Representational Structure of Neural Networks Trained on Biased and Out-Of-Distribution Data

Gnyanesh Bangaru[1]
cgnyanesh@gmail.com
Lalith Bharadwaj Baru[*,1,2]
lalithbharadwaj313@gmail.com
Kiran Chakravarthula[1]
kiran_c@vnrvjiet.in

[1] VNR Vignana Jyothi Institutions of Engineering and Technology, Hyderabad-90, T.S, India.
[2] IHub-Data, IIIT Hyderabad, Hyderabad-08, T.S, India.

Abstract

Neural networks trained on standard image classification data sets are observed to be less robust to distributional shifts and pertain to certain levels of bias in representations. Thus, it is pertinent to identify the kind of objective function that could correspond to better performance for data with biases and distribution shifts, and how can that objective function be justified to be the apt choice. There is, however, less literature that focuses on the choice of the objective function and its representational structure when trained on such data sets. In this work, we analyse the performance and the internal representational structure of convolution-based neural networks (eg. ResNets) trained by varying objective functions on biased and out-of-distribution (OOD) data. Specifically, we interpret similarities in representations (using CKA) acquired for distinct objective functions (probabilistic and margin-based) and provide a detailed analysis of the chosen ones. Our analysis reports that representations acquired by ResNets using Softmax Cross-Entropy ($\mathcal{L}_{SCE}$) and Negative Log-Likelihood ($\mathcal{L}_{NLL}$) as objectives are equally competent in providing superior performance and fine representations on OOD and biased data. Subsequently, we interpret that the ResNets are less likely to be robust on cross-data generalisation without refined representational similarity.
Code: *https://github.com/gnyanesh-bangaru/loss-analysis-cka*

1 Introduction

The advent of deep learning has provided us with robust models and the finest neural architectures. Even with tremendous performance, most neural networks do not discriminate representations well as they are trained in a controlled setting [25]. They thus tend to perform poorly with distributional shifts and biases in the data. Various methods with neural network optimisation or a new paradigm of solving were implied to mitigate this issue [2], [9], [21], [28].

[*] This work was done when *Lalith* was a final year undergrad student at VNRVJIET.

Dataset Kind	Dataset Name	Classes	Train-Val-Test
Generic	MNIST^S	10	50k-NA-10k
Generic	CIFAR-10^S	10	50k-NA-10k
Generic	ImageNet-200	200	100k-10k-10k
Biased	C-MNIST^S	10	55k-5k-10k
Biased	CIFAR-C^S	10	45k-4.8k-10k
OOD	MNIST-M^S	10	59k-NA-90k
OOD	ImageNet-R	200	NA-NA-30k

Table 1: The above table illustrates three kinds of data sets containing i.e. generic, biased, and OOD samples. 'NA' is specified to denote the absence of samples for that particular split. The tag X^S denotes that the data set X is considered to be small.

In the existing literature, the work of Simon *et al.* [16] has provided insight on the transferability of representations when exposed to various loss functions, but has focused mainly on probabilistic objectives. Kim *et al.* [13] utilised the gradient reversal layer and proposed a novel regularised loss function based on mutual information obtained from InfoGANs [6] to unlearn the target bias, that is, the bias present in the data. This eventually minimises the detrimental effects of bias in the data. Adeli *et al.* [1] proposed a loss function for adversarial training with two objectives to learn features that have maximum discriminative power and minimal statistical dependencies with protected bias. Certain works provided novel optimisation methods to solve the bias problem [2], [21]. StableNet by Zhang *et al.* [28] reduces the statistical relation between irrelevant and relevant features acquired from the data. To achieve this, they have implied a sample weighting technique and trained on a new objective which is modified on SCE. Under different settings, StableNet reported superior performance for various OOD datasets including MNIST-M. Sunil *et al.* [27] proposed a new method of solving the OOD problem using the *principle of abstention* (which encourages to predict the sample classes which were unseen by the model) to provide OOD generalisation.

So, we question thus: what could be an optimal objective function for data with biases and distributional shifts? Eventually, what kind of representations are learnt by the neural networks when exposed to such data? How are the internal representations of neural networks altered by varying the data samples? Finally, how transferable are the representations that are trained with data biases and data with distributional shifts?

Hence our work is motivated to address the above problem and provide some insights by conducting extensive analysis. The existing literature is less evident on the use of objective functions on data with biases and distributional shifts. The representational characteristics of a given objective function are not specifically studied. The transferability of the representations—produced by training neural networks on biased or OOD data—to standard classification data is not illustrated. Hence, these concerns motivate us to understand the behaviour of neural networks on various data sets, mainly biased and OOD. First, we study the empirical performance of the objective functions by dividing them into two variants: a) probabilistic, and b)margin-based. Secondly, we analyse the importance of individual objective functions which provide representation structure with good generalisation and transferability at an interpretable perspective. Additionally, we see the potential of CKA as one of the criterion to measure the Interpretability of the neural networks [12].

2 Setup

Terminology The input data is represented as $X \in \{x^{(1)}, x^{(2)}, \ldots, x^{(n)}\}$ where n is the total number of samples. The encoder $Enc(.)$ is used to extract features from a given input X. The features extracted from the encoder are presented as $fv \leftarrow Enc(X)$; where $fv \in \{fv^{(1)}, fv^{(2)}, \ldots, fv^{(n)}\}$. The dimensions of the feature vector vary by changing the encoder. As most of the experiments were carried out using a supervised framework, the data sets do have certain ground truth labels, and these are represented as $Y \in \{y^{(1)}, y^{(2)}, \ldots, y^{(n)}\}$. The activation functions, sigmoid and softmax, are indicated by $\sigma(a_i) = 1/(1+e^{-a_i})$ and $\mathcal{S}(a_i) = e^{a_i}/\sum_j e^{a_j}$, respectively. The loss (objective) function, is denoted by $\mathcal{L}_{(.)}$ and the suffixes indicate its specified variant. The norms $\|\cdot\|_1$ and $\|\cdot\|_2$ indicate Manhattan and Euclidean norms[1] respectively.

Datasets As mentioned, we choose the data for solving three diverse problems. First, we consider well-studied and experimented datasets such as MNIST [19], CIFAR-10[18] and ImageNet-200. These data sets are widely used for standard image classification tasks; also, they do not have inherent biases and do not contain OOD samples. Next, Colored MNIST (C-MNIST) and Corrupted CIFAR-10 (C-CIFAR) [20] data sets are utilized for understanding biased representations. For both the datasets, C-MNIST and C-CIFAR, we perform experiments according to Lee *et al.* [20]. Lastly, we consider data with distributional shifts such as ImageNet Renditions (ImageNet-R) [8] and Modified MNIST (MNIST-M) [7]. The collection of data sets considered for experimentation are tabulated in Table 1.

Models In this work, we use ResNet18 and ResNet50 to understand convolution-type representations. ResNet18 is used for small data, which are illustrated in Table 1, similarly, for medium-sized data, we imply ResNet50. We have considered standard ResNets with global average pooling. The fully connected layers for ResNet18 and ResNet50 are [512-c] and [2048-512-c] respectively (Where c denotes the number of classes). Intermediate dropout layers are used with a drop rate of 40% and a set constant for all data sets to have a fair evaluation.

Training and fine-tuning To train each model we utilized Adam [14] as an optimizer with a standard learning rate of 10^{-3} and a weight decay of 10^{-5}. For certain executions, we utilised pre-trained ImageNet weights and fine-tuned the models on that specific data set (and when not particularly mentioned, we trained the model from scratch). We fed 512 samples in batches to neural networks by varying the objective functions. The early stopping criterion is embedded with the patience of 12 epochs to ensure the model does not over-fit with excessive training. The results reported in Tables 3 and 4 are produced without any augmentations (except normalisation) and pre-trained weights.

3 Empirical Analysis

The choice of the objective function to train a deep neural network, on specified data remains a question. Janocha *et al.* [11] provided a theoretical justification and conducted experiments

[1] Suppose, $a \in R^n$ then, $\|a\|_1 = \sum_i^n |a_i|$; $\|a\|_2 = \sqrt{\sum_i^n a_i^2}$

Objective Function	Equation
Softmax Cross-Entropy ($\mathcal{L}_{SCE}$)	$\mathcal{L}_{SCE}(fv^{(i)}, y^{(i)}) = -\sum_{c=1}^{C} y_c^{(i)} \ \texttt{log}\left(\mathcal{S}(fv_c^{(i)})\right)$
Binary Cross-Entropy ($\mathcal{L}_{BCE}$)	$\mathcal{L}_{BCE}(fv^{(i)}, y^{(i)}) = y^{(i)}\texttt{log}(\sigma(fv^{(i)})) + (1-y^{(i)})\texttt{log}(1-\sigma(fv^{(i)}))$
Negative Log-Likelihood ($\mathcal{L}_{NLL}$)	$\mathcal{L}_{NLL}(fv^{(i)}, y^{(i)}) = -\sum_{c=1}^{C} y_c^{(i)} \ \texttt{log}\left(fv_c^{(i)}\right)$
Mean Absolute Error ($\mathcal{L}_1$)	$\mathcal{L}_1(fv^{(i)}, y^{(i)}) = -\sum_{c=1}^{C} \lVert y_c^{(i)} - \mathcal{S}(fv_c^{(i)}) \rVert_1$
Mean Squared Error ($\mathcal{L}_2$)	$\mathcal{L}_2(fv^{(i)}, y^{(i)}) = -\sum_{c=1}^{C} \lVert y_c^{(i)} - \mathcal{S}(fv_c^{(i)}) \rVert_2^2$
Sum-of-Squares ($\mathcal{L}_{SoS}$)	$\mathcal{L}_{SoS}(fv^{(i)}, y^{(i)}) = \frac{1}{C}\sum_{c=1}^{C} \left[\alpha y_c^{(i)}(fv_c^{(i)} - \beta) + (1-y_c^{(i)})(fv_c^{(i)})^2\right]$

Table 2: The above table illustrates all the objective functions that are experimented with in this work.

on MNIST pointing out the importance of $\mathcal{L}_1$ and $\mathcal{L}_2$ not just as regularizers, but as objective functions for better generalisations. Hui *et al.* [10] empirically proves that square loss with a little parametric tuning would produce significant results for most tasks of natural language processing (NLP) and automatic speech recognition (ASR). Hui *et al.* [10] specifically mentioned that the proposed square loss is not brittle for randomised initialisation. A recent analysis by Simon *et al.* [16] provides insights noting that the representations acquired to classify certain tasks with more class separation lead to poor transferable features. This work implies various objective functions to observe both the performance and quality of representations for the standard computer vision classification data sets.

The previous literature focuses on the training and transferability of features acquired by training standard neural architectures with varying objective functions. But, there is sparse literature noting the relevance of both probabilistic and margin-based objective functions on data with biases and distributional shifts. Hence, we provide empirical analysis for two variants of objective functions to understand the performance of each objective function on various data sets mentioned in Table 1.

3.1 Probabilistic Objectives

Probabilistic objective functions calculate the error that approximates the underlying probabilities for representations acquired from an encoder (ResNet). In this paper, we include three probabilistic objectives and they are detailed in Table 2. First, the Softmax cross-entropy [5], a highly used objective, is obtained by applying softmax activation in the final layer of the neural network, and this feed is minimised by the negative log-likelihood (NLL). Next, Binary cross-entropy (BCE) is obtained by applying sigmoid activation ($\sigma(\cdot)$) at the final layer of neural networks and this information is minimised by NLL. Primarily, the BCE loss function is used for binary classification problems but, a recent work empirically proves that its implication on multi-class would lead to better performance [3] by applying the one-vs-rest strategy. Finally, the likelihood provides the joint probability of the sample distribution and minimises the negative logarithm of the obtained likelihood [4].

3.2 Margin-based Objectives

Margin-based objective functions calculate the error by discriminating the representations extracted from an encoder (ResNet). Similarly to probabilistic objectives, we include three margin-based objectives, which are detailed in Table 2. First, the mean absolute error ($\mathcal{L}_1$ objective function) finds the Manhattan distance between the two representations. We acquire

Objectives	Variants	Generic			Biased		OOD		Mean
		CIFAR-10	MNIST	ImageNet-200	C-MNIST	C-CIFAR	MNSIT-M	ImageNet-R†	
Probabilistic	$\mathcal{L}_{SCE}$	82.69± 0.79	99.63± 0.30	**52.14±1.76**	95.31±1.21	34.44±1.84	95.75±0.36	**27.17±0.00**	**69.63**
	$\mathcal{L}_{BCE}$	82.63± 0.13	99.75± 0.17	37.38±1.66	93.75±1.67	32.47±4.34	96.56±0.47	2.7±0.00	63.60
	$\mathcal{L}_{NLL}$	80.50± 1.97	**99.75± 0.34**	51.84±0.30	**95.81±1.36**	**35.25±1.79**	95.94±0.60	23.91±0.00	69.01
Margin-based	$\mathcal{L}_1$	82.00± 0.60	99.35± 0.17	1.73±0.18	95.19±2.64	25.11±1.21	**97.56±1.81**	1.81±0.33	57.54
	$\mathcal{L}_2$	**84.44± 1.30**	99.75± 0.17	38.84±3.41	93.00±0.99	25.34±2.46	97.06±1.87	1.89±0.87	62.97
	$\mathcal{L}_{SoS}$	82.12± 1.52	99.63± 0.01	36.13±2.30	95.00±0.63	34.81±1.42	97.44±1.08	2.44±0.81	63.94

Table 3: The table below provides the empirical performance of the individual objective functions for the generic, bias, and OOD data. The experiments were carried out without any augmentations and did not use learnt weights (ImageNet1K or ImageNet21K) for the training models. These experiments were conducted three times for each objective function for a fair evaluation. The tabulated mean and standard deviation (mean ± std) in each cell depicts the accuracy scores obtained after experimenting thrice with the 'test' data. **Bold** and underline represent the accuracy scores of **first** and second best performing models, respectively.

the theoretical motivation of Janocha *et al.* [11] that $\mathcal{L}_1$ would reduce the sparseness in the representations. Rather than directly discriminating the representations in the final layer, we use softmax to ensure appropriate learning without saturation of partial derivatives[2]. Similar to $\mathcal{L}_1$, we use $\mathcal{L}_2$ to find find the Euclidean distance between two representations (final layer). Lastly, Hui *et al.* [10] rescaled SoS to be more robust by providing two parameters α, β. Injection of these parameters resulted in a decent performance for the NLP and ASR tasks, but was poorly performed on the computer vision tasks. For experimentation, we have chosen $\alpha, \beta = 1$, and this reduces to standard SoS.

3.3 Evaluation

Now, let us understand the empirical performance of these objective functions trained on all variants of the data detailed in Table 3. For generic data, in most of the cases, $\mathcal{L}_{SCE}$ and $\mathcal{L}_{NLL}$ were able to achieve top accuracy scores. But, $\mathcal{L}_2$ obtained the highest accuracy for CIFAR-10 and competed closely with $\mathcal{L}_{NLL}$ on MNIST. Taking into account the case of biased data $\mathcal{L}_{NLL}$ performed standalone; $\mathcal{L}_{SCE}$ and $\mathcal{L}_{SoS}$ were able to compete closely with $\mathcal{L}_{NLL}$. Surprisingly, $\mathcal{L}_1$ turned out to have good performance for MNIST-M, and $\mathcal{L}_{SoS}$ was closely on par. But for ImageNet-R except $\mathcal{L}_{SCE}$ and $\mathcal{L}_{NLL}$ all other objectives failed to obtain at least 10% accuracy score. Further, the variance in the accuracy is higher for margin-based objectives compared to that of probabilistic. Hence, aggregating these results, it is strongly recommended that using probabilistic loss functions ($\mathcal{L}_{SCE}$ and $\mathcal{L}_{NLL}$) would obtain decent performance on most of the data. The empirical performance attained by these objective functions is well-understood but, the question arises with the internal representations of the model trained on these data.

4 Representation Analysis

In systems neuroscience, the Representation Similarity Analysis (RSA) framework [17] was the most successful in understanding the representations acquired from various activity

[2] Without any non-linear activation it is observed that gradients saturate and halt the learning of neural networks. The experiments conducted with pure $\mathcal{L}_1$ i.e, without any non-linearity led to very poor learning and these have been experimented and detailed in the supplementary material

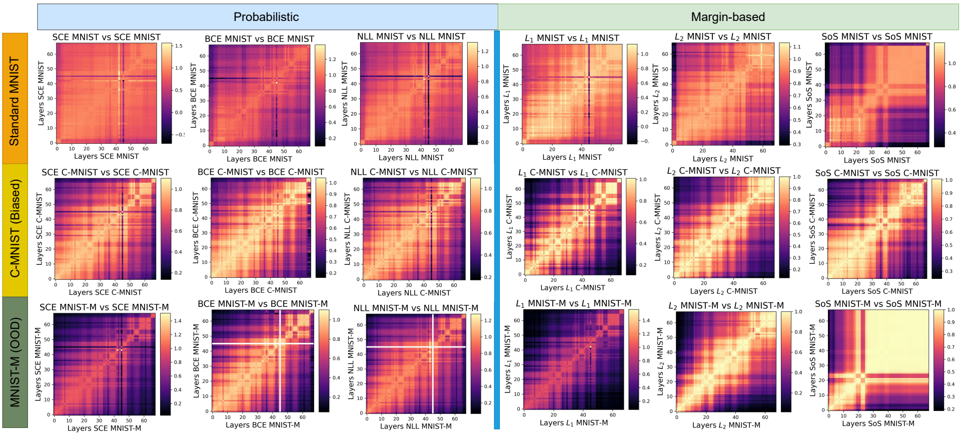

Figure 1: This figure illustrates CKA visualizations for probabilistic objective functions on MNIST, C-MNIST, and MNIST-M. The **First row**, **Second row**, and **Third row** describe the representations acquired by all objectives respectively for all three variants of data sets. Each tile of the image consists of a color, indicating the strength of representations i.e., the similarity of each layer representation. This corresponding color map is provided on the right side of each image tile. For visualizing CKA, we considered all the layers of the neural network (ResNet18) including activation, normalization, and fully-connected layers. The matrix is formed by comparing the features acquired by each layer of the ResNet18 with itself.

patterns in various regions of the brain. In addition, it provides insights into the similarity between representations by constructing Representation Dissimilarity Matrices (RDMs). With this motivation a method *Centered Kernel Alignment (CKA)* was devised to understand the representation structure of artificial neural networks. It is well established in the literature that CKA [15] acquires qualitative representations compared to PwCCA [22] and SVCCA [24]. CKA not only captures the correspondence between the representations of a neural network but also allows us to compute the similarity between pairs of layers. Therefore, three major interpretations can be made by visualising CKA. Which are

1. Which part of the layers, of a certain neural architecture, constitute similar representations?

2. Comparing the representations obtained from two diverse neural architectures, and determine which could be a robust choice for given data.

3. Comparing the internal representations of a certain neural architecture by varying the data sets, we determine which architecture would be sensitive to a certain data.

To reduce the computational expense consumed by linear CKA, mini-batch CKA is applied by computing the mean of HSIC (Hilbert-Schmidt Independence Criterion) scores on selected mini-batches (N). This strategy is implemented straightforwardly as Thao *et al.* [23]. The mini-batch CKA is detailed as follows:

$$\mathrm{CKA}^{mini} = \frac{\sum_{i=1}^{N} \mathrm{HSIC}_i(\tilde{X}_i, \tilde{Y}_i)}{\sqrt{\sum_{i=1}^{N} \mathrm{HSIC}_i(\tilde{X}_i, \tilde{X}_i)}\sqrt{\sum_{i=1}^{N} \mathrm{HSIC}_i(\tilde{Y}_i, \tilde{Y}_i)}} \tag{1}$$

where, $\tilde{X}_i = X_i X_i^T; \tilde{Y}_i = Y_i Y_i^T$. These $X_i \in R^{n\times d_1}$ and $Y_i \in R^{n\times d_2}$ are activation matrices for ith mini-batch of examples without replacement. We now try to analyse the representations using CKA for all the objective functions on the aforementioned data in two steps.

4.1 Step-I

The first step aims at addressing which layers correspond to similar representations in a specific neural network trained on a certain objective function. In the following, we are going to understand which representations would lead to better outcomes and which do not. For this, we intend to choose all the objective functions for representation analysis using CKA. As including all the data sets would be redundant, we have chosen MNIST variants i.e., choosing a similar type of data set from the variants mentioned in Table 1.

In Figure 1, when considering MNIST data, the CKA representation matrix formed for $\mathcal{L}_{BCE}$, $\mathcal{L}_{NLL}$ and $\mathcal{L}_2$ seems to have similar characteristics. While considering the case for the C-MNIST data set, all the loss functions tend to form a small box-like structure at the ultimate layers (after 50th layer). But, $\mathcal{L}_{NLL}, \mathcal{L}_1$, and $\mathcal{L}_{SCE}$ seem to have uniformly distributed representation with decreasing similarity with the depth of the neural network. Finally, for MNIST-M data the refined representation similarity is obtained for $\mathcal{L}_1, \mathcal{L}_{SCE}$ and $\mathcal{L}_{NLL}$. However, surprisingly, the performance for MNIST-M is higher for margin-based objectives. Considering the case of $\mathcal{L}_{SoS}$ objective, it is prone to have *block structured* representations on utmost all the data [23]. This structured block resembles the neural network as *overparameterised* model. The underlying reason is either that the model has fewer data samples or a deeper network. This can be surmounted by truncating the layers with identical representational similarity.

These indications are clear to note that, the objectives $\mathcal{L}_{SCE}$ and $\mathcal{L}_{NLL}$ not only provide decent empirical performance but capture fine representations with ResNets. Hence, it is understood that the representations acquired by probabilistic objectives are comparatively better to provide good generalisations for diverse data sets.

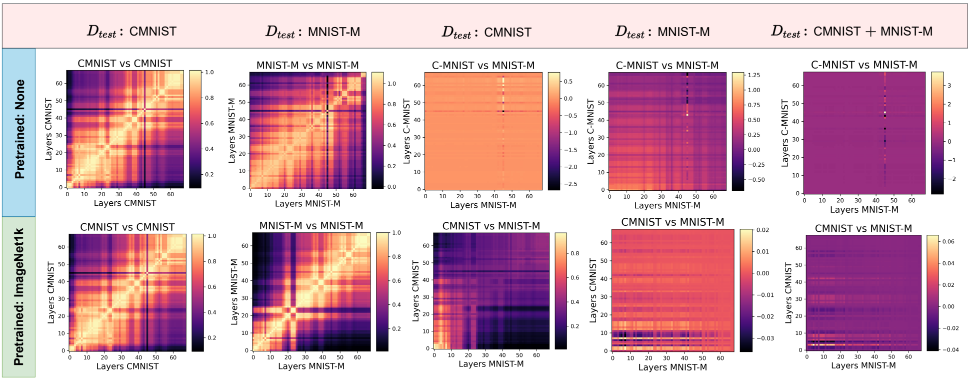

Figure 2: The figure visualizes the CKA plots for C-MNIST and MNIST-M on ResNet18 with and without ImageNet1k pre-trained weights. All these visualizations were obtained by training the model with $\mathcal{L}_{SCE}$ as objective.

4.2 Step-II

Following this, a new question arises. How similar are the representations that initialise the training with pre-trained weights (*e.g.* ImageNet1k) on the considered data sets? How similar are the representations of neural networks trained on biased data versus OOD data? Can we transfer neural architecture weights trained on biased data sets to solve the problem of distribution shifts and vice versa?

Train Test	pre-trained: **ImageNet1k**		pre-trained: **None**	
	C-MNIST	MNIST-M	C-MNIST	MNIST-M
C-MNIST	95.69±0.78	97.62 ±0.26	95.34 ±1.48	13.87 ±2.41
M-MNIST	52.40 ±3.72	98.03 ±0.56	11.45 ±0.09	95.83 ±0.55

(a) C-MNIST vs MNIST-M on $\mathcal{L}_{SCE}$

Train Test	pre-trained: **ImageNet1k**		pre-trained: **None**	
	CIFAR-10	C-CIFAR	CIFAR-10	C-CIFAR
CIFAR-10	90.19±0.55	22.85 ±2.47	82.71 ±0.97	15.62 ±2.40
C-CIFAR	24.66 ±2.13	38.72 ±3.07	24.45 ±2.54	34.42 ±2.24

(b) CIFAR-10 vs C-CIFAR on $\mathcal{L}_{SCE}$

Train Test	pre-trained: **ImageNet1k**		pre-trained: **None**	
	C-MNIST	MNIST-M	C-MNIST	MNIST-M
C-MNIST	96.80 ±0.76	96.01 ±1.57	95.82 ±1.65	12.64 ±0.53
M-MNIST	50.41 ±5.58	98.28 ±0.55	11.61 ±0.26	95.95 ±0.73

(c) C-MNIST vs MNIST-M on $\mathcal{L}_{NLL}$

Train Test	pre-trained: **ImageNet1k**		pre-trained: **None**	
	CIFAR-10	C-CIFAR	CIFAR-10	C-CIFAR
CIFAR-10	91.29±1.17	18.35 ±1.91	80.50 ±2.41	15.31 ±3.07
C-CIFAR	28.94 ±5.67	38.35 ±4.38	23.47 ±7.35	34.92 ±1.94

(d) CIFAR-10 vs C-CIFAR on $\mathcal{L}_{NLL}$

Table 4: These tables illustrate the performance of ResNets for both in-data and cross-data generalization. All the results depicted in the table are test results that were experimented thrice and the obtained the mean and standard deviation of test accuracy scores are noted. The 'Train' and 'Test' technically mean that data were trained on the mentioned dataset and the accuracy scores were obtained on the test datasets. We consider both the cases of training ResNets with and without pre-trained weights and as a note, ImageNet1k weights are used as pre-trained weights.

To address these questions, we first perform an empirical analysis on objectives $\mathcal{L}_{SCE}$ and $\mathcal{L}_{NLL}$. Now, we consider two combinations of data sets. From Table 4 (a) and Table 4 (c) we infer that in the presence of ImageNet1k weights the ResNet trained on C-MNIST and tested on M-MNIST provides half transferable performance ($\approx 50\%$) but, the accuracy drops if C-MNIST is trained from scratch. In the same line, the presence of ImageNet1k weights the ResNet trained on M-MNIST shows maximum transferability ($> 95\%$) and the accuracy diminishes if trained from scratch. But is counterintuitive for the combination of CIFAR-10 and C-CIFAR. It can be observed that, even with ImageNet1k weights, the transferable performance is minimal. Especially, when trained on CIFAR-10 and tested on C-CIFAR the transferable performance of $\mathcal{L}_{SCE}$ is small-scale ($< 1\%$). Hence, there is no guarantee to attain greater performance on biased data and data with distribution shifts for standard neural networks by initializing the training with pre-trained weights (*e.g.* ImageNet1k).

Now we compare and contrast the representational structure of the neural networks having initialised with pre-trained weights and those that are trained totally from scratch (without any augmentations). In Figure 2 we examine that when the network is trained and tested on the same data (that is, ResNet18 trained on MNIST-M and tested on its test set), the representations are refined and similarity decreases as the network progress with depth. If you consider the case of cross-data generalisation, the neural network with a uniform progression of decrease in similarity is highly likely to perform well. *E.g.* Consider the case of ResNet trained on C-MNIST and evaluated on MNIST-M. The resulting performance was superior only when the C-MNIST training was initialised with ImageNet1k weights. In this case, when ResNet is trained from scratch, it has poor performance, and this is reflected in the CKA representation matrix. As most of the layers tend to be the same, the ResNet (trained on C-MNIST) could not generalize well for the given new set of samples (tested on MNIST-M). So, if the neural network generalises well or provides good transferability, its performance can be assessed by visualising CKA and comprehending that the concentration of representational similarity progressively decreases with the depth of the network.

5 Conclusion

By summarising the above, we infer that $\mathcal{L}_{SCE}$ and $\mathcal{L}_{NLL}$ objectives would be apt for most of the data sets (inclusive of Biased and OOD). But while experimenting, it should be noted that the variance in accuracy must be minimalist to ensure robustness. Secondly, if neural networks are initialised with pre-trained weights there is no guarantee for superior performance on biased and OOD data. Finally, if the neural networks are exposed to cross-data, generalisation is attained only when the layers of CKA matrices have a progressive dissimilarity with the depth of the network.

In the future, we see the potential requirement of data sets comprising samples with distribution shifts with a greater sample size as of ImageNet. Likewise, the representations acquired by the models are to be ensured with the least bias possible. We believe that, comprehending the representations acquired from biased data would aid researchers in providing a novel debiasing neural network or a bias mitigation strategy. Also we believe that, this CKA framework can be extended to study the detailed intrepretability of the neural networks [12]

6 Broader Impact

Since scientists and analysts routinely conclude, albeit based on evidence-based insights, OOD and biased data can be inherently misleading. The impact of such misinterpretation can be corrected through exposure, experience, and expertise when humans are involved in the interpretation. However, if the data itself is OOD or biased or only the interpretations of the data were presented by researchers, the danger of unconscious or even conscious biases cannot be ruled out entirely. While Smith's book [26] is an elaborate example of how research methodologies are inherently designed with ignorance towards the subjects of study, subtler cases of bias can be innate to the ways data are sourced, stored, pre-processed, analyzed, and interpreted.

The current work emphasises the need to methodically improve the quality of OOD representation and biased data without proposing radical paradigm shifts in current methodologies. This work does not organically provide scope for misinterpretation of data or biased decisions made through data analytics. Furthermore, the work facilitates a better and more uniform representation of the data by reminding researchers to consciously consider the biases and OOD aspects of the data, which may be rather inconspicuous. A stronger motivation arises as Artificial Intelligence and Machine Learning continue to be used in various technology and social domains for diverse applications.

References

[1] Ehsan Adeli, Qingyu Zhao, Adolf Pfefferbaum, Edith V Sullivan, Li Fei-Fei, Juan Carlos Niebles, and Kilian M Pohl. Representation learning with statistical independence to mitigate bias. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2513–2523, 2021.

[2] Hilal Asi, Yair Carmon, Arun Jambulapati, Yujia Jin, and Aaron Sidford. Stochastic bias-reduced gradient methods. *Advances in Neural Information Processing Systems*, 34, 2021.

[3] Lucas Beyer, Olivier J Hénaff, Alexander Kolesnikov, Xiaohua Zhai, and Aäron van den Oord. Are we done with imagenet? *arXiv preprint arXiv:2006.07159*, 2020.

[4] Christopher M Bishop et al. *Neural networks for pattern recognition*. Oxford university press, 1995.

[5] John S Bridle. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In *Neurocomputing*, pages 227–236. Springer, 1990.

[6] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in neural information processing systems*, 29, 2016.

[7] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015.

[8] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8349, 2021.

[9] Rui Huang, Andrew Geng, and Yixuan Li. On the importance of gradients for detecting distributional shifts in the wild. *Advances in Neural Information Processing Systems*, 34, 2021.

[10] Like Hui and Mikhail Belkin. Evaluation of neural architectures trained with square loss vs cross-entropy in classification tasks. *ICLR*, 2021.

[11] Katarzyna Janocha and Wojciech Marian Czarnecki. On loss functions for deep neural networks in classification. *arXiv preprint arXiv:1702.05659*, 2017.

[12] Kohitij Kar, Simon Kornblith, and Evelina Fedorenko. Interpretability of artificial neural network models in artificial intelligence vs. neuroscience. *arXiv preprint arXiv:2206.03951*, 2022.

[13] Byungju Kim, Hyunwoo Kim, Kyungsu Kim, Sungjin Kim, and Junmo Kim. Learning not to learn: Training deep neural networks with biased data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9012–9020, 2019.

[14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[15] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pages 3519–3529. PMLR, 2019.

[16] Simon Kornblith, Ting Chen, Honglak Lee, and Mohammad Norouzi. Why do better loss functions lead to less transferable features? *Advances in Neural Information Processing Systems*, 34, 2021.

[17] Nikolaus Kriegeskorte, Marieke Mur, and Peter A Bandettini. Representational similarity analysis-connecting the branches of systems neuroscience. *Frontiers in systems neuroscience*, page 4, 2008.

[18] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[19] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

[20] Jungsoo Lee, Eungyeup Kim, Juyoung Lee, Jihyeon Lee, and Jaegul Choo. Learning debiased representation via disentangled feature augmentation. *Advances in Neural Information Processing Systems*, 34, 2021.

[21] Yi Li and Nuno Vasconcelos. Repair: Removing representation bias by dataset resampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9572–9581, 2019.

[22] Ari Morcos, Maithra Raghu, and Samy Bengio. Insights on representational similarity in neural networks with canonical correlation. *Advances in Neural Information Processing Systems*, 31, 2018.

[23] Thao Nguyen, Maithra Raghu, and Simon Kornblith. Do wide and deep networks learn the same things? uncovering how neural network representations vary with width and depth. *ICLR*, 2021.

[24] Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. *Advances in neural information processing systems*, 30, 2017.

[25] Jacob Russin, Randall C O'Reilly, and Yoshua Bengio. Deep learning needs a prefrontal cortex. *Work Bridging AI Cogn Sci*, 107:603–616, 2020.

[26] Linda Tuhiwai Smith. *Decolonizing methodologies: Research and indigenous peoples*. Bloomsbury Publishing, 2021.

[27] Sunil Thulasidasan, Sushil Thapa, Sayera Dhaubhadel, Gopinath Chennupati, Tanmoy Bhattacharya, and Jeff Bilmes. An effective baseline for robustness to distributional shift. In *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 278–285. IEEE, 2021.

[28] Xingxuan Zhang, Peng Cui, Renzhe Xu, Linjun Zhou, Yue He, and Zheyan Shen. Deep stable learning for out-of-distribution generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5372–5382, 2021.