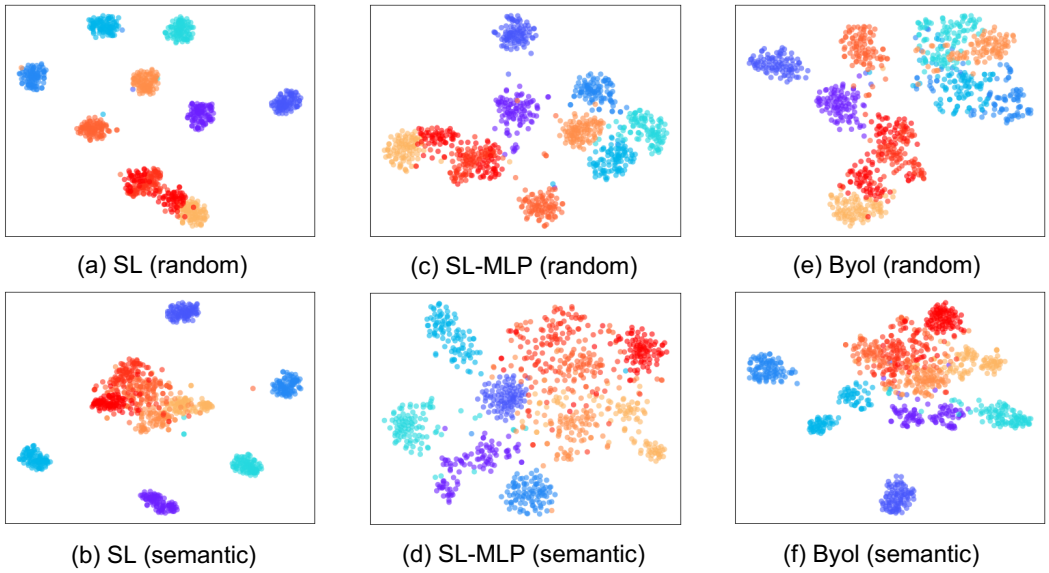

Figure 14: Visualization of Feature Mixtureness between pretraining dataset (pre-D) and evaluation dataset (eval-D). Different colors denote different classes. Classes in pre-D are denoted by cold colors, and classes in eval-D are denoted by warm colors. Comparing (a,c,e) and (b,d,f), we can conclude 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.

Table 7: Top-1 linear evaluation accuracy on eval-D when pretraining the model in pre-D by cosine softmax cross-entropy loss.

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**

G THEORETICAL ANALYSIS OF THEORY 1

G.1 PROOF OF THEORY 1

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

$$\mu(I_j) = \frac{1}{I_j} \sum_{(x_i,y_i)\in I_j} \boldsymbol{f}_i, \tag{6}$$

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

$$D_{inter}(I) = \frac{1}{C(C-1)} \sum_{j=1}^{C} \sum_{k=1,k\neq j}^{C} ||\mu(I_j) - \mu(I_k)||^2, \tag{7}$$

$$D_{intra}(I) = \frac{1}{C} \sum_{j=1}^{C} (\frac{1}{|I_j|} \sum_{(x_i,y_i)\in I_j} ||\boldsymbol{f}_i - \mu(I_j)||^2). \tag{8}$$

Substituting Eq. 6 into Eq. 7 and Eq. 8, we have

$$D_{inter}(I) = \frac{1}{C(C-1)} \sum_{j=1}^{C} \sum_{k=1,k\neq j}^{C} \left(\frac{1}{2|I_j||I_k|} \sum_{(\boldsymbol{x}_i,y_i)\in I_j} \sum_{(\boldsymbol{x}_l,y_l)\in I_k} ||\boldsymbol{f}_i - \boldsymbol{f}_l||^2 \right), \tag{9}$$

$$D_{intra}(I) = \frac{1}{C} \sum_{j=1}^{C} \left(\frac{1}{2|I_j|^2} \sum_{(\boldsymbol{x}_i,y_i)\in I_j} \sum_{(\boldsymbol{x}_l,y_l)\in I_j} ||\boldsymbol{f}_i - \boldsymbol{f}_l||^2 \right). \tag{10}$$

Taking expectation to Eq. 9 and Eq. 10, for any pair of data $(\boldsymbol{x}_i, y_i), (\boldsymbol{x}_l, y_l) \in I$, we have

$$\mathbb{E}(||\boldsymbol{f}_i - \boldsymbol{f}_l||^2) = \begin{cases} 2D_{intra}(I), y_i = y_l \\ 2D_{inter}(I), y_i \neq y_l \end{cases}. \tag{11}$$

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

$$\begin{aligned} D_{intra}(I^{eval}) &= \frac{1}{2}\mathbb{E}\left(||\boldsymbol{f}'_i - \boldsymbol{f}'_l||^2\right) \\ &= \frac{1}{2}\mathbb{E}\left[P(y_i = y_l)2D_{intra}(I^{pre}) + P(y_i \neq y_l)2D_{inter}(I^{pre})\right] \\ &= PD_{intra}(I^{pre}) + (1-P)D_{inter}(I^{pre}), \end{aligned} \tag{12}$$

where y_i is the label of an image $\boldsymbol{x}_i$ assigned by the classifier trained on pre-D, and $\boldsymbol{f}' = f(\boldsymbol{x}', \theta)$. 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 Liu et al. (2020). Mathematically, it can be formulated as

$$\psi(\phi_2^{-1}(I^{pre})) > \psi(\phi_1^{-1}(I^{pre})), \text{ if } \phi_2^{-1}(I^{pre}) > \phi_1^{-1}(I^{pre}). \tag{13}$$

By substituting $D_{intra}(I^{eval}) = PD_{intra}(I^{pre}) + (1-P)D_{inter}(I^{pre})$ (Eq. 12) into the discriminative ratio inequality $\phi_2(I^{eval}) < \phi_1(I^{eval})$ given $\phi_2(I^{pre}) > \phi_1(I^{pre})$, we have

$$\phi_2(I^{eval}) < \phi_1(I^{eval}) \tag{14}$$

$$\iff \frac{D^2_{inter}(I^{eval})}{D^2_{intra}(I^{eval})} < \frac{D^1_{inter}(I^{eval})}{D^1_{intra}(I^{eval})} \tag{15}$$

$$\iff \frac{D^2_{inter}(I^{eval})}{PD^2_{intra}(I^{pre}) + (1-P)D^2_{inter}(I^{pre})} < \frac{D^1_{inter}(I^{eval})}{PD^1_{intra}(I^{pre}) + (1-P)D^1_{inter}(I^{pre})}, \tag{16}$$

$$\iff P < \frac{\frac{D^1_{inter}(I^{eval})}{D^1_{inter}(I^{pre})} - \frac{D^2_{inter}(I^{eval})}{D^2_{inter}(I^{pre})}}{\frac{D^1_{inter}(I^{eval})}{D^1_{inter}(I^{pre})} \cdot \left(1 - \frac{D^2_{intra}(I^{pre})}{D^2_{inter}(I^{pre})}\right) - \frac{D^2_{inter}(I^{eval})}{D^2_{inter}(I^{pre})} \cdot \left(1 - \frac{D^1_{intra}(I^{pre})}{D^1_{inter}(I^{pre})}\right)}, \tag{17}$$

$$\iff P < \frac{\psi(\phi_1^{-1}(I^{pre})) - \psi(\phi_2^{-1}(I^{pre}))}{\psi(\phi_1^{-1}(I^{pre}))\left(1 - \phi_2^{-1}(I^{pre})\right) - \psi(\phi_2^{-1}(I^{pre}))\left(1 - \phi_1^{-1}(I^{pre})\right)}, \tag{18}$$

$$\iff P < \frac{1}{1 - \phi_1^{-1}(I^{pre}) + \frac{\phi_2^{-1}(I^{pre}) - \phi_1^{-1}(I^{pre})}{\psi(\phi_2^{-1}(I^{pre})) - \psi(\phi_1^{-1}(I^{pre}))}\psi(\phi_1^{-1}(I^{pre}))}, \tag{19}$$

$$\iff P < \frac{1}{1 - \phi_1^{-1}(I^{pre}) + r\psi(\phi_1^{-1}(I^{pre}))}, \tag{20}$$

$$\iff r\psi(\phi_1^{-1}(I^{pre})) - \phi_1^{-1}(I^{pre}) < P^{-1} - 1, \tag{21}$$

$$\iff \frac{d\phi_1^{-1}(I^{pre})}{d\psi(\phi_1^{-1}(I^{pre}))}\psi(\phi_1^{-1}(I^{pre})) - \phi_1^{-1}(I^{pre}) < P^{-1} - 1, \tag{22}$$

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

where

$$r = \frac{\phi_2^{-1}(I^{pre}) - \phi_1^{-1}(I^{pre})}{\psi(\phi_2^{-1}(I^{pre})) - \psi(\phi_1^{-1}(I^{pre}))} \tag{24}$$

$$\approx \frac{d\phi^{-1}(I^{pre})}{d\psi(\phi^{-1}(I^{pre}))}, \text{ when } \phi_2^{-1}(I^{pre}) - \phi_1^{-1}(I^{pre}) \to 0. \tag{25}$$

We take integration of Eq. 23 as

$$\iff \int_0^{\phi^{-1}(I^{pre})} \frac{d\phi^{-1}(I^{pre})}{P^{-1}-1+\phi^{-1}(I^{pre})} < \int_{\psi(0)}^{\psi(\phi^{-1}(I^{pre}))} \frac{1}{\psi(\phi^{-1}(I^{pre}))} d\psi(\phi^{-1}(I^{pre})), \tag{26}$$

$$\iff \ln\left[\phi^{-1}(I^{pre}) + P^{-1} - 1\right] < \ln\left[\psi(\phi^{-1}(I^{pre}))\right] + \ln\left(\frac{P^{-1}-1}{\psi(0)}\right), \tag{27}$$

$$\iff \phi^{-1}(I^{pre}) + P^{-1} - 1 < \psi(\phi^{-1}(I^{pre}))\frac{P^{-1}-1}{\psi(0)}, \tag{28}$$

$$\iff \phi^{-1}(I^{pre}) < 1 - P^{-1} + \psi(\phi^{-1}(I^{pre}))\frac{P^{-1}-1}{\psi(0)}, \tag{29}$$

$$\iff \phi^{-1}(I^{pre}) < (\frac{\psi(\phi^{-1}(I^{pre}))}{\psi(0)} - 1)(P^{-1}-1) \tag{30}$$

$$\iff \phi(I^{pre}) > t \tag{31}$$

where the threshold t is defined as

$$t = \left[(\frac{\psi(\phi^{-1}(I^{pre}))}{\psi(0)} - 1)(P^{-1}-1)\right]^{-1}. \tag{32}$$

According to Formulation 13, $\psi(\phi^{-1}(I^{pre})) > \psi(0)$ 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. □

G.2 Analysis of P

In the following, we explain how P in Equation 12 can be theoretically computed, and how P negatively relates to the feature distribution distance briefly.

G.2.1 Computational Method of P

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

$$P_{jk} = \frac{1}{|I_j^{eval}|} \sum_{(\boldsymbol{x}_i, y_i) \in I_j^{eval}} \frac{\exp(\boldsymbol{w}_k \cdot f(\boldsymbol{x}; \boldsymbol{\theta}))}{\sum_{k'=1}^{C^{pre}} \exp(\boldsymbol{w}_{k'} \cdot f(\boldsymbol{x}; \boldsymbol{\theta}))}, \tag{33}$$

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

$$P_j = \sum_{k=1}^{C^{pre}} P_{jk}^2. \tag{34}$$

The average probability of P_j is

$$P = \frac{1}{C^{eval}} \sum_{j=1}^{C^{eval}} P_j. \tag{35}$$

G.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,

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

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

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

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

$$= 1, \tag{40}$$

where Inequality 38 is derived by Cauchy Schwarz Inequality (Wu & Wu, 2009), 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,

$$P = \frac{1}{C^{eval}} \sum_{j=1}^{C^{eval}} P_j \tag{41}$$

$$= \frac{1}{C^{eval}} \sum_{j=1}^{C^{eval}} \sum_{k=1}^{C^{pre}} P_{jk}^2 \tag{42}$$

$$\geq \frac{1}{C^{eval}} \sum_{j=1}^{C^{eval}} \frac{1}{C^{pre}} \left(\sum_{k=1}^{C^{pre}} P_{jk} \right)^2 \tag{43}$$

$$= \frac{1}{C^{eval}} \sum_{j=1}^{C^{eval}} \frac{1}{C^{pre}} \tag{44}$$

$$= \frac{1}{C^{pre}}, \tag{45}$$

where Inequality 43 is derived by Fundamental Inequality (Beckenbach et al., 1961), 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 $(\boldsymbol{x}_m, y'_m)$ and $(\boldsymbol{x}_n, y'_n)$ 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 $(\boldsymbol{x}_m, y'_m)$ and $(\boldsymbol{x}_n, y'_n)$ 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.

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 includes random-crop (224x224), color-jitter, 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 UNSEEN CLASS GENERALIZATION TASK

In unseen generalization task, we divide ImageNet-1K into two semantically exclusive datasets following the hierarchical structure built in WordNet (Fellbaum, 2000) - 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 the other 652 classes in ImageNet-1K. All the networks are pretrained on pre-D, and then examined by linear evaluation protocal on eval-D. As in (van den Oord et al., 2018; Kolesnikov et al., 2019; Chen et al., 2020), 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 mentioned in Islam et al. (2021), we use 12 datasets from different domains to evaluate the transferability of different methods, including natural (Mohanty et al., 2016; Nilsback & Zisserman, 2008; Olsen et al., 2019), satellite (Helber et al., 2019; Cheng et al., 2017), symbolic (Lake et al., 2015; Netzer et al., 2011), illustrative (Tian et al., 2020; Wang et al., 2019), medical (Codella et al., 2019; Wang et al., 2017), and texture (Cimpoi et al., 2014). The statistics of datasets are illustrated in Table 8.

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. We split the training set on 70% training and 30% 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.

H.4 OBJECT DETECTION AND INSTANCE SEGMENTATION

For object detection and instance segmentation, we train Mask-RCNN (He et al., 2017) (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 (Wu et al., 2019).

Table 8: 12-domains datasets used for downstream image classification.

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 9: 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.

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

I PRETRAIN RESULTS ON PRE-D

We also provide the top-1 accuracy of SL-MLP on pre-D in Table 9. 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 Figure 15. As backbones and classifiers are jointly trained during pretraining, classifiers 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 (Zhao et al., 2021) 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.

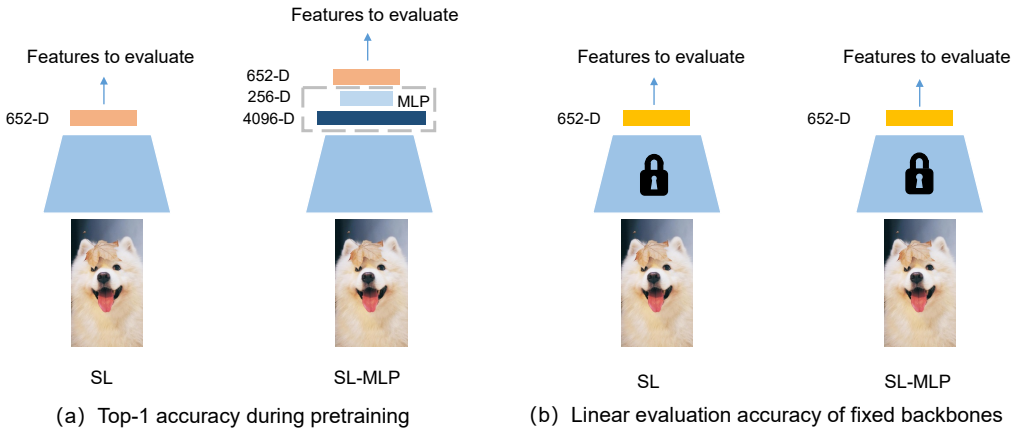

Figure 15: Evaluation of features extracted by SL and SL-MLP. **(a)**: During pretraining, features after the classifier is used to evaluate the accuracy on pre-D. **(b)**: After pretraining, we use the fixed backbones from different epochs to evaluate the performance of SL and SL-MLP.