Statistical Characteristics of Deep Representations: An Empirical Investigation

Daeyoung Choi*, Kyungeun Lee*, Duhun Hwang, and Wonjong Rhee
Department of Transdisciplinary Studies
Seoul National University
Seoul, 08826, South Korea
{choid, ruddms0415, yelobean, wrhee}@snu.ac.kr

Abstract

In this study, the effects of eight representation regularization methods are investigated, including two newly developed rank regularizers (RR). The investigation shows that the statistical characteristics of representations such as correlation, sparsity, and rank can be manipulated as intended, during training. Furthermore, it is possible to improve the baseline performance simply by trying all the representation regularizers and fine-tuning the strength of their effects. In contrast to performance improvement, no consistent relationship between performance and statistical characteristics was observable. The results indicate that manipulation of statistical characteristics can be helpful for improving performance, but only indirectly through its influence on learning dynamics or its tuning effects.

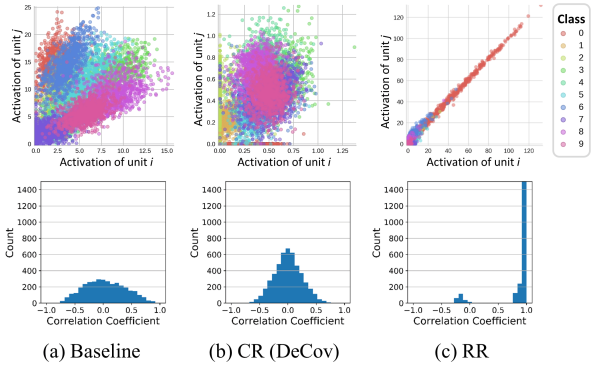


Figure 1: Two representative units' activation scatter plots (upper plots) and histograms of the correlation coefficient distribution (lower plots) for the MNIST dataset. For a 6-layer MLP with 100 units for each layer, the fifth layer's activation vectors, calculated using 10,000 test samples, were used to select two neurons randomly and to generate the plots. (a) The baseline model shows a moderate correlation. (b) CR (DeCov) shows very low correlation. (c) Rank regularizer (RR) has completely different characteristics (high correlation) compared to CR. Despite exhibiting totally different representation characteristics, the performances of the three models are comparable.

1 Introduction

A learned representation can affect the performance of deep neural networks; the distributed and deep natures of the representation are the essential elements for the success of deep learning (Bengio, Courville, and Vincent, 2013). Owing to the depth, deep networks have a greater *expressiveness* compared to other machine learning algorithms (Hinton and others, 1986), or shallow networks (Montufar et al., 2014; Telgarsky, 2015; Eldan and Shamir, 2016; Raghu et al., 2016). In addition to the distributed and deep natures that have been intensively studied, a hidden layer's *representation characteristics* are considered to be important as well. Nonetheless, a relatively limited number of studies have been conducted on this matter. The goal of the present study is to better understand representation characteristics. In this study, the meaning of representation is restricted to *the activation vector of a single hidden layer*, and representation characteristics refer to *the statistical characteristics of the activation vector, such as correlation and sparsity.*

In the past several years, dropout (Srivastava et al., 2014) and batch normalization (BN) (Ioffe and Szegedy, 2015) have become essential regularization options, in addition to the default options of L1 (Hoerl and Kennard, 1970) and L2 (Tibshirani, 1996) weight regularizations. Additionally, manipulating representation characteristics has become increasingly popular for the improvement of performance (Glorot, Bordes, and Ben-

*Authors contributed equally.

gio, 2011; Cogswell et al., 2015; Xiong et al., 2016; Liao et al., 2016; Wen et al., 2016; Belharbi et al., 2017; Choi and Rhee, 2019; Hofer et al., 2020). The regularization methods often lead to improved performance, but rigorous explanation has been missing. Instead, it has been implied or conjectured that the manipulation of the representation can lead to improved performance because of known and relevant concepts in machine learning (typically, a reduced generalization gap is quoted as the supporting empirical evidence). For instance, reduced co-adaptation (that is closely related to the correlation of the representation) has been put forth as a possible reason for the good performance of dropout; sparser or less correlated representations have been argued as better representations because the number of true underlying features must be limited. As another example, reducing covariate shift was the reason for inventing batch normalization, but later it was shown that no reduction of internal covariate shift is observable (Santurkar et al., 2018). Instead, Santurkar et al. (2018) show that BN makes the optimization landscape significantly smoother, Bjorck et al. (2018) demonstrate how large gradient updates can result in diverging loss and activations growing uncontrollably with network depth and how BN avoids these, and De and Smith (2020) provide a downscaling method that can replace BN at a comparable performance. As a more general result on regularizing representation, Locatello et al. (2019) recently showed that aiming for certain activation characteristics ('disentangled representations' in their work, in contrast to 'representation characteristics' in our work) is not as universally meaningful as is often assumed in unsupervised learning.

In our study, statistical characteristics of deep representations are investigated for common supervised learning tasks. As a basic framework for the study, an extensive set of representation regularizers are considered, including a baseline model (no regularization). A total of eight regularization options are investigated to examine six characteristics of deep representations. Among the eight, rank regularizer (RR) and class-wise rank regularizer (cw-RR) were newly designed and tested in this study because of their association with important representation characteristics, such as correlation and rank. The regularizers aim to decrease the rank of representation (increase the correlation of representation) by reducing the stable rank of each mini-batch activation from the all-class samples (RR) or the same-class samples (cw-RR).

Some examples of the representations found with the regularizers are shown in Figure 1 (more examples are shown in Figure 2). In the figure, correlation characteristics vary largely, depending on which regularizers are used. RR shows a strong correlation and has a performance comparable to CR (DeCov), as shown in the lower plot of Figure 1. The comparable performance of RR under strong correlation is precisely the opposite of what has been conjectured for DeCov (Cogswell et al., 2015).

As we will show in Section 4, representation characteristics can be manipulated as intended, by applying a variety of regularizers. Additionally, all these regularizers, as a set, can be a useful tool for improving performance. However, the problem is that there is (perhaps unsurprisingly) no distinct pattern that can be used to assess which regularizer (representation characteristic) is likely to be helpful for a given task. All that can be concluded is that some regularizers would be helpful for any given task; but it is not possible to find which ones would qualify to be helpful for the same. In this paper, we do not claim that deep learning practitioners should not attempt to change representation characteristics owing to this problem. Instead, we empirically show that representation regularization can be a useful option for improving performance at the cost of tedious tuning. Despite of inconspicuous relationship between representation characteristics and performance, representation regularization can work as proxies that affect the learning dynamics of deep network training and thus indirectly improve the performance.

2 Related Works

Popular Regularization Methods Many distinct regularization methods have been developed for deep learning. The most traditional methods are L1 and L2 weight regularizations (L1W, L2W). Dropout (Srivastava et al., 2014) and batch normalization (Ioffe and Szegedy, 2015) have shown large performance improvements in the context of many interesting tasks. With the extended definition of regularization in (Goodfellow, Bengio, and Courville, 2016), many other methods such as data augmentation, adversarial training, and multi-task learning can be considered as regularization methods too. In this work, however, we limit our focus to the traditional, dropout, batch normalization, and representation regularizations.

Representation Regularization Methods A representation regularizer explicitly aims to modify a statistical property of the activation vectors, typically by using a penalty. One of the earliest representation regularizers is the L1 representation regularizer (Glorot, Bordes, and Bengio, 2011), and it applies an L1 penalty to the activation vector instead of the weight vectors. It encourages sparsity in the representation, and it is called L1R in this work. Cheung et al. (2014) reduce a sum-squared cross-covariance between autoencoding

Table 1: Symbols and expressions of representation characteristics.

Characteristic	Symbol	Expression
Amplitude	$\|\bar{z}\|$	$\mathbb{E}_i[\|\,\mathbf{z}_{l,i}\,\|]$
Covariance	$\bar{c}$	$\mathbb{E}_{i\neq j}[\|c_{i,j}\|]$, where $c_{i,j} \triangleq \{\mathbf{C}_l\}_{i,j} = \mathbb{E}[(\mathbf{z}_{l,i} - \mu_{z_{l,i}})(\mathbf{z}_{l,j} - \mu_{z_{l,j}})]$
Correlation	$\bar{\rho}$	$\mathbb{E}_{i\neq j}[\|\rho_{i,j}\|]$, where $\rho_{i,j} \triangleq \{\mathbf{C}_l\}_{i,j}/\sigma_{z_{l,i}}\sigma_{z_{l,j}} = \mathbb{E}[(\mathbf{z}_{l,i} - \mu_{z_{l,i}})(\mathbf{z}_{l,j} - \mu_{z_{l,j}})]/\sigma_{z_{l,i}}\sigma_{z_{l,j}}$
Sparsity	P_s	$\mathbb{E}_{i,n}[\mathbb{1}(z_{l,i}^n)]$, where $\mathbb{1}$ is an indicator function whose output is 1 only when $z_{l,i}^n = 0$
Dead unit	P_d	$\mathbb{E}_i[\mathbb{1}(z_{l,i})]$, where $\mathbb{1}$ is an indicator function whose output is 1 only when $z_{l,i}^n = 0$ for all $n = 1, .., N$
Rank	r	$rank(\mathbf{C}_l)$; numerical evaluations are approximated as the stable rank $\|\mathbf{C}_l\|_F^2 / \|\mathbf{C}_l\|_2^2$

and label unit activations to disentangle representations. Similarly, Cogswell et al. (2015) suggest DeCov that utilizes a penalizing loss function to reduce activation covariance among hidden units. Choi and Rhee (2019) consider the extension to class-wise regularization and provide four representation regularizers: CR (Covariance regularizer), cw-CR (class-wise covariance regularizer), VR (variance regularizer), and cw-VR (class-wise variance regularizer). Among them, CR is equivalent to DeCov.

Role of Explicit Regularizations Zhang et al. (2016) showed that explicit regularizations such as L2W and dropout are not directly responsible for reducing or controlling the generalization error. Rather, they argue that performance improvement can be because of a tuning effect. Arpit et al. (2017) investigated the impact of explicit regularization on the memorization speed and generalization.

Generalization of Deep Networks Recently, generalization bounds for deep neural networks have been heavily studied(Neyshabur, Tomioka, and Srebro, 2015; Dziugaite and Roy, 2017; Bartlett, Foster, and Telgarsky, 2017; Arora et al., 2018; Jiang et al., 2019). Complexity measure is a core component of the generalization bounds. For instance, Neyshabur, Tomioka, and Srebro (2015); Sanyal, Torr, and Dokania (2019) showed that norm-based regularizations can control network complexity. Most of the bounds found so far, however, are far from being tight for deep networks.

3 Representation Characteristics and Regularizers

3.1 Representation Characteristics

Consider a neural network $\mathcal{N}_\mathcal{A}$ whose architecture $\mathcal{A}$ is fixed and the weights for the l^{th} layer are given by $\{\mathbf{W}_l\}$ and $\{\mathbf{b}_l\}$ after training. We write $\mathcal{N}_\mathcal{A} = (\mathbf{W}, \mathbf{b})$ to denote a network and $\mathbf{y}$ or $\mathcal{N}_\mathcal{A}(\mathbf{x})$ to refer to its deterministic output for a given input $\mathbf{x}$. The index l is omitted when the meaning is obvious. The activation vector of the l^{th} layer for the given input $\mathbf{x}$ is denoted as $\mathbf{z}_l(\mathbf{x})$ or simply $\mathbf{z}_l$, and the i^{th} element of $\mathbf{z}_l$ is denoted as $z_{l,i}$. The mean, variance, and standard deviation of $z_{l,i}$ over $p(\mathbf{x})$ are defined as $\mu_{z_{l,i}}$, $v_{z_{l,i}}$, and $\sigma_{z_{l,i}}$, respectively. The covariance of $\mathbf{z}_l$ is defined as $\mathbf{C}_l$. The definitions of class-wise statistics are included in Section A of the supplementary materials.

The basic representation characteristics can be summarized as in Table 1. Since the true distribution of the data is not accessible, the numerical results in the following sections are evaluated using the empirical distribution of the test dataset. For instance, $\mathbf{C}_l$ is calculated as the covariance matrix of N activation vectors $\{\mathbf{z}_l^1, ..., \mathbf{z}_l^N\}$ where $\mathbf{z}_l^n$ corresponds to the activation vector for the n^{th} test data example, $\mathbf{x}^n$. Rank can be calculated by examining $\mathbf{C}_l$, but often there are small eigenvalues that hinder a proper assessment of the rank. Therefore, *stable rank* is evaluated instead.

3.2 Representation Regularizers

In this study, mainly eight options are considered: the baseline model (no regularizer) and seven models of representation regularizers (CR, cw-CR, VR, cw-VR, L1R, RR, cw-RR). Even though dropout and BN do not explicitly target to modify representation characteristics, they were also studied together because the popular regularizers certainly affect the representation characteristics. Regularization terms are added to the original cost function as penalty regularizers. The total cost function $\widetilde{J}$ can be denoted as

$$\widetilde{J} = J + \lambda\Omega(\mathbf{z}), \tag{1}$$

where λ is the loss weight ($\lambda \in [0, \infty)$). Each regularizer targets a different statistical characteristic of the representations. For example, CR and VR reduce covariance and variance of the activations calculated from all-class samples, respectively. L1R decreases the absolute amplitude of activations calculated from all-class samples to make the activations sparser. Regularizers with prefix 'cw-' are the class-wise counterparts of all-class regularizers. All the loss functions are summarized in Section A of the supplementary material.

Rank Regularizer In deep learning, a low-rank approximation of convolutional filters (Jaderberg, Vedaldi,

and Zisserman, 2014; Lebedev et al., 2014; Tai et al., 2015) and weight matrices (Nakkiran et al., 2015; Masana et al., 2017; Alvarez and Salzmann, 2017) has been widely used for network compression and fast network training. The recent work by Sanyal, Torr, and Dokania (2019) proposes stable rank normalization (SRN), which can improve the generalization of the network in classification tasks. Given the available literature, regularization methods are typically applied to weights, and not to activations. However, in this study, RR and cw-RR are applied to activations as penalty regularizers. RR is designed to encourage a lower rank of representations and is used while training the network. Because the usual definition of rank can be very sensitive to small singular values, we use *stable rank* of the activation matrix $\mathbf{Z} = [\mathbf{z}_l^1, \ldots, \mathbf{z}_l^{N_{MB}}]^T$ as a surrogate. Note that N_{MB} instead of N activation vectors are used for each mini-batch. The stable rank of $\mathbf{Z}$ is defined as

$$\Omega_{RR} = \frac{\|\mathbf{Z}\|_F^2}{\|\mathbf{Z}\|_2^2} = \frac{\sum_i s_i^2}{\max_i s_i^2}, \tag{2}$$

where $\|\mathbf{Z}\|_F$ is the Frobenius norm, $\|\mathbf{Z}\|_2$ is the spectral norm, and $\{s_i\}$ are the singular values of $\mathbf{Z}$. From $\frac{\sum_i s_i^2}{\max_i s_i^2}$, it can be clearly seen that stable rank is upper-bounded by the rank that counts strictly positive singular values. As the spectral norm is based on singular value decomposition, calculating the derivative of the stable rank for every mini-batch is a computationally heavy operation. To reduce the computational burden, we apply an approximation using a special case of Hölder's inequality.

$$\Omega_{RR} = \frac{\|\mathbf{Z}\|_F^2}{\|\mathbf{Z}\|_2^2} = \frac{\mathrm{trace}(\mathbf{Z}^T\mathbf{Z})}{\|\mathbf{Z}\|_2^2} \tag{3}$$

$$\geq \frac{\mathrm{trace}(\mathbf{Z}^T\mathbf{Z})}{\|\mathbf{Z}\|_1\|\mathbf{Z}\|_\infty} \tag{4}$$

$$= \frac{\sum_{i,n}(z_i^n)^2}{(\max_i \sum_{n=1}^{N_{MB}} |z_i^n|)(\max_n \sum_{i=1}^{M} |z_i^n|)} \tag{5}$$

The inequality $\|\mathbf{Z}\|_2 \leq \sqrt{\|\mathbf{Z}\|_1\|\mathbf{Z}\|_\infty}$ is used, where $\|\mathbf{Z}\|_1$ is the maximum absolute column-wise sum of the matrix $\mathbf{Z}$ (sum of all activation values of unit i) and $\|\mathbf{Z}\|_\infty$ is the maximum absolute row-wise sum of the matrix $\mathbf{Z}$ (sum of all activation values of sample n). The extension of RR to cw-RR is straightforward.

4 Experiments

In this section, it is empirically shown that the regularization affects the statistical characteristics of deep representations. The relationship between performance and the representation characteristics is also examined. Finally, performance results on a variety of tasks are presented.

4.1 Experimental Settings

As examples of simple networks, we used a 6-layer MLP for the MNIST dataset, and a CNN with four convolutional layers and one fully-connected layer for the CIFAR-10/100 dataset. (In this paper, we call them 'MLP' and 'CNN' respectively, for convenience.) As examples of sophisticated networks, VGG-16 on the CIFAR-10/100, ResNet-18/50 on the ImageNet/Tiny-ImageNet datasets were used. For ResNet, a single fully-connected layer was added following the last average pooling layer. Validation performance was evaluated with different loss weights {0.001, 0.01, 0.1, 1, 10, 100, 1000}, and the one with the best validation performance for each regularizer and condition was chosen for testing. For ResNet, pre-trained models were fine-tuned with the regularizers. Each training trial was repeated five times unless mentioned; the mean and standard deviation of the five trials are reported. The mini-batch size was set to 100 for MLP (MNIST) and CNN (CIFAR-10), 128 for VGG-16 (CIFAR-10) and ResNet-50 (Tiny-ImageNet), and 256 for ResNet-18 (ImageNet). For CIFAR-100 that has 100 classes, mini-batch size of 500 was used to calculate meaningful class-wise statistics. Experiments with class-wise regularizers were not performed for ImageNet and Tiny-Imagenet datasets to avoid inefficient training of large mini-batch size. More details of the experimental settings can be found in Section B of the supplementary material.

4.2 Effect of Regularization on Representation Characteristics

The representation characteristics were visually and quantitatively investigated, as shown in Figure 2 and Table 2. In Figure 2, it can be observed that the representation characteristics exhibit large variations depending on the choice of the regularizer. In particular, dropout shows a strong pair-wise correlation, as shown in the lower plot of Figure 2(b). This is precisely the opposite of what has been believed for dropout. Even though not shown, the visualization of the CIFAR-10/100, the Tiny-ImageNet, and the ImageNet datasets showed similar patterns as in Figure 2 (the patterns were less distinct for the class-wise regularizers). The plots of the top three principal components of the representations are included in Section C of the supplementary material, to present distinct global trends of the representations.

Our quantitative result confirms the visualization.

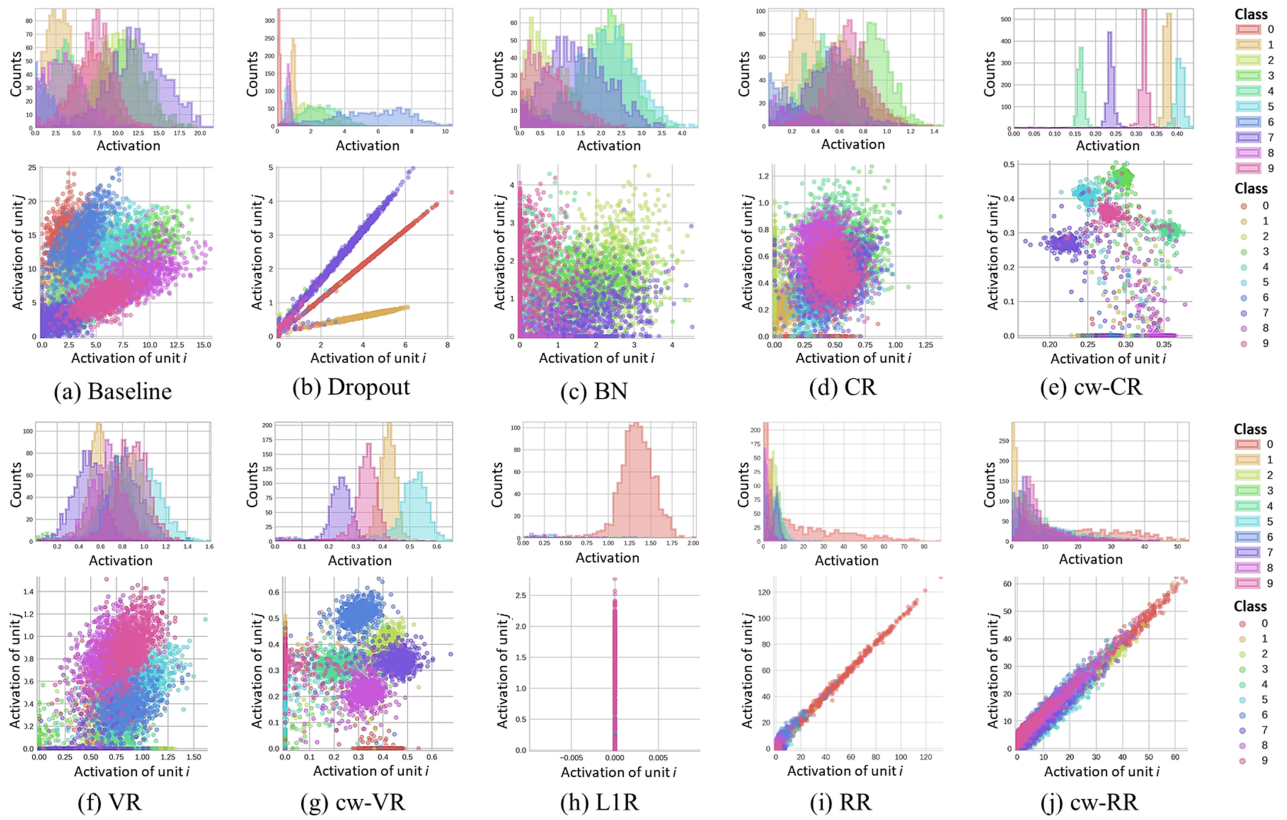
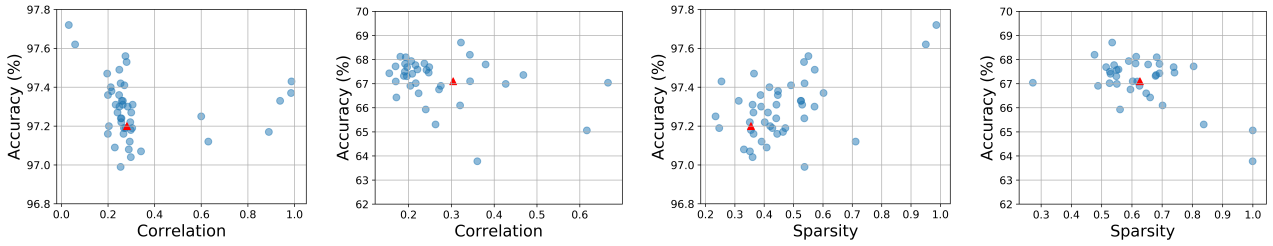


Figure 2: Activation histogram of a single unit (upper plots) and the activation scatter plots of two randomly chosen units (lower plots) for a 6-layer MLP trained with the MNIST dataset. The plots were produced in the same way as in Figure 1. **(upper)** The baseline has a large class-wise variance and inter-class overlaps; BN and CR (covariance regularizer) show similar properties. Dropout looks completely different where activation values are more spread out. cw-CR and cw-VR show well-separated activation distributions because they are regularized class-wise. L1R increases the sparsity of representation. **(lower)** As mentioned in the caption of Figure 1, CR, RR, and cw-RR show completely different patterns. cw-CR and cw-VR show low correlation per class because they are regularized class-wise.

Table 2: Statistical characteristics of learned representations. The characteristics of MLP were generated in the same way as in Figure 1. For ResNet, one fully-connected layer was added next to the last average pooling layer and regularizers were applied on it. One can observe that the characteristics are modified, as initially predicted (indicated in **bold**).

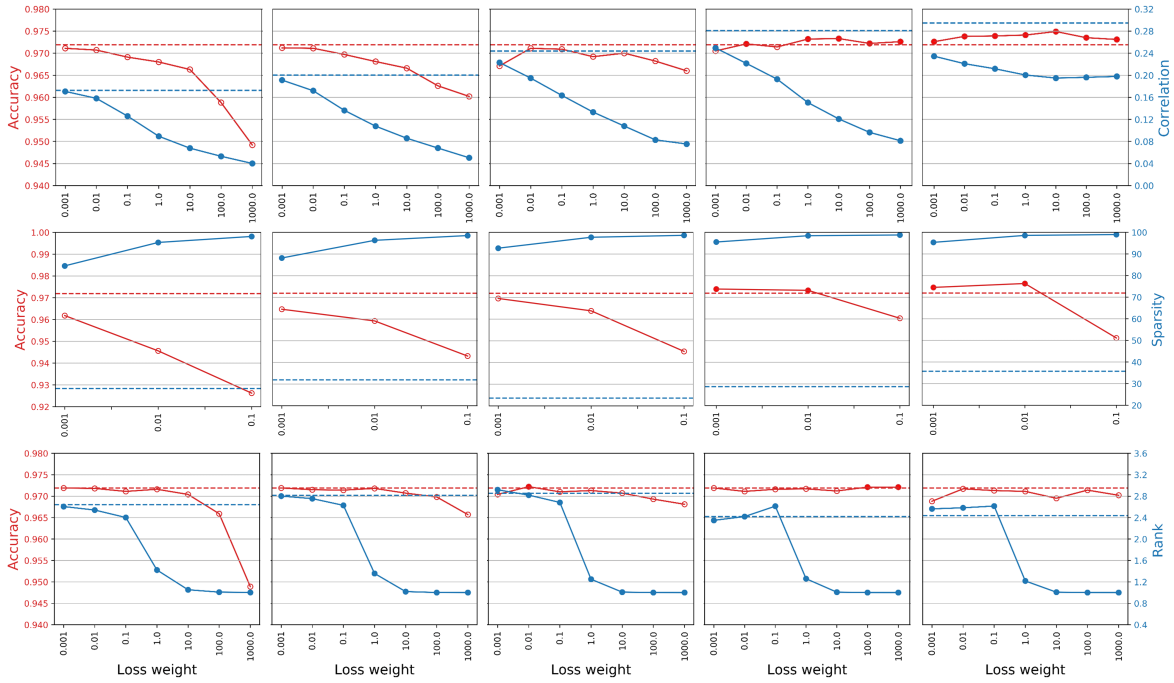
Data & Net.	Reg.	Accuracy (%)	Amplitude	Covariance	Correlation	Sparsity	Dead unit	Rank
MNIST MLP	Baseline	97.15	4.93	2.08	0.27	0.34	0.13	2.41
	CR	97.50	0.50	**0.01**	**0.19**	0.40	0.03	7.12
	L1R	97.65	1.29	0.03	0.40	**0.97**	**0.39**	5.94
	RR	97.19	7.23	226.20	0.90	0.43	0.18	**1.00**
Tiny-ImageNet ResNet-50	Baseline	78.56	1.06	0.155	0.08	0.436	0.00	6.51
	CR	78.14	0.26	**0.007**	**0.04**	0.585	0.00	26.09
	L1R	78.32	0.22	0.016	0.05	**0.780**	0.00	5.36
	RR	77.99	1.59	0.204	0.12	0.155	0.00	**1.46**
ImageNet ResNet-18	Baseline	70.34	0.90	0.049	0.062	0.010	0.00	6.46
	CR	68.76	0.52	**0.005**	**0.051**	0.000	0.00	22.46
	L1R	69.51	0.83	0.067	0.078	**0.010**	0.00	2.40
	RR	69.75	0.92	20.448	0.968	0.012	0.00	**1.00**

Each characteristic was obtained by applying the largest loss weight possible while maintaining comparable performance with the baseline model. The result confirms that the statistical characteristics targeted by each regularizer are manipulated as expected **(Bold)** in Table 2. In particular, RR regularizes the stable rank, and thus works as intended. RR (highly correlated representations) shows comparable performance to CR (decorrelated representations). This result is somewhat counter-intuitive to the conventional wisdom.



(a) Correlation (MNIST) **(b)** Correlation (CIFAR-100) **(c)** Sparsity (MNIST) **(d)** Sparsity (CIFAR-100)

Figure 3: Relationship between the representation characteristics and the performance on the MNIST (MLP) and the CIFAR-100 (VGG-16). Each blue point indicates a single pair of representation characteristic and performance, from the corresponding model that utilizes specific regularizer and loss weight. The red triangle indicates the baseline model.



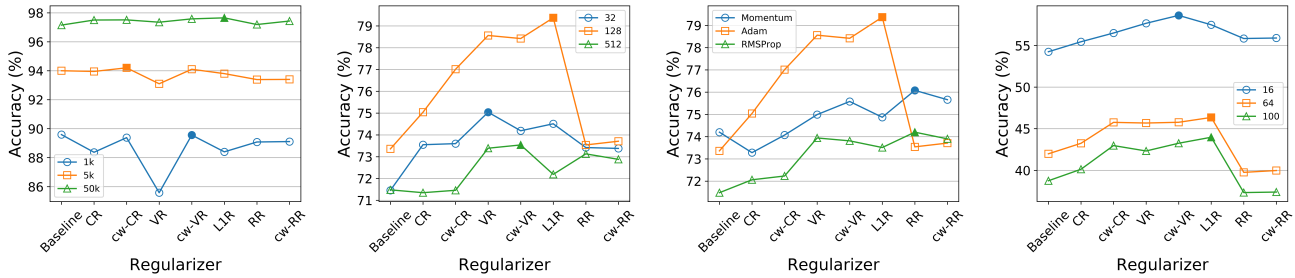
(a) Layer 1 (b) Layer 2 (c) Layer 3 (d) Layer 4 (e) Layer 5

Figure 4: Layer dependence of representation regularizations. Each plot was generated with the MLP on the MNIST in the same manner, as shown in Figure 1. Regularizers were applied to all the layers. The top, middle, and bottom rows correspond to results of CR, L1R, and RR; the red and blue dotted lines indicate the baseline model's performance and the characteristics of each of its layers, respectively. Note that some models of L1R are excluded because they cannot be trained with loss weights that are greater than 0.1.

4.3 Relationship between Representation Characteristics and Performance

To examine the relationship between representation characteristics and performance more precisely, the scatter plots of CORRELATION, SPARSITY, and performance were drawn in Figure 3. Each circle corresponds to one characteristic and performance pair from a specific choice of model (regularizer and loss weight), and the red triangle is that of the baseline model. Only the points with comparable results to the baseline were drawn, and each model was trained and tested only once. One can observe that neither correlation nor sparsity has a clear relationship with performance. We discuss this phenomenon in Section 5.

So far, the experimental results have been shown when the regularizers are applied to the last fully-connected layer where the representation can be considered as the most processed feature set. We now investigate how differently regularizers behave when applied to different layers. In Figure 4, we apply CR (top), L1R (middle), and RR (bottom) to each layer of the 6-layer MLP. The result confirms that the regularizers perform



(a) Data size (MLP on MNIST)

(b) Layer width (CNN on CIFAR-10)

(c) Optimizer (CNN on CIFAR-10)

(d) Number of classes (CNN on CIFAR-100)

Figure 5: Results of the 'task condition' experiment. Each color indicates different task conditions such as data size, number of hidden units, choice of optimizer, and number of classes. The result shows that the seven regularizers often outperform the baseline; however, the best performing regularizer cannot be specified for the given task condition, especially when the additional experiment results are considered together.

Table 3: Test accuracy (%) of MLP, VGG-16, and ResNet-50 models on the MNIST, the CIFAR-10/100, and the Tiny-ImageNet datasets, respectively. RR and cw-RR often perform better than the others, and seven representation regularizers often mildly outperform the baseline. For Tiny-ImageNet, we did not experiment the class-wise regularizers because their mini-batch size is required to be much larger than the number of classes and such a configuration leads to inefficient training.

Regularizer	MLP on MNIST	VGG-16 on CIFAR-10	VGG-16 on CIFAR-100	ResNet-50 on Tiny-ImageNet
Baseline	97.15 ± 0.11	92.26 ± 0.14	67.11 ± 0.44	78.53 ± 0.09
CR	97.50 ± 0.05	92.39 ± 0.14	67.07 ± 1.20	78.41 ± 0.08
cw-CR	97.51 ± 0.10	92.31 ± 0.16	67.54 ± 0.22	-
VR	97.35 ± 0.11	92.40 ± 0.22	67.38 ± 0.45	77.84 ± 0.18
cw-VR	97.58 ± 0.06	92.46 ± 0.27	$\mathbf{67.63 \pm 0.32}$	-
L1R	$\mathbf{97.65 \pm 0.08}$	92.46 ± 0.10	65.56 ± 0.31	78.54 ± 0.13
RR	97.19 ± 0.10	92.21 ± 0.12	67.37 ± 0.29	$\mathbf{78.57 \pm 0.13}$
cw-RR	97.43 ± 0.08	$\mathbf{92.56 \pm 0.08}$	67.45 ± 0.60	-

better than the baseline (red dotted line) when applied on layers 4 and 5. Conversely, when regularizers are applied to lower layers, the performance declines, as loss weight increases even though corresponding characteristics (blue lines) can be controlled (the blue dotted lines are each layer's characteristic of the baseline model). We conjecture that this is because low-level features that should flow to the upper layers with a rich level of information can be negatively affected by putting constraints on the activations.

4.4 Performance Improvement by Representation Regularization

We investigate if regularizers can indeed improve the performance for a given task condition. For instance, a regularizer might be effective when the task has a small number of data examples and another regularizer might be effective when the task has a large number of classes. The following task conditions were chosen for the experiment: a learning task with 1k, 5k, or 50k data size, 32, 128, or 512 layer width, a specific dataset, a small number of classes, or a specific optimizer. Rigorously speaking, layer width and optimizer choice are not relevant to the 'task', but relevant to the architecture and hyperparameter. Nonetheless, we use the term 'task condition' loosely in this work. We performed experiments on the MNIST and the CIFAR-10/100 datasets using the twelve regularization setups and the four task conditions. All the regularizers are applied to the last fully-connected layer only.

We first investigated simple MLP and CNN models. The results in Figure 5 indicate that the regularizers are generally beneficial for improving the performance. Even though no single representation characteristic consistently outperforms the rest, it is possible to improve performance by using the regularizers as a set and by choosing the best performing regularizer for the given task. On the other hand, we have performed an extensive experiment in addition to the results shown in Figure 5 (see Table 6, 7 and 8 in the supplementary materials) but we were not able to observe any meaningful relationship between a type of task condition and a type of regularizer.

We have also investigated more sophisticated networks

of VGG-16 and ResNet-50 as shown in Table 3. Even for the sophisticated networks, we were able to affect representation characteristics using the regularizers and achieve a mild performance improvement. Considering that the networks have a long history of enhancement by numerous researchers and that they might have approached the best possible performance for the given task condition, even the mild improvements can be regarded as meaningful if not significant. While the performance improvements are clearly observable, again it was impossible to identify a meaningful relationship when analyzed together with the results in Table 6, 7 and 8.

Typically, previous works on regularizers have considered only a small number of regularizers in each work. By evaluating only a small number of regularizers over a small number of task conditions, it can be easy to identify a possibly meaningful relationship between a regularizer and a task condition. When many regularizers and many task conditions are evaluated as in our work, however, it becomes apparent that a strong relationship is extremely difficult to observe. We conclude that it is not only risky but also likely to be incorrect to imply or conjecture that a manipulation of representation can lead to an improved performance.

5 Discussion and Future Work

Equivalent Networks Infinitely many global optima exist for deep neural networks (Du et al., 2018). By re-arranging the hidden units or by properly scaling the incoming and outgoing weights of ReLU networks, one can easily construct equivalent networks with different representation characteristics but with exactly same outputs (Dinh et al., 2017). Therefore, statistical characteristics such as correlation and covariance can be easily altered without affecting performance, simply by choosing one of the equivalent networks. The easiness of constructing equivalent networks clearly indicates that at least some of the statistical characteristics of representation do not need to have a certain property (e.g. low correlation) when the best performance is achieved.

Landscape of Minima Training a deep network corresponds to finding minima of a high-dimensional non-convex loss function, and understanding the loss landscape has been an important research topic. Garipov et al. (2018) have shown that the minima of the complex loss functions are in fact connected by simple curves over which training and test accuracy are nearly constant, and Draxler et al. (2018) have shown that continuous paths can be constructed between minima where the loss is essentially flat over the paths. These results can have a stronger implication than the existence of equivalent networks, because the statistical characteristics of representation over one of such connected path can be much more complicated with a wild variation.

Learning Dynamics In the linear least square case, a model converges along the eigenvectors of the covariance matrix at a rate depending on the magnitude of their corresponding eigenvalues (LeCun, Kanter, and Solla, 1991). Therefore, representation regularization affects not only representation characteristics but also learning dynamics. Desjardins et al. (2015) propose a method to reparameterize the weights of the neural network by implicitly whitening each layer's activations. The method improves the learning dynamics owing to reparameterization; thus, the networks can be trained more efficiently. Also, Combes et al. (2018) prove various properties of learning dynamics in deep nonlinear neural networks by studying the case of binary classification under strong assumptions such as linear separability of the data.

While a large progress has been made in recent years, the learning dynamics of deep network still remain largely as an open problem. Together with our experiment results, it can be concluded that representation characteristics, performance, and learning dynamics are all interwoven together. While we negatively concluded on the causal and direct effect of representation characteristics to the performance, causal effects via learning dynamics is still a possibility - representation characteristics can certainly affect learning dynamics, and learning dynamics might be able to affect the performance in a causal and explainable way. In this case, representation characteristics would indirectly affect the performance. Empirical study of all three elements remains as a possible future work.

Generalization Bounds Jiang et al. (2019) performed a large scale study on generalization of deep networks. In the study, more than 40 complexity measures from the existing studies were chosen and investigated. The measures include traditional ones (such as VC dimension), weight matrices' norm and margin-based ones, local minima's sharpness related ones, and optimization-based ones. Representation characteristics, however, have been hardly considered in the theoretical and empirical studies of deep network generalization, despite representation regularizers certainly being able to affect generalization bounds through its influence on the learning of weights. The overall effect of representation characteristics can be difficult to formulate because such generalization bounds will need to depend on $p(\mathbf{x})$, but perhaps a tighter bound might be obtainable for the same reason. Thus, any theoretical result might shed a light on representation characteristics' influence on the generalization performance.

References

Alvarez, J. M., and Salzmann, M. 2017. Compression-aware training of deep networks. In Advances in Neural Information Processing Systems, 856–867.

Arora, S.; Ge, R.; Neyshabur, B.; and Zhang, Y. 2018. Stronger generalization bounds for deep nets via a compression approach. arXiv preprint arXiv:1802.05296.

Arpit, D.; Jastrzębski, S.; Ballas, N.; Krueger, D.; Bengio, E.; Kanwal, M. S.; Maharaj, T.; Fischer, A.; Courville, A.; Bengio, Y.; et al. 2017. A closer look at memorization in deep networks. arXiv preprint arXiv:1706.05394.

Bartlett, P. L.; Foster, D. J.; and Telgarsky, M. J. 2017. Spectrally-normalized margin bounds for neural networks. In Advances in Neural Information Processing Systems, 6240–6249.

Belharbi, S.; Chatelain, C.; Herault, R.; and Adam, S. 2017. Neural networks regularization through invariant features learning. arXiv preprint arXiv:1709.01867.

Bengio, Y.; Courville, A.; and Vincent, P. 2013. Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence 35(8):1798–1828.

Bjorck, N.; Gomes, C. P.; Selman, B.; and Weinberger, K. Q. 2018. Understanding batch normalization. In Advances in Neural Information Processing Systems, 7694–7705.

Cheung, B.; Livezey, J. A.; Bansal, A. K.; and Olshausen, B. A. 2014. Discovering hidden factors of variation in deep networks. arXiv preprint arXiv:1412.6583.

Choi, D., and Rhee, W. 2019. Utilizing class information for dnn representation shaping. Thirty-Third AAAI Conference on Artificial Intelligence.

Cogswell, M.; Ahmed, F.; Girshick, R.; Zitnick, L.; and Batra, D. 2015. Reducing overfitting in deep networks by decorrelating representations. arXiv preprint arXiv:1511.06068.

Combes, R. T. D.; Pezeshki, M.; Shabanian, S.; Courville, A.; and Bengio, Y. 2018. On the learning dynamics of deep neural networks.

De, S., and Smith, S. 2020. Batch normalization biases residual blocks towards the identity function in deep networks. arXiv preprint arXiv:2002.10444.

Desjardins, G.; Simonyan, K.; Pascanu, R.; et al. 2015. Natural neural networks. In Advances in Neural Information Processing Systems, 2071–2079.

Dinh, L.; Pascanu, R.; Bengio, S.; and Bengio, Y. 2017. Sharp minima can generalize for deep nets. In Proceedings of the 34th International Conference on Machine Learning-Volume 70, 1019–1028. JMLR. org.

Draxler, F.; Veschgini, K.; Salmhofer, M.; and Hamprecht, F. A. 2018. Essentially no barriers in neural network energy landscape. arXiv preprint arXiv:1803.00885.

Du, S. S.; Lee, J. D.; Li, H.; Wang, L.; and Zhai, X. 2018. Gradient descent finds global minima of deep neural networks. arXiv preprint arXiv:1811.03804.

Dziugaite, G. K., and Roy, D. M. 2017. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. arXiv preprint arXiv:1703.11008.

Eldan, R., and Shamir, O. 2016. The power of depth for feedforward neural networks. In Conference on Learning Theory, 907–940.

Garipov, T.; Izmailov, P.; Podoprikhin, D.; Vetrov, D. P.; and Wilson, A. G. 2018. Loss surfaces, mode connectivity, and fast ensembling of dnns. In Advances in Neural Information Processing Systems, 8789–8798.

Glorot, X.; Bordes, A.; and Bengio, Y. 2011. Deep sparse rectifier neural networks. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, 315–323.

Goodfellow, I.; Bengio, Y.; and Courville, A. 2016. Deep learning. MIT press.

Hinton, G. E., et al. 1986. Learning distributed representations of concepts. In Proceedings of the eighth annual conference of the cognitive science society, volume 1, 12. Amherst, MA.

Hoerl, A. E., and Kennard, R. W. 1970. Ridge regression: Biased estimation for nonorthogonal problems. Technometrics 12(1):55–67.

Hofer, C. D.; Graf, F.; Niethammer, M.; and Kwitt, R. 2020. Topologically densified distributions. International Conference on Machine Learning.

Ioffe, S., and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International Conference on Machine Learning, 448–456.

Jaderberg, M.; Vedaldi, A.; and Zisserman, A. 2014. Speeding up convolutional neural networks with low rank expansions. arXiv preprint arXiv:1405.3866.

Jiang, Y.; Neyshabur, B.; Mobahi, H.; Krishnan, D.; and Bengio, S. 2019. Fantastic generalization measures and where to find them. arXiv preprint arXiv:1912.02178.

Lebedev, V.; Ganin, Y.; Rakhuba, M.; Oseledets, I.; and Lempitsky, V. 2014. Speeding-up convolutional

neural networks using fine-tuned cp-decomposition. arXiv preprint arXiv:1412.6553.

LeCun, Y.; Kanter, I.; and Solla, S. A. 1991. Second order properties of error surfaces: Learning time and generalization. In Advances in neural information processing systems, 918–924.

Liao, R.; Schwing, A.; Zemel, R.; and Urtasun, R. 2016. Learning deep parsimonious representations. In Advances in Neural Information Processing Systems, 5076–5084.

Locatello, F.; Bauer, S.; Lucic, M.; Gelly, S.; Schölkopf, B.; and Bachem, O. 2019. Challenging common assumptions in the unsupervised learning of disentangled representations. International Conference on Machine Learning.

Masana, M.; van de Weijer, J.; Herranz, L.; Bagdanov, A. D.; and MAlvarez, J. 2017. Domain-adaptive deep network compression. network 16:30.

Montufar, G. F.; Pascanu, R.; Cho, K.; and Bengio, Y. 2014. On the number of linear regions of deep neural networks. In Advances in neural information processing systems, 2924–2932.

Nakkiran, P.; Alvarez, R.; Prabhavalkar, R.; and Parada, C. 2015. Compressing deep neural networks using a rank-constrained topology. In Sixteenth Annual Conference of the International Speech Communication Association.

Neyshabur, B.; Tomioka, R.; and Srebro, N. 2015. Norm-based capacity control in neural networks. In Conference on Learning Theory, 1376–1401.

Raghu, M.; Poole, B.; Kleinberg, J.; Ganguli, S.; and Sohl-Dickstein, J. 2016. On the expressive power of deep neural networks. arXiv preprint arXiv:1606.05336.

Santurkar, S.; Tsipras, D.; Ilyas, A.; and Madry, A. 2018. How does batch normalization help optimization? In Advances in Neural Information Processing Systems, 2483–2493.

Sanyal, A.; Torr, P. H.; and Dokania, P. K. 2019. Stable rank normalization for improved generalization in neural networks and gans. arXiv preprint arXiv:1906.04659.

Srivastava, N.; Hinton, G. E.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. Journal of machine learning research 15(1):1929–1958.

Tai, C.; Xiao, T.; Zhang, Y.; Wang, X.; et al. 2015. Convolutional neural networks with low-rank regularization. arXiv preprint arXiv:1511.06067.

Telgarsky, M. 2015. Representation benefits of deep feedforward networks. arXiv preprint arXiv:1509.08101.

Tibshirani, R. 1996. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society. Series B (Methodological) 267–288.

Wen, Y.; Zhang, K.; Li, Z.; and Qiao, Y. 2016. A discriminative feature learning approach for deep face recognition. In European Conference on Computer Vision, 499–515. Springer.

Xiong, W.; Du, B.; Zhang, L.; Hu, R.; and Tao, D. 2016. Regularizing deep convolutional neural networks with a structured decorrelation constraint. In Data Mining (ICDM), 2016 IEEE 16th International Conference on, 519–528. IEEE.

Zhang, C.; Bengio, S.; Hardt, M.; Recht, B.; and Vinyals, O. 2016. Understanding deep learning requires rethinking generalization. arXiv preprint arXiv:1611.03530.

Supplementary Materials

A. Class-wise Statistics and Representation Regularizers

Based on the notations in Section 3, we define class-wise statistics that are calculated using only samples of class k, out of a total of K labels in the mini-batch. Class-wise mean, covariance, and variance are defined as follows.

$$\mu_{z_{l,i}}^{(k)} = \mathbb{E}_{n \in S_k}[z_{l,i}^n]. \tag{6}$$

$$c_{i,j}^{(k)} = \mathbb{E}_{n \in S_k}[(z_{l,i}^n - \mu_{z_{l,i}}^{(k)})(z_{l,j}^n - \mu_{z_{l,j}}^{(k)})]. \tag{7}$$

$$v_{z_{l,i}}^{(k)} = c_{i,i}^{(k)}. \tag{8}$$

Here, S_k is the set that contains the indices of the samples with the class label k. Note that superscripts with and without parenthesis indicate class label and sample index, respectively.

Penalty loss functions of the representation regularizers are summarized in Table 4.

Table 4: Penalty loss functions of representation regularizers.

Symbol	Penalty loss function	Description of regularization term
Ω_{CR}	$= \sum_{i \neq j} (c_{i,j})^2$	*Covariance* of representations calculated from all-class samples.
$\Omega_{cw\text{-}CR}$	$= \sum_k \sum_{i \neq j} (c_{i,j}^{(k)})^2$	*Covariance* of representations calculated from **the same class samples**.
Ω_{VR}	$= \sum_i v_i$	*Variance* of representations calculated from all-class samples.
$\Omega_{cw\text{-}VR}$	$= \sum_k \sum_i v_{z_{l,i}}^{(k)}$	*Variance* of representations calculated from **the same class samples**.
Ω_{L1R}	$= \sum_n \sum_i \lvert z_{l,i}^n \rvert$	*Absolute amplitude* of representations calculated from all-class samples.
Ω_{RR}	$= \lVert \mathbf{Z}_l \rVert_F^2 / \lVert \mathbf{Z}_l \rVert_2^2$	*Stable rank* of representations calculated from all-class samples.
$\Omega_{cw\text{-}RR}$	$= \sum_k (\lVert \mathbf{Z}_l^{(k)} \rVert_F^2 / \lVert \mathbf{Z}_l^{(k)} \rVert_2^2)$	*Stable rank* of representations calculated from **the same class samples**.

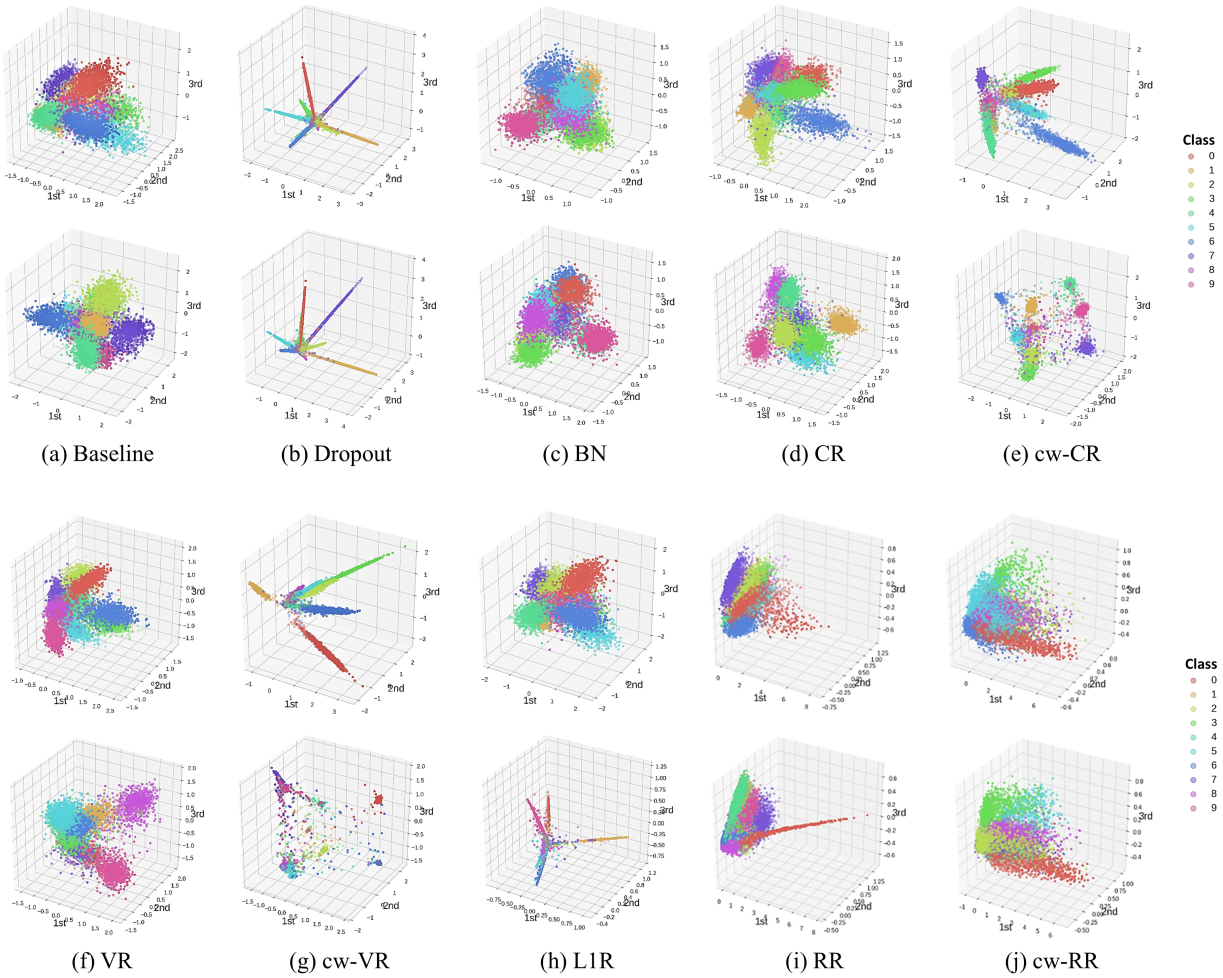
B. Experiment Details

B.1 Parameters of CNNs

A 6-layer MLP with 100 units per hidden layer was used for MNIST image classification tasks. A CNN with four convolutional layers and one fully-connected layer was used for CIFAR-10/100 image classification tasks. Architecture details are described in Table 5.

Table 5: Architecture hyperparameters of CIFAR-10/100 CNN model.

Layer	Parameter
Convolutional layer-1	Number of filters=32, Filter size=3 × 3, Convolution stride=1
Convolutional layer-2	Number of filters=64, Filter size=3 × 3, Convolution stride=1
Max-pooling layer-1	Pooling size=2 × 2, Pooling stride=2
Convolutional layer-3	Number of filters=128, Filter size=3 × 3, Convolution stride=1
Max-pooling layer-2	Pooling size=2 × 2, Pooling stride=2
Convolutional layer-4	Number of filters=128, Filter size=3 × 3, Convolution stride=1
Max-pooling layer-3	Pooling size=2 × 2, Pooling stride=2
Fully connected layer	Number of units=128



(a) Baseline (b) Dropout (c) BN (d) CR (e) cw-CR

(f) VR (g) cw-VR (h) L1R (i) RR (j) cw-RR

Figure 6: Top three principal components of learned representations. For each regularization, the upper one shows the scatter plot of activations before passing through the ReLU, and the lower one shows the scatter plot of activations after passing through the ReLU. Note that the top three PCA directions are affected when ReLU converts the negative values to zero, and thus the upper and lower plots look different.

B.2 Parameters of Sophisticated Networks

VGG-16 We adapt the state-of-the-art VGG-16 network for the CIFAR-10 and CIFAR-100 datasets from (https://github.com/geifmany/cifar-vgg). To manipulate the properties of the representation statistics, for the penultimate fully-connected layer, we exclude additional processes (e.g. batch normalization, dropout), then apply the representation regularizer. All hyperparameters are chosen to be the same as the SOTA network. While fine-tuning the network, the network was trained for only 100 epochs with a smaller initial learning rate of 0.001.

Resnet We adapt Resnet-50 for Tiny-Imagenet and Resnet-18 for Imagenet. To manipulate the properties of the representation characteristics, we add a fully-connected layer following the last average pooling layer. Then we apply the representation regularizers to the added layer. All hyperparameters are chosen to be the same as the SOTA network. While fine-tuning the network, the network was trained for only 100 epochs with smaller initial learning rates of 0.02 (for Tiny-Imagenet) and 0.0001 (for Imagenet).

C. Principal Component Analysis of Learned Representations

In Figure 2, we have shown activation histogram of a single unit and activation scatter plots of two randomly chosen units. The plots clearly demonstrate the variations in representation characteristics as different representation regularizers are applied. Alternatively, we can choose the directions with the largest variations using PCA, and generate similar plots. In Figure 6, the top three PCA directions were chosen to generate activation scatter plots. As in Figure 2, variations in representation characteristics can be conspicuously observed.

D. Result of Condition Tasks

D.1 Task Conditions

Experimental conditions are listed as follows. (Default conditions are shown in bold.)

- Training data size: 1k, 5k, **50k**
- Layer width: (MNIST) 2, 8, **100** / (CIFAR-10/100): 32, **128**, 512
- Optimizer (CIFAR-10): **Adam (lr=0.0001)**, Momentum (lr=0.01, momentum=0.9), RMSProp (lr=0.0001)
- Number of classes (CIFAR-100): 16, 64, **100**

D.2 Result of Condition Tasks

Table 6: Condition experiment results for the MNIST MLP model. A 6-layer MLP with 100 units per hidden layer was used. The best performing regularizer in each condition (each column) is shown in bold; other regularizers whose performance range overlaps with the best one are highlighted in green. For the default condition, the standard values of data size=50k and layer width=100 were used, and the Adam optimizer was applied. For other columns, all the conditions were the same as the default, except for the condition indicated on the top part of the columns.

Regularizer	Default	Data Size		Layer Width	
		1k	5k	2	8
Baseline	97.15 ± 0.11	88.59 ± 0.19	94.00 ± 0.07	68.38 ± 0.07	89.48 ± 0.57
L1W	97.15 ± 0.06	88.36 ± 0.27	94.96 ± 0.11	68.33 ± 0.15	88.98 ± 0.58
L2W	96.98 ± 0.40	88.62 ± 0.18	94.14 ± 0.10	68.34 ± 0.13	89.35 ± 0.23
Dropout	97.30 ± 0.08	**89.71 ± 0.23**	**94.41 ± 0.11**	37.91 ± 1.32	86.06 ± 1.05
BN	97.19 ± 0.12	89.19 ± 0.04	94.40 ± 0.10	57.92 ± 0.93	**92.49 ± 0.58**
CR	97.50 ± 0.05	88.37 ± 0.24	93.95 ± 0.06	65.20 ± 0.25	89.75 ± 0.74
cw-CR	97.51 ± 0.10	89.38 ± 0.05	94.20 ± 0.15	68.50 ± 0.11	89.19 ± 1.11
VR	97.35 ± 0.11	85.58 ± 0.14	93.10 ± 0.22	67.61 ± 0.13	90.78 ± 0.28
cw-VR	97.58 ± 0.06	89.56 ± 0.18	94.10 ± 0.12	**69.66 ± 0.06**	89.99 ± 0.63
L1R	**97.65 ± 0.08**	88.40 ± 0.20	93.80 ± 0.13	35.61 ± 0.26	11.35 ± 0.00
RR	97.19 ± 0.10	89.08 ± 0.17	93.39 ± 0.05	61.65 ± 0.20	87.69 ± 0.16
cw-RR	97.43 ± 0.08	89.11 ± 0.19	93.40 ± 0.17	61.43 ± 0.12	87.37 ± 0.39

Table 7: Condition experiment results for the CIFAR-10 CNN model (Test error %). The best performing regularizer in each condition (each column) is shown in bold; other regularizers whose performance range overlaps with the best one are highlighted in green. For the default condition, the standard values of data size=50k, layer width=128, and the Adam optimizer was applied. For the others, all the conditions were the same as the default, except for the condition indicated on the top part of the columns. Regularizers were applied to the fully-connected layer.

Regularizer	Default	Data Size		Layer Width		Optimizer	
		1k	5k	32	512	Momentum	RMSProp
Baseline	73.36 ± 0.16	43.93 ± 0.36	56.05 ± 0.43	71.46 ± 0.63	71.48 ± 1.06	74.22 ± 0.37	71.48 ± 1.21
L1W	73.54 ± 0.39	43.36 ± 0.91	55.68 ± 0.66	71.35 ± 1.14	72.04 ± 0.72	74.27 ± 0.40	71.70 ± 0.99
L2W	74.29 ± 0.98	43.43 ± 0.22	55.13 ± 0.81	71.46 ± 0.30	72.21 ± 0.83	73.65 ± 0.54	71.98 ± 0.88
Dropout	73.63 ± 0.21	43.89 ± 0.83	55.22 ± 0.41	72.34 ± 0.51	71.57 ± 0.88	74.05 ± 0.57	72.31 ± 0.38
BN	68.03 ± 3.10	43.51 ± 0.32	56.25 ± 0.76	71.17 ± 0.47	71.80 ± 0.40	74.50 ± 0.55	71.62 ± 0.86
CR	75.04 ± 0.63	42.60 ± 2.11	54.84 ± 0.94	73.55 ± 0.22	71.35 ± 1.21	73.28 ± 0.61	72.06 ± 0.43
cw-CR	77.01 ± 0.58	46.50 ± 1.05	57.85 ± 0.64	73.60 ± 0.62	71.46 ± 1.01	74.07 ± 0.59	72.23 ± 0.88
VR	78.56 ± 0.88	46.10 ± 0.97	57.67 ± 0.57	**75.04 ± 0.26**	73.39 ± 0.47	74.99 ± 0.41	73.94 ± 0.72
cw-VR	78.42 ± 0.21	**48.07 ± 1.09**	57.00 ± 0.95	74.19 ± 0.64	**73.54 ± 0.25**	75.58 ± 0.31	73.81 ± 1.35
L1R	**79.37 ± 0.50**	47.61 ± 0.99	**59.08 ± 0.33**	74.51 ± 0.61	72.19 ± 0.43	74.87 ± 0.52	73.51 ± 0.96
RR	73.54 ± 0.25	42.91 ± 1.08	55.65 ± 1.09	73.42 ± 0.66	73.13 ± 0.58	**76.08 ± 0.37**	**74.20 ± 0.85**
cw-RR	73.71 ± 0.41	42.45 ± 0.46	55.29 ± 1.59	73.38 ± 0.77	72.88 ± 0.46	75.66 ± 0.27	73.90 ± 0.59

Table 8: Condition experiment results for the CIFAR-100 CNN model. The best performing regularizer in each condition (each column) is shown in bold, and other regularizers whose performance range overlaps with the best one are highlighted in green. For the default condition, the standard values of data size=50k, layer width=128, and number of classes=100 were used. For the other columns, all the conditions were the same as the default, except for the condition indicated on the top part of the columns. Regularizers were applied to the fully-connected layer.

Reg.	Default	Data Size		Layer Width		Number of Classes	
		1k	5k	32	512	16	64
Baseline	38.74 ± 0.52	9.11 ± 0.30	17.79 ± 0.72	37.59 ± 0.34	38.70 ± 0.64	54.25 ± 0.73	41.98 ± 0.40
L1W	39.03 ± 0.64	8.67 ± 0.37	17.70 ± 0.60	37.77 ± 0.58	39.08 ± 0.47	54.92 ± 1.53	41.92 ± 1.18
L2W	39.77 ± 0.31	9.47 ± 0.39	17.95 ± 0.70	37.22 ± 0.36	38.45 ± 0.99	54.72 ± 1.59	42.53 ± 0.66
Dropout	36.12 ± 0.72	$\mathbf{9.78 \pm 0.48}$	18.32 ± 0.81	35.92 ± 0.99	35.69 ± 0.37	54.27 ± 1.57	40.86 ± 0.46
BN	39.07 ± 0.39	8.82 ± 0.36	17.99 ± 0.58	37.82 ± 1.49	37.84 ± 0.57	55.45 ± 1.43	42.28 ± 0.66
CR	40.12 ± 0.50	8.30 ± 0.14	17.53 ± 0.41	$\mathbf{39.53 \pm 0.63}$	39.30 ± 0.94	55.45 ± 1.10	43.24 ± 0.86
cw-CR	42.97 ± 0.73	9.15 ± 0.29	18.71 ± 0.62	38.59 ± 0.67	41.98 ± 0.25	56.50 ± 1.21	45.76 ± 0.64
VR	42.32 ± 0.94	8.57 ± 0.32	18.15 ± 0.38	38.65 ± 0.45	43.13 ± 0.74	57.67 ± 1.03	45.68 ± 0.40
cw-VR	43.25 ± 0.64	9.55 ± 0.22	$\mathbf{18.97 \pm 0.57}$	39.33 ± 0.59	43.09 ± 0.73	$\mathbf{58.62 \pm 0.53}$	45.77 ± 1.06
L1R	$\mathbf{43.97 \pm 0.81}$	8.85 ± 0.35	18.02 ± 0.47	38.89 ± 0.31	$\mathbf{43.54 \pm 0.62}$	57.49 ± 1.43	$\mathbf{46.35 \pm 1.00}$
RR	37.32 ± 0.35	8.80 ± 0.27	18.68 ± 0.36	31.46 ± 0.46	40.71 ± 0.32	55.84 ± 0.80	39.75 ± 0.35
cw-RR	37.38 ± 0.31	9.38 ± 0.34	18.43 ± 0.14	31.89 ± 0.31	40.75 ± 0.29	55.90 ± 0.65	39.97 ± 0.41