EMPIRICAL STUDIES ON THE CONVERGENCE OF FEATURE SPACES IN DEEP LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

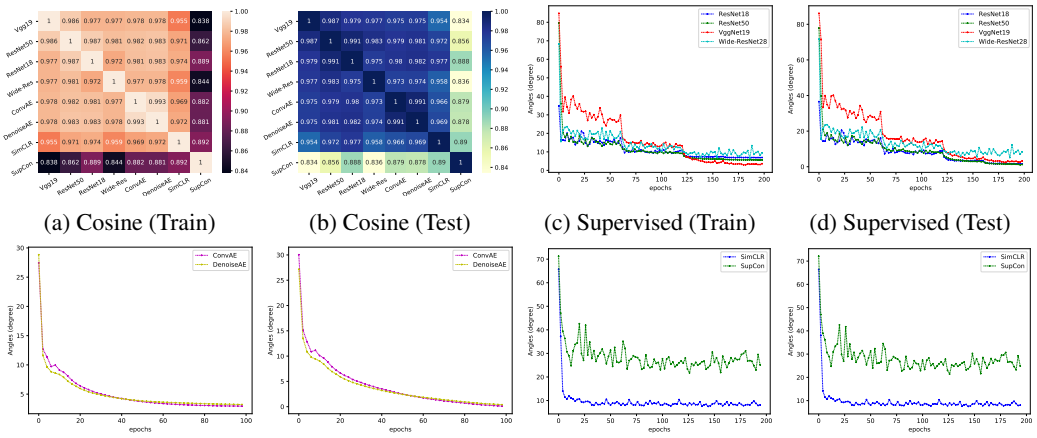
While deep learning is effective to learn features/representations from data, the distributions of samples in feature spaces learned by various architectures for different training tasks (e.g., latent layers of AEs and feature vectors in CNN classifiers) have not been well-studied or compared. We hypothesize that the feature spaces of networks trained by various architectures (AEs or CNNs) and tasks (supervised, unsupervised, or self-supervised learning) share some common subspaces, no matter what types of DNN architectures or whether the labels have been used in feature learning. To test our hypothesis, through Singular Value Decomposition (SVD) of feature vectors, we demonstrate that one could linearly project the feature vectors of the same group of samples to a similar distribution, where the distribution is represented as the top left singular vector (i.e., principal subspace of feature vectors), namely $\mathcal{P}$-vector. We further assess the convergence of feature space learning using angles between $\mathcal{P}$-vectors obtained from the well-trained model and its checkpoint per epoch during the learning procedure, where a quasi-monotonic trend of convergence to small angles has been observed. Finally, we carry out case studies to connect $\mathcal{P}$-vectors to the data distribution, and generalization performance. Extensive experiments with practically-used MLP, AE and CNN architectures for classification, image reconstruction, and self-supervised learning tasks on MNIST, CIFAR-10 and CIFAR-100 datasets have been done to support our claims with solid evidences.

1 INTRODUCTION

Blessed by the capacities of feature learning, deep neural networks (LeCun et al., 2015) have been widely used to perform learning tasks, ranging from classification, to generation (Goodfellow et al., 2014; Radford et al., 2015), in various settings (e.g., supervised, unsupervised, and self-supervised learning). To better euclid the features learned by deep models, numerous works have studied on interpreting the features spaces of the well-trained models (Simonyan et al., 2013; White, 2016; Zhu et al., 2016; Bau et al., 2017; 2019; Jahanian et al., 2020; Zhang & Wu, 2020).

Invariance beyond the use of architectures and labels. While existing studies primarily focus on the interpolation of a given model to discover mappings from the feature space to outputs of the model (e.g., classification (Bau et al., 2017) and generation (Jahanian et al., 2020)), the work is so few that compares the feature spaces learned by deep models of varying architectures (e.g., MLP/CNN classifiers versus Autoencoders) for different learning paradigms (Chen et al., 2020; Khosla et al., 2020; Spinner et al., 2018). More specifically, we are particularly interested in whether there exists certain *"statistical invariance"* in the feature space, no matter what type of architectures or whether label information (e.g., supervised vs. unsupervised vs. self-supervised (Chen et al., 2020) learning) are used in feature learning with the same training dataset.

Hypotheses. It is not difficult to imagine that the feature spaces of well-trained DNN classifiers in supervised learning setting might share some linear subspace (Vaswani et al., 2018). When models are well fitted to the same training set, the feature vectors of training samples should be projected to the ground-truth labels after a Fully-Connected Layer (i.e., a linear transform), while such linear subspace are supposed to distribute samples in a discriminative manner. We doubt that such subspace might be not only shared by supervised learners but also with AEs which are trained to



(a) Cosine (Train) (b) Cosine (Test) (c) Supervised (Train) (d) Supervised (Test)

(e) Unsupervised (Train) (f) Unsupervised (Test) (g) Self-Sup. (Train) (h) Self-Sup. (Test)

Figure 1: **Convergence to the Common Feature Subspace with CIFAR-10.** Figure 1. (a)–(b) present cosine (in the range of [0,1]) of angles between the $\mathcal{P}$-vectors of well-trained models of various architectures under different learning paradigms, using training and testing datasets respectively. *A well-trained model here is the one trained under the suggested settings after 200 epochs for supervised/self-supervised CNN classifiers and 100 epochs for unsupervised AEs.* Figure 1. (c)–(h) present angles of $\mathcal{P}$-vectors between the well-trained model and its checkpoint per training epoch of three learning paradigms, where the trends of convergence has been observed in all models, no matter whether the feature extractors of these models are trained with / without labels. More discussion are provided in Section 4.

reconstruct input data without any label information in an unsupervised manner, or even shared with self-supervised DNN classifiers (e.g., SimCLR (Chen et al., 2020)) which train (1) CNN feature extractor (using contrastive loss without labels) and (2) linear classifiers (using discriminative loss based on labels) separately in an ad-hoc manner. More specifically, we hypothesize that (H.I:) there exists certain *common feature subspaces* shared by well-trained deep models using the same training dataset, even though the architectures (MLPs, CNNs, and AEs) and the learning paradigms (supervised, unsupervised, and self-supervised) are significantly different. Further, as the training procedure usually initializes the DNN models from random weights and learns features from the training set step-by-step, we hypothesize that (H.II:) the training procedure gradually shapes the feature subspace over training iterations and asymptotically converge to the common subspace in certain statistical measure. Finally, we hypothesize that (H.III) the convergence to the common feature subspace would connect to the data distribution and performance of models, as such behavior indicates how well the features are learned from data.

Contributions. To test above three hypotheses, this work makes contributions in proposing new measures to the DNN features, namely $\mathcal{P}$-vectors, and conducting extensive experiments for empirical studies. We train deep models using various DNN architectures, multiple learning paradigms, and datasets, with the checkpoint restored per epoch. Then, we extract the feature vectors for either training or testing sample sets, from the model (Please see Section 3 for details) and discover some interesting relationships or associations as discussed below.

I. $\mathcal{P}$-vector and Convergence: Given the matrix of feature vectors (#samples[1] $\times$#features) for either training or testing samples, we perform the singular value decomposition (SVD) to obtain *left* and *right* singular vectors, characterizing the subspaces that samples distribute and the projection of features to subspaces respectively. We observe that deep models well-trained using the same dataset share similar *top left-singular vectors* (referring to the principal subspace of feature vectors), namely $\mathcal{P}$-vector in this study. For example, Figure 1 (a)–(b). show the high cosine similarity (close to 1) between the $\mathcal{P}$-vectors of any two well-trained DNNs for eight different deep learning architectures/tasks. This observation well backups H.I, where the common subspace in feature spaces shared by models trained with different architectures/paradigms has been observed. With checkpoints obtained along the training procedure, we estimate the angle that compares $\mathcal{P}$-vectors of the checkpoint

[1]We follow the convenience that denotes # as the term "the number of" for short.

per epoch and the well-trained one. We find such angle decrease over the number of epochs (in an overall manner) and would converge to a small one. For example, Figure 1 (c)–(h). demonstrate the consistent converging trends of angles between $\mathcal{P}$-vectors of the well-trained model and its checkpoints in progress of training for eight different deep learning architectures/tasks. This observation supports H.II, where the convergence to the common feature subspace has been observed.

II. Data Distribution and Performance: To connect to the data distributions, we intend to compare feature vectors and original data vectors of samples using $\mathcal{P}$-vectors. In addition to the $\mathcal{P}$-vector estimated from the feature vectors of a deep model (namely a "model $\mathcal{P}$-vector"), We simply form the sample vectors (either for training or testing) into a #samples$\times$#data_dimension matrix and perform SVD to obtain the top left singular vector as the $\mathcal{P}$-vector of samples (namely a "data $\mathcal{P}$-vector"). We estimate the angles between the model and data $\mathcal{P}$-vectors, and find a trend of convergence (from nearly orthogonal to relatively small angles), where we can see the well-trained models would incorporate smaller angles than ones in the early stage of training processes. We further correlate such angles with training and testing accuracy of the models, where we observe significant negative correlations in most cases of experiments. The trends show that the model with a smaller angle between the model and data $\mathcal{P}$-vectors would enjoy better performance. The evidences backup H.III.

2 Related Work

In this section, we first present the preliminaries in understanding feature learning of DNNs, then discuss the most relevant works to our studies.

As early as 2013, (Simonyan et al., 2013) proposed to visualize the features learned by deep convolutional neural networks (CNN) and made sense of discriminative learning via deep feature extraction. For generative models, (White, 2016; Zhu et al., 2016) studied the interpolation of latent spaces while (Zhu et al., 2016) discovered an user-controlled way to manipulate the images generated through the surrogation of latent spaces via manifolds. Later, (Bau et al., 2017) presented the visual concepts learned in the feature spaces of discriminative models through network dissection on specific datasets while the same group of researchers also proposed GAN dissection (Bau et al., 2019) – an interactive way to manipulate the semantics and style of image synthesis. (Richardson & Weiss, 2018) compared GAN and Gaussian Mixture Models (GMMs) to understand the capacity of distribution learning in GAN. (Berthelot et al., 2019) proposed to improve understanding and interpolation of Autoencoders using adversarial regularizer while (Spinner et al., 2018) compared AEs with its variational derivatives to interpret the latent spaces. More recently, (Saxe et al., 2019) mathematically analyzed the process of neural representation construction from the perspectives of learning dynamics of deep neural networks. (Jahanian et al., 2020) studied the "steerability" of GAN, where it discovered point-to-point editing paths for content/style manipulation. (Zhang & Wu, 2020) uncovered the phenomena that DNN classifiers with piecewise linear activation tend to map the input data to linear subregions. Other impressive studies in this line of research include (Nguyen et al., 2016; Arvanitidis et al., 2018; Sercu et al., 2019).

Discussion. The most relevant studies to our work are (Bau et al., 2017; Zhang & Wu, 2020; Saxe et al., 2019; Lee et al., 2019). For discriminative models, (Bau et al., 2017) recovered visual features learned by CNN classifiers with a priorly labeled dataset, and quantified then compared the feature learning capacities (namely "interpretability" in the work) of different DNN models through patterns matching with the ground truth. Compared to (Bau et al., 2017), we carry out the empirical studies on a wide range of datasets without any prior information on their features and observe consistent phenomena in the distribution of samples in the feature spaces. Furthermore, while (Zhang & Wu, 2020) studied the properties of regions where a supervised DNN classifier with piecewise linear activation behaves linearly, our work observes the common linear subspaces shared by the features learned by the networks that are trained with different architectures (e.g., MLP/CNN classifiers and AEs with ReLU activation) and paradigms (e.g., supervised, unsupervised, and self-supervised learning). Furthermore, both our work and (Saxe et al., 2019) propose to compare feature/representation learning by various models through SVD, while we perform SVD to investigate the distribution of samples in feature space and (Saxe et al., 2019) uncovered the latent structures in input-and-output of neurons.

To understand the latent space of generative models, some statistical/mathematical tools (Richardson & Weiss, 2018; Lee et al., 2019) have been introduced. Through comparisons between GAN and

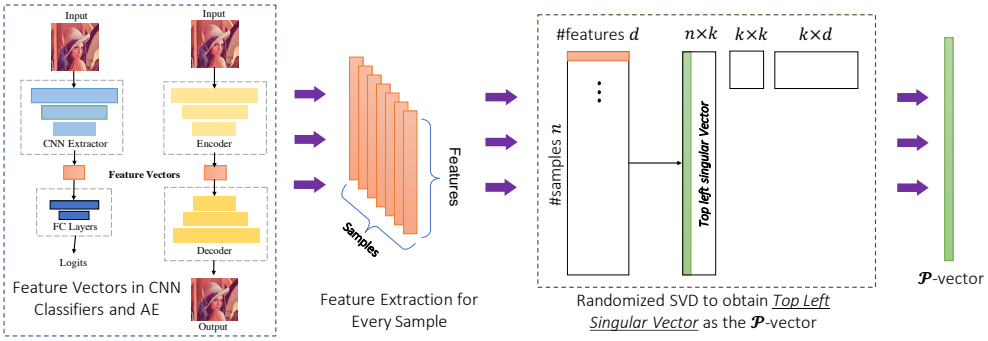


Figure 2: Obtain $\mathcal{P}$-vectors of deep neural networks using a group of samples

GMM, (Richardson & Weiss, 2018) uncovered GAN's superiority in feature/texture reconstruction and in the meanwhile its incompetence in distribution learning, while (Lee et al., 2019) provides some analytical insights on the structure of latent spaces. In our work, we propose $\mathcal{P}$-vectors to model the distribution of samples in the feature/latent spaces of DNN classifiers and AEs. To the best of our knowledge, we make unique contributions compared to the above work.

3 Methodologies: Feature Vectors, Singular Value Decomposition, and $\mathcal{P}$-vectors

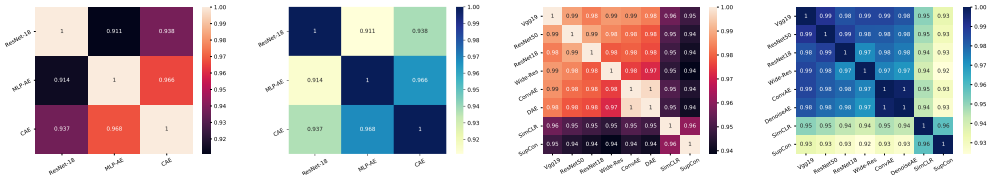
As mentioned in Section. 1, we carry out extensive experiments on analyzing the feature vectors of DNN models with various architectures amd different training paradigms through the newly-proposed $\mathcal{P}$-vectors. In this section, we present the design of experiments in details, where we cover three key procedures of the experiment as shown in Figure 2.

Feature Vector Extraction. Given a model, either the well-trained one or a checkpoint obtained during the training process, we extract the feature vector for every sample, with respect to the architectures. For DNN classifiers (either under supervised/self-supervised learning), we use the output of CNN feature extractor (i.e., the input to the Fully-Connected Layer) as the feature vector of the given sample, while we vectorize the output bottleneck layer as the feature vector for AEs. Note that, in our research, we consider AEs with symmetric architectures of encoders and decoders only.

Singular Value Decomposition with Feature Vectors. Given the feature vector for every sample, we form a #samples$\times$#features matrix and perform SVD to obtain the top left singular vector as the $\mathcal{P}$-vector. Furthermore, there is no need to solve the singular vectors of the complete spectrum, as only the top singular vector is requested for $\mathcal{P}$-vector estimation. In this way, we propose to use Randomized SVD (Halko et al., 2011) that compresses the feature domain and approximate the low-rank structure of SVD for acceleration purpose. Actually, we compare the numerical solution of Randomized SVD and Common SVD for ResNet-50 on CIFAR-10 dataset (#features=256 and #samples= 50,000), where we need to perform SVD on a #samples$\times$#features matrix and the $\mathcal{P}$-vector/top left singular vector should be with dimensions. Compared to the vanilla SVD, around 109x (from 44.88 seconds to 0.41 seconds) speedup has been achieved by Randomized SVD while no significant numerical errors having been found in the results.

$\mathcal{P}$-vectors, Principal Subspace, and the Distribution of Samples in Feature Spaces. The top left singular vector of #samples$\times$#feature matrix represents the *principal subspace* (JOLLIFFE, 2002) that samples are distributed into a one-dimensional space (with the maximal variances) by the projection of principal component (Abdi & Williams, 2010) of features. In this way, we can compare and measure the divergence between the feature spaces of two DNNs through assessing the angle between their $\mathcal{P}$-vectors based on the same set of samples.

Discussion. Given the same set of samples, $\mathcal{P}$-vectors of the two DNNs should be in the equal length, as they are both the top singular vectors in the sample side. Furthermore, we can easily measure their divergence via Cosine of two vectors. A larger cosine of two vectors (e.g., close to 1.0) usually refers to the evidence that the two networks share subspace in their features. Note that,



(a) MNIST (Train) (b) MNIST (Test) (c) CIFAR-100 (Train) (d) CIFAR-100 (Test)

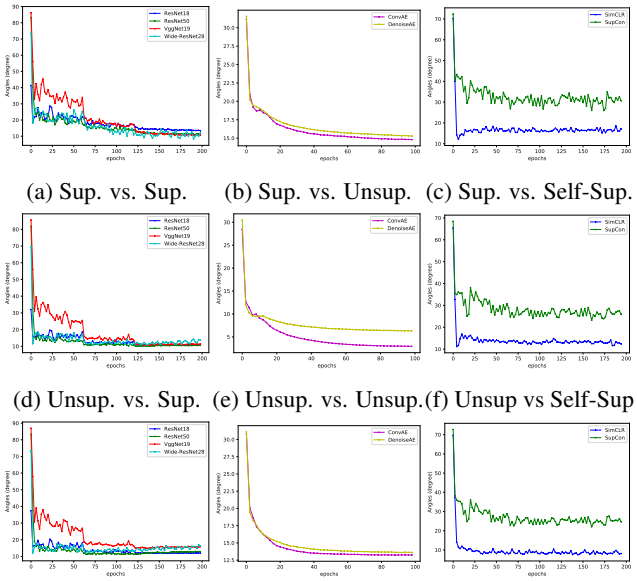
Figure 3: Cosine of angles between the $\mathcal{P}$-vectors of well-trained models

in high-dimensional spaces, the chance of orthogonality between two random vectors appears more frequently, due to the curse of dimensionality (Pestov, 1999). Thus, given a sample set such as CIFAR-10 with more than 60,000 samples, when the cosine measure close to 1.0 or the angle between the two $\mathcal{P}$-vectors (with 60,000 dimensions) is small, we can conclude that the two networks would share a subspace in the feature spaces in high confidence.

4 Uncovering Common Feature Subspace with $\mathcal{P}$-vectors

In this section, we further elaborate the phenomenons that we introduced in Figure 1 of Section 1, where we would incorporate more details using rich datasets.

Common Subspaces. To demonstrate the existence of common subspaces, we propose to measure the angles between the $\mathcal{P}$-vectors extracted from well-trained models using the same datasets. For any two models, a small angle between the two $\mathcal{P}$-vectors refers to the facts that the two models would project these samples to similar principal subspaces in feature spaces, as the $\mathcal{P}$-vectors are estimated as the top left singular vector of the feature matrix (i.e., #samples×#features) SVD based on the same group of samples. Figure. 1 (a)–(b) presents the cosine of angles between the $\mathcal{P}$-vectors of well-trained models (trained with various architectures and tasks using CIFAR-10 dataset). To generalize our observations, we propose to compare the $\mathcal{P}$-vectors of models using the feature matrices based on the training and testing samples in Figures. 1 (a) and (b) respectively. We carry out the same experiments on CIFAR-10 and CIFAR-100 datasets, and present the cosine of angles between the $\mathcal{P}$-vectors of well-trained models in Figure 3. All experiments demonstrate that, with a relatively large cosine (close to 1), the models well-trained using the same dataset, no matter what types of architectures or whether labels has been used in the feature learning of various tasks, share a common subspace.



(a) Sup. vs. Sup. (b) Sup. vs. Unsup. (c) Sup. vs. Self-Sup.

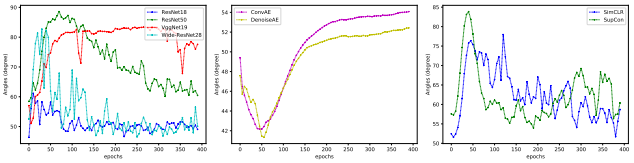
(d) Unsup. vs. Sup. (e) Unsup. vs. Unsup. (f) Unsup vs Self-Sup

(g) Self. vs Sup. (h) Self. vs Unsup. (i) Self. vs Self-Sup.

Figure 4: Convergence of $\mathcal{P}$-vector angles between checkpoint per epoch and well-trained models using CIFAR-10.

Convergence to the Common Subspace To understand the dynamics of feature learning that shapes the common subspace from scratch, we measure the change of angles over epochs between the $\mathcal{P}$-vectors of training models (i.e., checkpoint in the epoch) and well-trained ones in comparisons. In Figure 1 (c)–(h), for every model of various tasks, we present angles between the $\mathcal{P}$-vectors of the well-trained model and itself's training checkpoint per epoch. A consistent convergence could be observed. As the $\mathcal{P}$-vectors of well-trained models are close to each others (see in Figures 1 and 3), we can conclude the feature vectors extracted by these models would evolve on time and gradually converge to share the common subspace during the learning procedure.

We carry out a more comprehensive model-to-model comparison using CIFAR-10 dataset. Figures. 4 (a)–(c) present the convergence of $\mathcal{P}$-vector angles between the well-trained supervised models and the checkpoints per epoch of supervised, unsupervised, and self-supervised learning models , where we use the well-trained Wide-ResNet28 (trained with 200 epochs under suggest settings) as the reference of supervised models. Figures. 4 (d)–(f) present $\mathcal{P}$-vector angles between the well-trained ConvAE (trained with 100 epochs under suggest settings as the reference of unsupervised learning) and the checkpoints per epoch of supervised, unsupervised, and self-supervised learning models. Figures. 4 (g)–(i) present the $\mathcal{P}$-vector angles between the well-trained SimCLR representations (trained under suggest settings (Chen et al., 2020) as the reference of self-supervised learning) and the checkpoints per epoch of supervised, unsupervised, and self-supervised learning models. The trends of convergence in all comparison further validate our hypotheses.



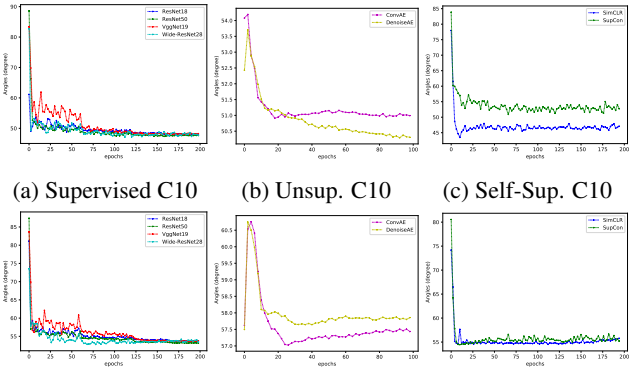
(a) Supervised (b) Unsupervised (c) Self-Sup.

Figure 5: Angles between the $\mathcal{P}$-vectors of the training and well-trained models over the number of iterations in the first epoch using CIFAR-10.

Discussion. Note that, to better visualize every comparison, we use the maximal angles achieved during the first epoch to represent the angles between $\mathcal{P}$-vectors corresponding to the first epoch in Figures 1 and 4. Actually, in the first epoch, there would incorporate some non-monotonic trends for the angles varying over the number of iterations. Figure 5 present the angles between the $\mathcal{P}$-vectors of the training and well-trained models over the number of iterations in the first epoch in three settings of learning, where we use CIFAR-10 dataset for the experiments. In this way, we recover the procedure that the learning procedure shapes the feature space – *with the random weight initialization, the angles of $\mathcal{P}$-vectors between the initial models and the well-trained models are large, the angles drop down and rebound in a zigzag trend quickly in all three learning paradigms. In the rest of learning procedure, the angles between the $\mathcal{P}$-vectors drop down and converge to smaller ones over the number of iterations.* (**H.II**)

5 Case Studies on $\mathcal{P}$-vectors with Data Distributions and Model Performance

In this section, we intend to extend our observations to connect $\mathcal{P}$-vector with the (raw) data distributions and the generalization performance of models.



(a) Supervised C10 (b) Unsup. C10 (c) Self-Sup. C10

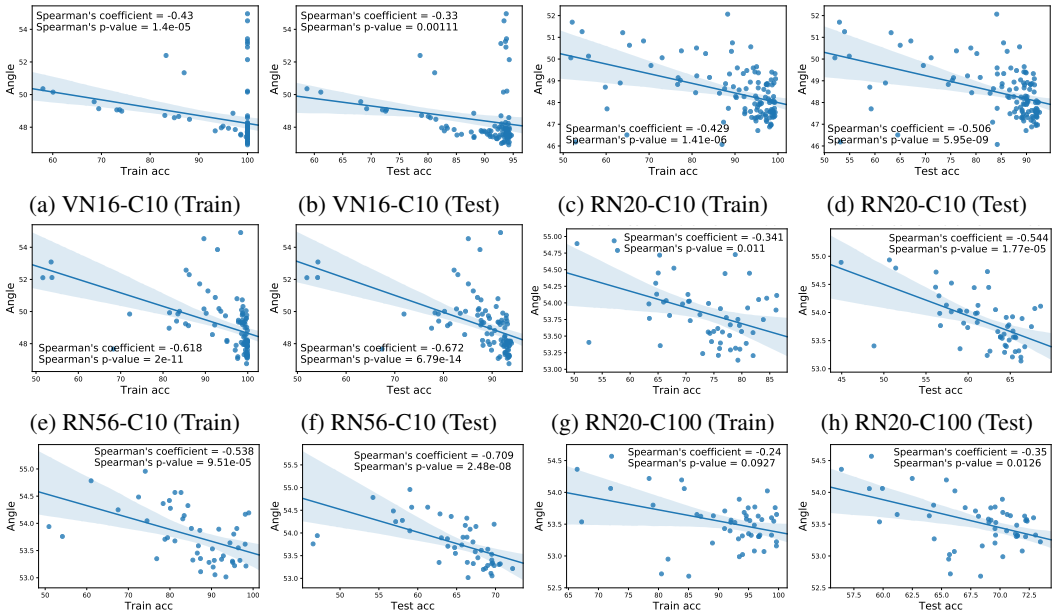
(d) Supervised C100 (e) Unsup. C100 (f) Self-Sup. C100

Figure 6: Convergence of angles between the model and data $\mathcal{P}$-vectors per training epoch using CIFAR-10 and CIFAR-100 datasets. C10: CIFAR-10, C100: CIFAR-100

Correlation to Data $\mathcal{P}$-vectors. Given the raw data matrix, i.e., a #samples$\times$#data dimensions matrix, we obtain the Data $\mathcal{P}$-vector[2] of these samples using the top left singular vector of the raw data matrix, which represents the principal subspace of data (or the position of every sample projected by the principal component of the data). To understand the connection between models and data, We carry out the case study using CIFAR-10 dataset, so as to measure the angles between the model $\mathcal{P}$-vector and the data $\mathcal{P}$-vector and how the angles change over the number of training epochs.

Figure.6 shows the convergence of $\mathcal{P}$-vector angles between the raw data and the checkpoints per epoch of supervised, unsupervised, and self-supervised learning models. The angles between the

[2]We use the term "model $\mathcal{P}$-vector" to represent the $\mathcal{P}$-vector estimated using the feature vectors of a deep model, while using "data $\mathcal{P}$-vector" as the top left singular vector of the raw data matrix.



(a) VN16-C10 (Train) (b) VN16-C10 (Test) (c) RN20-C10 (Train) (d) RN20-C10 (Test)

(e) RN56-C10 (Train) (f) RN56-C10 (Test) (g) RN20-C100 (Train) (h) RN20-C100 (Test)

(i) RN56-C100 (Train) (j) RN56-C100 (Test) (k) RN110-C100 (Train) (l) RN110-C100 (Test)

Figure 7: Strong and consistent correlations between the model performance (training and testing accuracy) and the angels between model and data $\mathcal{P}$-vectors using CIFAR-10 and CIFAR-100 datasets. VN: VggNet, RN: ResNet, C10: CIFAR-10, and C100: CIFAR-100.

data and model $\mathcal{P}$-vectors starts from about orthogonal and generally decrease and converge to about 50° degree to 60° degree. Note that the angle points corresponding to the first epoch on the curves are the largest $\mathcal{P}$-vector angles during the training procedure of the first epoch from the random scratch. The results indicate that the principal subspace of the well-trained models are more close to the principal subspace of the raw data, no matter what type of architectures are used for what kind of learning tasks and whether the labels are used for training.

Correlation to Deep Learning Performance. We further explore the relationship between the performance of the various models on different datasets and the angles between $\mathcal{P}$-vectors and data $\mathcal{P}$-vectors. Experiments are delivered using ResNet-20/56/110 and VGG-16 on CIFAR-10/100 datasets. As shown in Fig.7, there exists a strong and consistent correlations between the training/testing accuracy of models and the angles between the model and data $\mathcal{P}$-vectors. Note that in these experiments, we use the samples in the training dataset to estimate the model and data $\mathcal{P}$-vectors while avoiding the use of validation information, so as the demonstrate the connection of $\mathcal{P}$-vectors all based on training samples to the generalization performance. Note that for CIFAR-100 dataset, we use ResNets with pre-activation enabled.

We also conduct correlation analysis via Spearman's coefficients and p-values. With 0.05 as the threshold for significance, we find significance in the correlations between the angles between model and data $\mathcal{P}$-vectors and the training/testing accuracy for all above cases, expect the correlation between the angles and training accuracy on ResNet-100 using CIFAR-100 dataset (p-value=0.0927). Further for all models on all datasets, the correlation between angles and the testing accuracy is slightly higher than the correlations between angles and training accuracy (and significantly lower p-value). In this way, we could conclude that (1) both training and testing accuracy are correlated to the angles between the model and data $\mathcal{P}$-vectors, (2) the strong correlations between angles and the testing accuracy might not be caused by the correlations between the angles and training accuracy, as the earlier ones are even stronger, (3) the angles between the model and data $\mathcal{P}$-vectors would be a reasonable performance indicator, as they are strongly, consistently, and significantly correlated to the testing accuracy.

Applications to Generalization Performance Prediction. To further demonstrate the feasibility of using the $\mathcal{P}$-vector as a "validation-free" measure of generalization performance, we use the experiment settings of "Predicting Generalization in Deep Learning Competition" at NeurIPS 2020 to

Methods	Prediction Score
VC Dimension (Vapnik, 2013)	0.020
Jacobian norm w.r.t intermediate layers (Jiang et al.)	2.061
Angles between the model and data $\mathcal{P}$-vectors	**3.325**
Distance to Initialization (Nagarajan & Kolter, 2019)	4.921
Distance from Initialization + Angles between the model and data $\mathcal{P}$-vectors	**4.971**
Sharpness of the convergence point (Jiang et al., 2019)	10.667
Pseudo validation accuracy	13.531
Pseudo validation accuracy + Angles between the model and data $\mathcal{P}$-vectors	**15.618**

Table 1: Scores of different method to predict the generalization gap using CIFAR-10 and SHVN.

evaluate $\mathcal{P}$-vectors on predicting the generalization performance of a wide range of models using the training dataset. The competition offers a large number of deep models trained with various hyper-parameters and DNN architectures, while the official evaluator for the competition first predicts the generalization performance of every model using the proposed measure, then verify the prediction results through the mutual information (the higher the better) between the proposed measures and the (observable) ground truth of generalization gaps.

In the experiments, we propose to use the angles between the model and data $\mathcal{P}$-vectors using the training dataset as the metrics of generalization performance. In the comparisons with the proposed $\mathcal{P}$-vectors, we include several baseline measures in generalization performance predictors, including VC Dimension (Vapnik, 2013), Jacobian norm w.r.t intermediate layers (Jiang et al.), Distance from the convergence point to initialization (Nagarajan & Kolter, 2019), and the Sharpness of convergence point (Jiang et al., 2019). In addition to these methods, we also propose "Pseudo Validation Accuracy" as a measure for comparisons, where this measure first uses random data augmentation apply to the original set of training data to generate a set of "pseudo validation samples", then tests the accuracy of the model using "pseudo validation samples".

Table 1 presents the comparisons between the proposed measures and baselines. It shows that when the proposed measure – angles between the model and data $\mathcal{P}$-vectors – stands alone, the measures significantly outperforms the baseline methods including Jacobian norm w.r.t intermediate layers and the VC dimensions. However, through complementing with other metrics, the metrics based on $\mathcal{P}$-vector angles could be further improved in predicting the generalization performance and finally outperform all baseline methods when combining with "Pseudo validation accuracy". Note that we combine the results of two metrics through weighted aggregation (Pihur et al. (2009), with a constant weight 0.05) of two ranking lists that are sorted according to the two metrics respectively.

All above experiment results in three case studies backup our hypothesis that the $\mathcal{P}$-vectors of the model feature spaces would connect to the data distribution and performance of models (**H.III**).

6 Conclusion

In this work, we propose the $\mathcal{P}$-vector, defined as the top singular vector of the feature matrix (the #samples$\times$#features matrix) through SVD, to characterize the principal subspace of the feature space founded by the samples. We observe that, no matter what type of DNN architectures or whether the labels have been used to train the models, the $\mathcal{P}$-vectors of different models would converge to the same one (or the cosine measure of the $\mathcal{P}$-vector angle between any two models is close to 1), when the models are well trained with the same dataset. We conclude that the feature spaces all well-trained DNN models using the same training dataset would share principal subspace. Furthermore, during the training procedure from the random scratch, the model $\mathcal{P}$-vector would slowly approach to the data $\mathcal{P}$-vector (defined as the top singular vector of the raw data matrix, i.e., #samples$\times$#data dimensions), where the data and model $\mathcal{P}$-vectors start from an almost-orthogonal status while the angles between the data and model $\mathcal{P}$-vectors overall decrease and converge to smaller ones (50°–60°). Finally, we find that angles between the model and data $\mathcal{P}$-vectors are strongly correlated to the performance of models (i.e., both training/testing accuracy) while they are capable of predicting generalization performance, even when the model and data $\mathcal{P}$-vectors are all estimated using training dataset only.

REFERENCES

Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

Georgios Arvanitidis, Lars Kai Hansen, and Søren Hauberg. Latent space oddity: on the curvature of deep generative models. In *International Conference on Learning Representations*, 2018.

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Conference on Computer Vision and Pattern Recognition*, 2017.

David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B. Tenenbaum, William T. Freeman, and Antonio Torralba. Gan dissection: Visualizing and understanding generative adversarial networks. In *International Conference on Learning Representations*, 2019.

David Berthelot, Colin Raffel, Aurko Roy, and Ian Goodfellow. Understanding and improving interpolation in autoencoders via an adversarial regularizer. In *International Conference on Learning Representations*, 2019.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pp. 2672–2680, 2014.

Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53 (2):217–288, 2011.

Ali Jahanian, Lucy Chai, and Phillip Isola. On the "steerability" of generative adversarial networks. In *International Conference on Learning Representations*, 2020.

Yiding Jiang, Pierre Foret, Scott Yak, Behnam Neyshabur, Isabelle Guyon, Hossein Mobahi, Gintare Karolina, Daniel Roy, Suriya Gunasekar, and Samy Bengio. Predicting generalization in deep learning, competition at neurips 2020. URL `https://sites.google.com/view/pgdl2020/home?authuser=0`.

Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. *arXiv preprint arXiv:1912.02178*, 2019.

IT JOLLIFFE. Principal component analysis. *Springer Series in Statistics*, 2002.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*, 2020.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

Dennis Lee, Christian Szegedy, Markus N Rabe, Sarah M Loos, and Kshitij Bansal. Mathematical reasoning in latent space. *arXiv preprint arXiv:1909.11851*, 2019.

Vaishnavh Nagarajan and J Zico Kolter. Generalization in deep networks: The role of distance from initialization. *arXiv preprint arXiv:1901.01672*, 2019.

Anh Nguyen, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *Advances in Neural Information Processing Systems*, pp. 3387–3395, 2016.

Vladimir Pestov. On the geometry of similarity search: Dimensionality curse and concentration of measure. *arXiv preprint cs/9901004*, 1999.

Vasyl Pihur, Susmita Datta, and Somnath Datta. Rankaggreg, an r package for weighted rank aggregation. *BMC bioinformatics*, 10(1):62, 2009.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Eitan Richardson and Yair Weiss. On gans and gmms. In *Advances in Neural Information Processing Systems*, pp. 5847–5858, 2018.

Andrew M Saxe, James L McClelland, and Surya Ganguli. A mathematical theory of semantic development in deep neural networks. *Proceedings of the National Academy of Sciences*, 116 (23):11537–11546, 2019.

Tom Sercu, Sebastian Gehrmann, Hendrik Strobelt, Payel Das, Inkit Padhi, Cicero Dos Santos, Kahini Wadhawan, and Vijil Chenthamarakshan. Interactive visual exploration of latent space (ivels) for peptide auto-encoder model selection. In *ICLR 2019 DeepGenStruct Workshop*, 2019.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *ICLR Workshop*, 2013.

Thilo Spinner, Jonas Körner, Jochen Görtler, and Oliver Deussen. Towards an interpretable latent space: an intuitive comparison of autoencoders with variational autoencoders. In *IEEE VIS*, 2018.

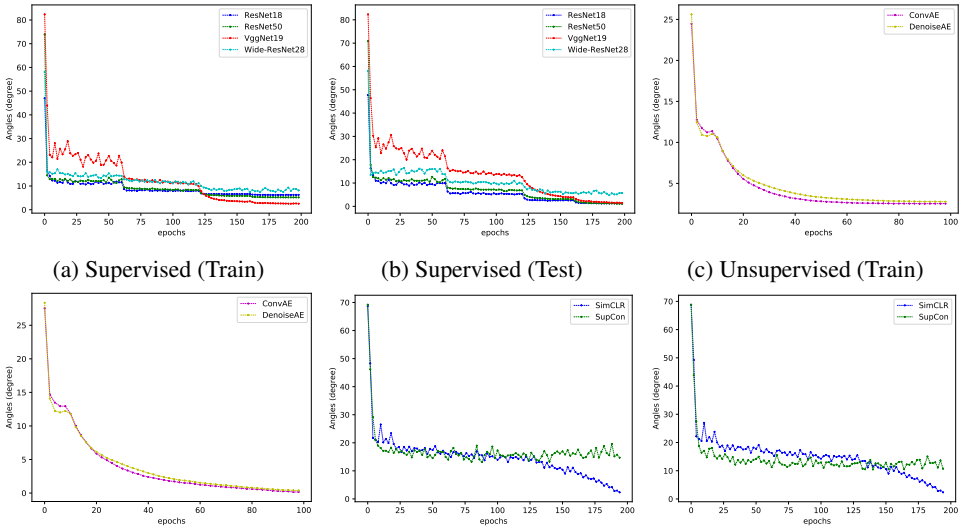
Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Springer Science & Business Media, 2013.

Namrata Vaswani, Thierry Bouwmans, Sajid Javed, and Praneeth Narayanamurthy. Robust subspace learning: Robust pca, robust subspace tracking, and robust subspace recovery. *IEEE Signal Processing Magazine*, 35(4):32–55, 2018.

Tom White. Sampling generative networks. *arXiv preprint arXiv:1609.04468*, 2016.

Xiao Zhang and Dongrui Wu. Empirical studies on the properties of linear regions in deep neural networks. In *International Conference on Learning Representations*, 2020.

Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. Generative visual manipulation on the natural image manifold. In *European conference on computer vision*, pp. 597–613. Springer, 2016.



(a) Supervised (Train) (b) Supervised (Test) (c) Unsupervised (Train)

(d) Unsupervised (Test) (e) Self-Sup. (Train) (f) Self-Sup. (Test)

Figure 8: Convergence to the Common Feature Subspace with CIFAR-100. Curves of angles of $\mathcal{P}$-vectors between the well-trained model and its checkpoint per training epoch of three learning supervisory manners. The trends of convergence for the angles can be observed in all models.

A Appendix

A.1 Comparison of angles between model and well-trained $\mathcal{P}$-vector Using Different Architectures on CIFAR-100

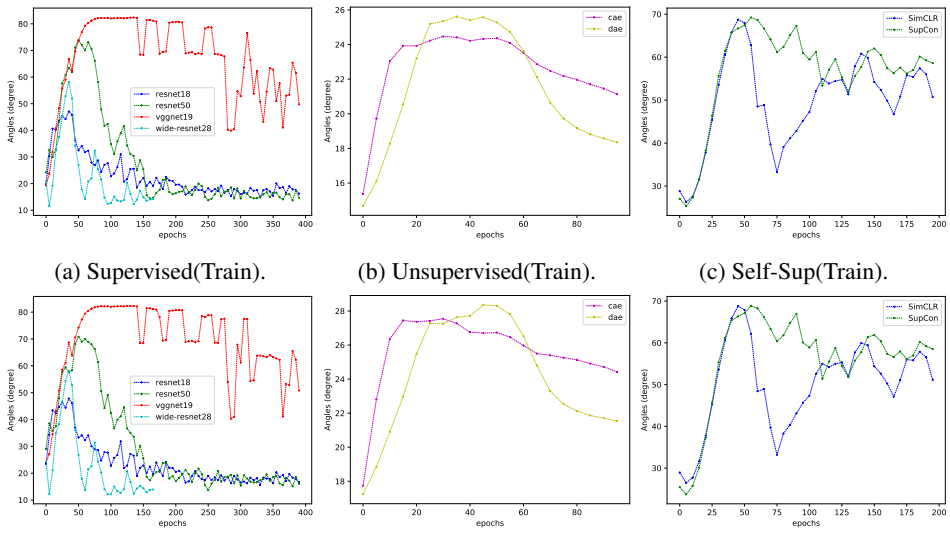
In the main text, we presented the result on CIFAR-10 dataset. To generalize the observations, we repeated the experiments on CIFAR-100 dataset to validate our hypothesis of the convergence of the angles between model checkpoints and well-trained model $\mathcal{P}$-vectors. We investigate the change of angles over the $\mathcal{P}$-vectors of training model checkpoints per epoch with comparison to the $\mathcal{P}$-vectors of well-trained models (model of epoch 200 in our case). As shown in Fig.8, a gradually decreasing manner of the curves for the angle between $\mathcal{P}$-vectors and all angles between $\mathcal{P}$-vectors cross models with different supervisory manners generally converge to a value that smaller than 10° degree. We can conclude that the hypothesis of the existence of common subspace during the learning procedure also stands on the experiments with CIFAR-100 dataset.

A.2 The Mon-monotonic Trend within the First Epoch

The variation of angles between $\mathcal{P}$-vectors for the well-trained model and its checkpoint per training epoch of each iteration in the first epoch. The non-monotonic trends within the first epoch also incorporate in the experiments on CIFAR-100 datasets. Fig.11 shows the curves indicating the variation of angles between the training model $\mathcal{P}$-vectors and the well-trained model $\mathcal{P}$-vectors in the iterations in the first epoch. As we use 128 as the batch size in training procedure, the number of iterations for updates is 391 per epoch. We obtain the observation of a non-monotonic trend that the angle first rises with the random initialization and drop down. And in the rest of training process, the angles keeps the approximately monotonically decreasing and converging to small values. The experiments shows consistent result and conclusion on CIFAR-100 dataset with the discussion in section 4.

A.3 Model-to-model Common Subspace

We also test and verify the model-to-model common subspace shared by models trained with different supervisory manners on CIFAR-100 dataset. Experiments carried out to evaluate the angles between $\mathcal{P}$-vectors for checkpoints of all models and $\mathcal{P}$-vectors for well-trained supervised, unsupervised and self-supervised models, where we use the well-trained Wide-ResNet28/Convolution



(a) Supervised(Train). (b) Unsupervised(Train). (c) Self-Sup(Train).

(d) Supervised(Test). (e) Unsupervised(Test). (f) Self-Sup(Test).

Figure 9: Angles between the $\mathcal{P}$-vectors of the training and well-trained models over the number of iterations in the first epoch using CIFAR-100.

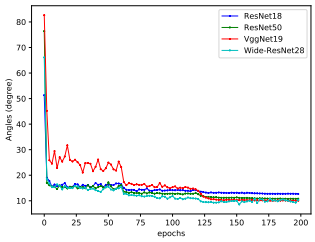
Auto-encoder and SimCLR model (trained with 200 epochs under suggest settings) as the reference of supervised, unsupervised and self-supervised models, respectively. As shown in Fig.10, a consistent convergence for the curves of the angles can be observed and support our hypothesis that the dynamics learning procedure construct the common subspace gradually.

A.4 The Non-monotonic Trend in the First Epoch of Comparison of Angles between Model and Raw Data $\mathcal{P}$-vectors

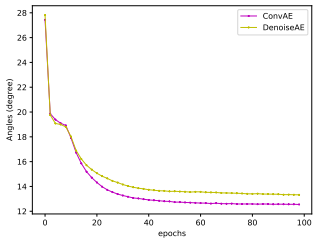
We also explore the construction procedure for the common subspace share between feature vectors and the raw data during the training process. Experiments are carried out to compare the space of models and raw data $\mathcal{P}$-vectors on the training dataset. We obtain the observation of a non-monotonic trend that the angle first rises with the random initialization and drop down. The angles keeps the approximately monotonically decreasing and converging to small values in following training epochs. The experiments shows consistent result on both CIFAR-10 and 100 dataset. Note that we follow the default random data augmentation policy to pre-process the training dataset.

A.5 Case Studies on the Angle Variety Between Model and Raw Data for Each Layer

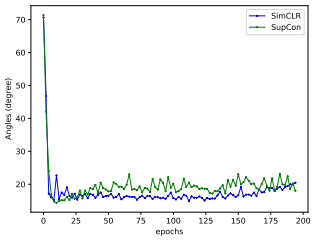
To explore the dynamic variation process by zooming in to every layer variation in each epoch, we perform the case study using Resnet-18 structure on CIFAR-10 dataset. As shown in Fig.12, we give the angles according to layers of 5 different epochs, where the x-axis indicating the indices of residual blocks in the network structure and y-axis refers to the angles between the model checkpoint and raw data $\mathcal{P}$-vectors. We observed that in early training stage, the angles between the model checkpoint and raw data $\mathcal{P}$-vectors keeps an increasing manner when the features passing though layers and turn into a decrease trend towards the stacked layers in the late training stage. This set of experiments further support our hypothesis that the dynamics learning procedure construct the common subspace gradually through training process.



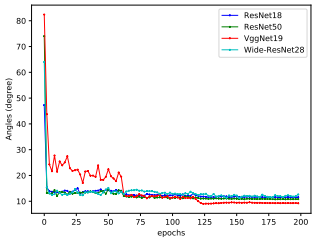
(a) Sup. vs. Sup.



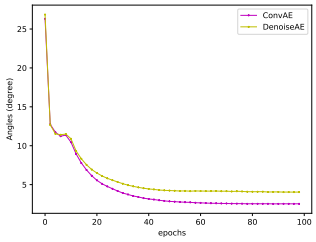
(b) Sup. vs. Unsup.



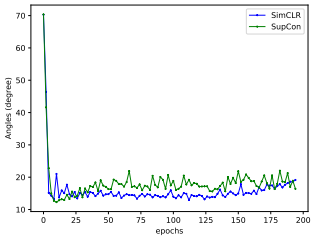
(c) Sup. vs. Self-Sup.



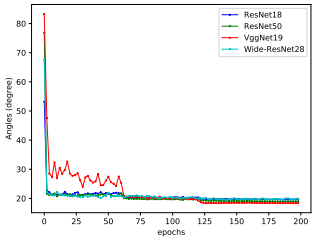
(d) Unsup. vs. Sup.



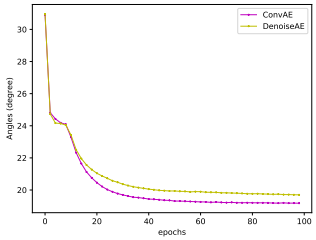
(e) Unsup. vs. Unsup.



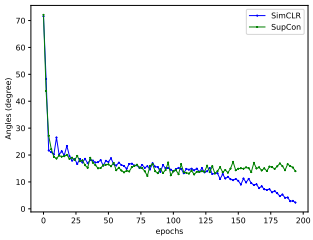
(f) Unsup vs Self-Sup



(g) Self. vs Sup.

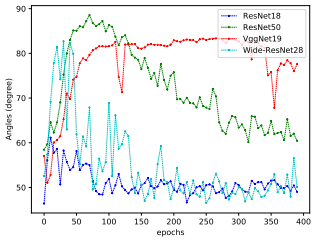


(h) Self. vs Unsup.

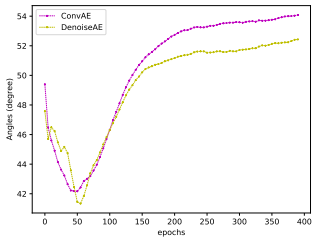


(i) Self. vs Self-Sup.

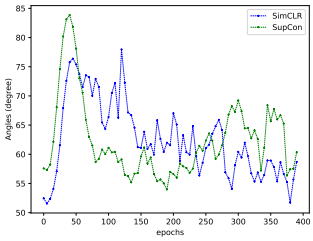
Figure 10: Convergence of $\mathcal{P}$-vector angles between checkpoint per epoch and well-trained models using CIFAR-100.



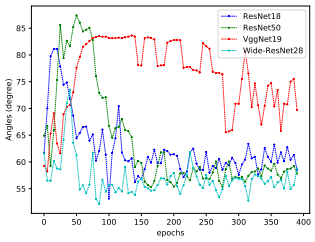
(a) Supervised C10



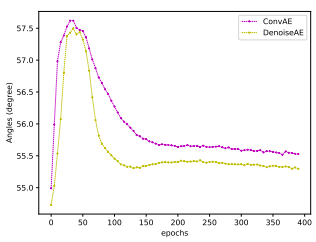
(b) Unsupervised C10



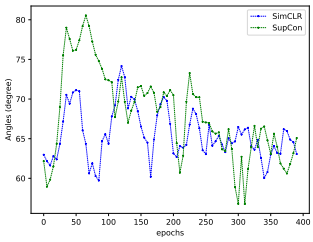
(c) Self-Sup C10.



(d) Supervised C100

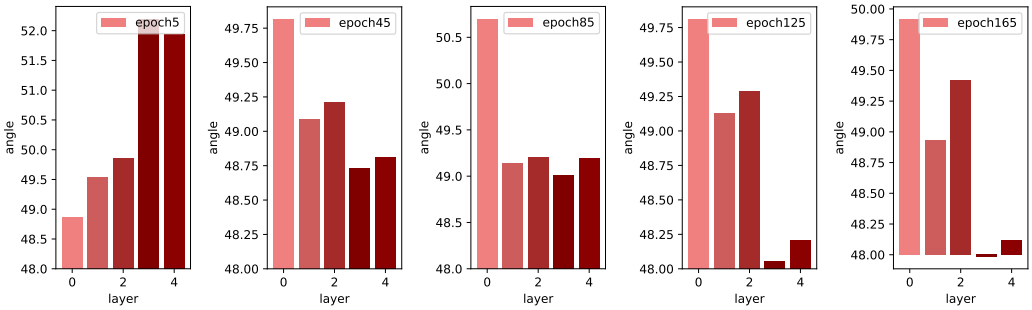


(e) Unsupervised C100



(f) Self-Sup C100.

Figure 11: Angles between the $\mathcal{P}$-vectors of the training and well-trained models over the number of iterations in the first epoch using CIFAR-10/100.



(a) Epoch 5. (b) Epoch45 (c) Epoch85. (d) Epoch125. (e) Epoch165.

Figure 12: Angles changes through layer between the $\mathcal{P}$-vectors of the training and raw data over the number of iterations in the first epoch using CIFAR-10.