Decoding Generalization from Memorization in Deep Neural Networks

Anonymous authors
Paper under double-blind review

Abstract

Overparameterized Deep Networks that generalize well have been key to the dramatic success of Deep Learning in recent years. The reasons for their remarkable ability to generalize are not well understood yet. It has also been known that Deep Networks possess the ability to memorize training data, as evidenced by perfect or high training accuracies on models trained with corrupted data that have class labels shuffled to varying degrees. Concomitantly, such models are known to generalize poorly, i.e. they suffer from poor test accuracies, due to which it is thought that the act of memorizing substantially degrades the innate ability to generalize. It has, however, been unclear why the poor generalization that accompanies such memorization, comes about. One possibility is that during training, the layers of the network irretrievably re-organize their representations in a manner that makes generalization difficult. The other possibility is that the network retains significant latent ability to generalize, but the trained network somehow "chooses" to readout in a manner that is detrimental to generalization. Here, we provide evidence for the latter possibility by demonstrating, empirically, that such models possess information in their representations for substantially-improved generalization. Furthermore, such generalization abilities can be easily decoded from the internals of the trained model, and we build a technique to do so. We demonstrate results on multiple models trained with standard datasets.

1 Introduction

Prior to the advent of Deep Learning, the conventional wisdom for long[1], was that in building a predictive model, the model should have as few parameters as possible and this number should certainly be less than the number of training samples that one was fitting. The dogma was that, otherwise, the model would exactly fit the training points, but invariably generalize poorly to unseen data, i.e. overfit. This intuition was also largely borne out by the models of the day. Modern Deep Learning, however, has gone on to show the opposite, namely that overparameterized models not only don't necessarily overfit, but that they can generalize remarkably well to unseen data. However, over a decade later, we still do not satisfactorily understand why this is so. Interestingly, it has been shown (Zhang et al., 2017; 2021) that when one randomly shuffles class labels of data points from standard training datasets to varying degrees, Deep Networks can still have high/perfect training accuracy when trained on such corrupted training data; however, this appears to typically be accompanied by poor performance on unseen test data (that have true labels). This phenomenon has been called *memorization*, since it is thought that the model rote-learned the training data without acquiring the ability to generalize to unseen examples. It has been suggested that progress on understanding memorization could enable a better understanding of generalization in Deep Networks trained on real-world data (Zhang et al., 2017; 2021) and indeed that a detailed understanding of mechanisms of generalization should also be able to explain the phenomenon of memorization.

An open question arising in this context is about the detailed mechanisms that lead to poor generalization in models trained with shuffled labels, i.e. models that memorize. A natural hypothesis governing such

[1] von Neumann famously said, "With four parameters I can fit an elephant, and with five I can make him wiggle his trunk." (Dyson et al., 2004)

mechanisms, stated informally, is that, during training, the network organizes its internal representations, in a manner suited to doing well on the (corrupted) training data. Since this data is significantly noisy, on being given unseen data with true labels, it fundamentally lacks the ability to have good prediction performance, leading to poor generalization. An alternative hypothesis is that layerwise representations in the network retain the ability to generalize easily, but that the network somehow chooses to readout in favor of high training accuracy in a manner that incidentally causes poor generalization performance. A corollary to this alternative hypothesis is that one ought to be able to construct a decoder (i.e. a probe) for the outputs of the network's layers that has better generalization performance, i.e. higher test accuracy.

Here, surprisingly, we show evidence for this alternative hypothesis. In particular, we study the organization of subspaces of class-conditioned training data on layerwise outputs, in Deep Networks. We estimate these subspaces using Principal Components Analysis (PCA). In order to remain agnostic to the information decoded by subsequent layers, we build a simple probe that leverages the geometry of the present layer's output of an incoming datapoint, relative to these class-conditioned subspaces. Specifically, we measure the angle between this output vector and its projection on each of these class-conditioned subspaces and the probe predicts the datapoint's class to be the class whose subspace has the minimum such angle. We call this probe the Minimum Angle Subspace Classifier (MASC). Notably, unlike probes used conventionally (e.g. in (Alain & Bengio, 2018)) whose parameters are determined by iteratively minimizing a crossentropy loss, the parameters of MASC are directly determined from the subspace geometry of the training data. A schematic illustrating the geometry of MASC is presented in Figure 1 of the Supplementary Material.

We train a number of Deep Networks with standard datasets in the memorization setting. Here, a randomly-chosen fraction of training data points have their labels changed to a randomly-chosen label from the available labels in the dataset. We do so for differing fractions of the training dataset and – consistent with previous work (Zhang et al., 2017; 2021; Arpit et al., 2017) – see that training with such corrupted training datasets causes correspondingly poor test accuracies in the model. However, MASC – which uses the internals of the network to predict the class label – tends to do significantly better than the model on the test set. A schematic illustration of the memorization setting with MASC is shown in Figure 1.

We outline a more detailed summary of our main contributions below.

1. For models trained with standard methods & datasets with training data corrupted by label noise to varying degrees, we demonstrate that MASC applied on at least one layer, when using subspaces corresponding to such corrupted training data, has significantly better test accuracy than the model. For example, MASC outperforms the model test accuracy by upto 159.93%, 188.99% and 64.86% on MLP-MNIST, CNN-Fashion-MNIST and AlexNet-CIFAR-100 respectively. A more detailed account of these numbers is in Table 1.

2. For the aforementioned models, if the true training class labels are known post hoc, i.e. after the model is trained, we can build MASC using subspaces corresponding to true class labels. These MASC classifiers usually have better generalization performance than in (1). For example, MASC using true labels outperforms the model by upto 198.43%, 212.42% and 337.51% on MLP-MNIST, CNN-Fashion-MNIST and AlexNet-CIFAR-100 respectively. A more detailed account of these numbers is in Table 5 of the Supplementary Material. This demonstrates that the layers of the memorized network maintain representations in a manner that is amenable to straightforward generalization to a degree not previously recognized.

3. Conversely, we asked if a model trained on true training labels similarly retained internal representations that have the capability to memorize easily, as manifested by MASC. Adapting our technique to this setting, we create corrupted training sets which we use to build MASC. In this setting, we find that we can extract a high degree of memorization, in some cases. The results are presented in Section 6 of the Supplementary Material.

4. Finally, leveraging the MASC classifiers built in (1) and (2), we ask, if we can retrain the memorized model for a few epochs to achieve better model generalization. We find that indeed, in many cases, there is an improvement in the generalization performance of the model.

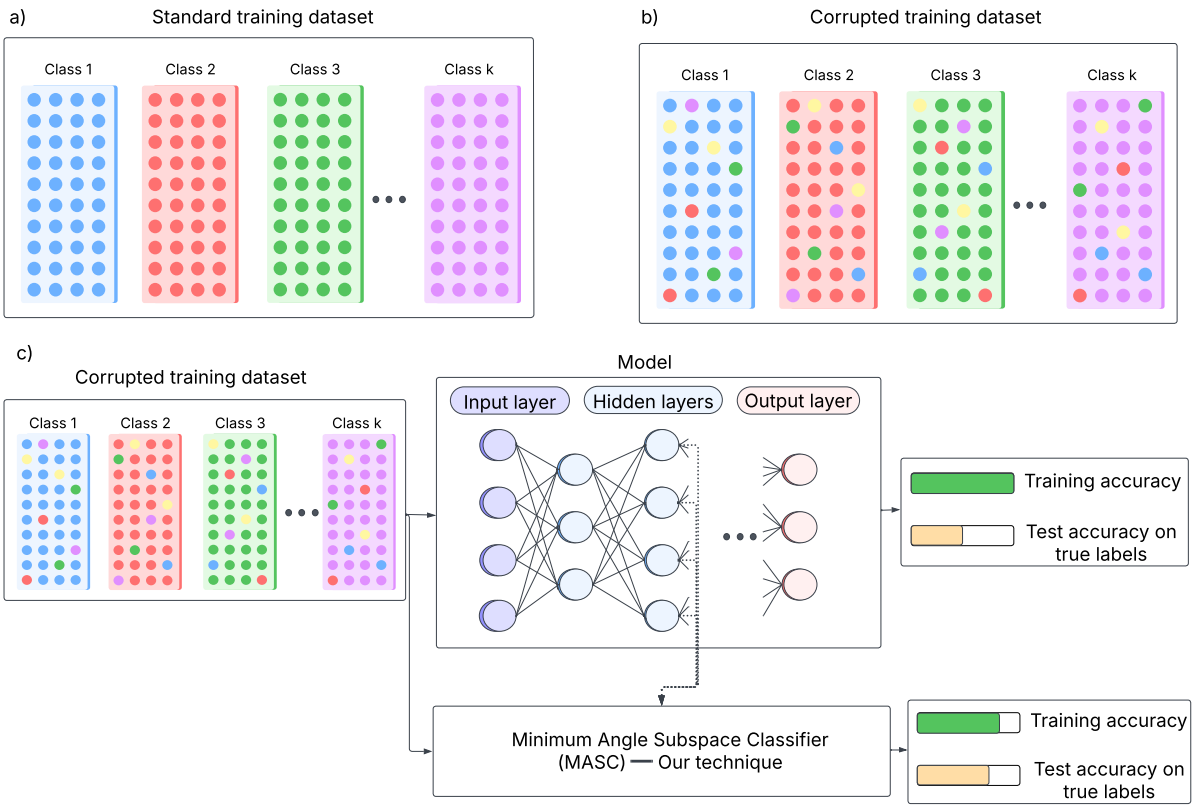

Figure 1: A schematic of the memorization setting used in our work and the application of the MASC classifier in it. a) Illustration of a standard training dataset. b) A corrupted training dataset is created by changing the labels of the standard training dataset with a specific probability (due to which a few of the colors are changed, representing the changed labels). Changing the labels happens uniformly at random for the whole dataset. c) A Deep Network is trained with this corrupted training dataset to achieve ∼100% training accuracy, which is usually accompanied by poor test accuracy (as measured on true labels from the test set). We have shown that the Minimum Angle Subspace Classifier (MASC) – our technique – which uses the internals of the Deep Network, tends to have significantly better generalization (test accuracy) than the Deep Network itself.

2 Related Work

The idea of probing intermediate layers of Deep Networks isn't new. For example, kernel-PCA (Montavon et al., 2011) with RBF kernels has been used to analyze layerwise evolution of representations of Deep Networks. In that work, they quantify the quality of layerwise representations and find that the last layers of the network tend to have representations that are more simple and accurate than previous layers. Likewise, linear classifier probes (Alain & Bengio, 2018) have been used to study the roles and dynamics of intermediate layers in Deep Networks. There, they show that the degree of linear separability increases over the layers of the network. However, they explicitly avoid examining memorized networks (Zhang et al., 2017) because they thought such probes would inevitably overfit. Our results are therefore especially surprising in this context, because we demonstrate, on the contrary, that intermediate representations, in fact, tend to resist overfitting, to a degree not previously recognized.

It has been known (Arpit et al., 2017) that early on in training, memorized networks (Zhang et al., 2017) start off by having better generalization ability; however generalization worsens as training accuracy increases across epochs of training. Leveraging this observation, there have been efforts to build training algorithms

that are designed to extract better generalization in the case where the data is known to be noisy (Jiang et al., 2018; Han et al., 2018; Liu et al., 2020). Stephenson et al (Stephenson et al., 2021) investigate memorized models, suggesting that memorization predominantly occurs in the later layers. This is based, in part, on the observation that rewinding early-layer weights to their early-stopping values can recover generalization, whereas rewinding later-layer weights does not yield the same effect. In general, the thinking in the field has been that while there is an initial peak in generalization ability, it is lost during further training, although one can mitigate some of this loss by modifying training (Jiang et al., 2018) or by rewinding a subset of weights to their early values. On the contrary, our results suggest that layerwise outputs of Deep Networks retain significant ability to generalize after training and we demonstrate that this generalization can be extracted without modifying the weights of the trained network that are obtained via standard training methods.

An important line of theoretical research on deep linear models has explored the question of generalization (Saxe et al., 2013). Here, a theoretical explanation for the phenomenon of memorization in networks trained with noisy labels has been proposed (Lampinen & Ganguli, 2018).

Studies have investigated training dynamics across layers using various forms of Canonical Correlation Analysis (Raghu et al., 2017), including analyses in both generalized and memorized networks (Morcos et al., 2018). Centered Kernel Alignment has been employed to examine the effects of different random initializations (Kornblith et al., 2019), as well as to study network similarity between models trained on the same data with different initializations (Kornblith et al., 2019). Additionally, experiments have explored the use of representational geometry measures to understand the dynamics of layerwise outputs (Chung et al., 2016; Cohen et al., 2020), along with other structural measures such as curvature dimensionality (Hénaff et al., 2019), which aim to capture underlying properties of learned representations (Sussillo & Abbott, 2009; Farrell et al., 2019; Gao & Ganguli, 2015; Litwin-Kumar et al., 2017; Bakry et al., 2015; Cayco-Gajic & Silver, 2019; Yosinski et al., 2014; Stringer et al., 2019).

To address label noise, various heuristic approaches have been proposed (Khetan et al., 2017; Scott et al., 2013; Reed et al., 2014; Zhang & Sabuncu, 2018; Malach & Shalev-Shwartz, 2017), particularly in the context of classification tasks (Frénay et al., 2014; Ren et al., 2018; Menon et al., 2018; Shen & Sanghavi, 2019). In the case of overparameterized models, Li et al (Li et al., 2020) demonstrate that memorization requires the network weights to deviate significantly from their initial random state in order to overfit noisy labels. Additionally, in a theoretical model of epochwise double descent (Stephenson & Lee, 2021), it has been suggested that for smaller models, moderate levels of label noise can lead to a reduction in generalization error at later stages of training.

3 Methods

3.1 Minimum Angle Subspace Classifier Algorithm (MASC)

For a given data point $\boldsymbol{x}$ from the training or test set, a layer output data point $\boldsymbol{x_l}$ from layer l when input $\boldsymbol{x}$ is passed through the network and its corresponding training subspaces $\{S_k\}_{k=1}^{K}$ with K classes, we use Minimum Angle Subspace Classifier (MASC) Algorithm 1 for predicting class labels $y(\boldsymbol{x_l})$.

Algorithm 1 Minimum Angle Subspace Classifier (MASC)

Input: Training subspaces $\{S_k\}_{k=1}^{K}$, layer output data point $\boldsymbol{x_l}$ from layer l when input $\boldsymbol{x}$ is passed through the network and classes $\{C_k\}_{k=1}^{K}$.
Output: MASC prediction class label $y(\boldsymbol{x_l})$ according to layer l .

1: **for** each class C_k **do**
2: $\quad \boldsymbol{x_{lk}} \longleftarrow$ compute the projection of $\boldsymbol{x_l}$ onto subspace S_k.
3: $\quad$ Compute the angle $\theta(\boldsymbol{x_l}, \boldsymbol{x_{lk}})$ between $\boldsymbol{x_l}$ and $\boldsymbol{x_{lk}}$
4: **end for**
5: Assign the label $y(\boldsymbol{x_l}) = C_k$ where $k = \arg\min_k \theta(\boldsymbol{x_l}, \boldsymbol{x_{lk}})$
6: **Return:** label $y(\boldsymbol{x_l})$

Given training dataset $\mathcal{D}\{(\boldsymbol{x_i}, y_i)\}_{i=1}^{m}$, where each $\boldsymbol{x_i} \in \mathbb{R}^d$ and $y_i \in \{C_k\}_{k=1}^{K}$ are input-label pairs, we estimate training subspaces $\{S_k\}_{k=1}^{K}$ for all classes K, for a given layer l of the neural network using Algorithm 2 and 3. In practice, S_k is represented via its principal components, which form a basis for the subspace. We have used 99% as the percentage of variance explained by the principal components, unless otherwise mentioned.

Algorithm 2 Subspaces Estimator for MASC

Input: Training dataset $\mathcal{D}\{(\boldsymbol{x_i}, y_i)\}_{i=1}^{m}$, where each $\boldsymbol{x_i} \in \mathbb{R}^d$ and $y_i \in \{C_k\}_{k=1}^{K}$ are input-label pairs, neural network, and layer l.
Output: Subspaces $\{S_k\}_{k=1}^{K}$ for classes K, for given layer l.

1: $\mathcal{D}_l = \phi$
2: **for** each input pair $(\boldsymbol{x_i}, y_i)$ in $\mathcal{D}$ **do**
3: Pass $\boldsymbol{x_i}$ through the network layers to obtain the output of layer l, denoted as $\boldsymbol{x_l} \in \mathbb{R}^{ld}$.
4: $\mathcal{D}_l = \mathcal{D}_l \cup \{(\boldsymbol{x_l}, \boldsymbol{y_i})\}$
5: **end for**
6: Estimated subspaces $\{S_k\}_{k=1}^{K} \longleftarrow$ **PCA-Based Subspace Estimation**$(\mathcal{D}_l)$
7: **Return:** Subspaces $\{S_k\}_{k=1}^{K}$

Algorithm 3 PCA-Based Subspace Estimation

Input: Layer output $\mathcal{D}_l = \{(\boldsymbol{x_i}, y_i)\}_{i=1}^{m}$, where $\boldsymbol{x_l} \in \mathbb{R}^{ld}$ and $y_i \in \{C_k\}_{k=1}^{K}$.
Output: Subspaces $\{S_k\}_{k=1}^{K}$ for classes K.

1: $\mathcal{D}_{\text{new}} \leftarrow \mathcal{D}_l$
2: **for** each $(\boldsymbol{x_i}, \boldsymbol{y_i}) \in \mathcal{D}_l$ **do**
3: $\mathcal{D}_{\text{new}} \leftarrow \mathcal{D}_{\text{new}} \cup \{(-\boldsymbol{x_i}, \boldsymbol{y_i})\}$
4: **end for**
5: **for** each $k \in \{1, \ldots, K\}$ **do**
6: Extract the subset of data $\mathcal{D}_{\text{new},k} = \{\boldsymbol{x_i} \mid y_i = C_k\}$
7: $S_k =$PCA$(\mathcal{D}_{\text{new},k})$
8: **end for**
9: **Return:** Subspaces $\{S_k\}_{k=1}^{K}$

While the subspaces are estimated using the training data alone, accuracy of the Minimum Angle Subspace Classifier is determined for the training data and the test data separately. This process is followed for all the layers in the network independently. MASC is using labels of the dataset while creating the class-specific subspaces. The process of creation and use of subspaces with MASC for a new data point are shown schematically in Figure 1 of the Supplementary Material.

3.2 Experimental Setup

We have used multiple models and datasets, namely Multi-layer Perceptron (MLP) trained on MNIST (Deng, 2012) and CIFAR-10 (Krizhevsky, 2009) datasets, Convolutional Neural Networks (CNN) [2] trained on MNIST, Fashion-MNIST (Xiao et al., 2017), and CIFAR-10 and AlexNet (Krizhevsky et al., 2012) trained on CIFAR-100 (Krizhevsky, 2009) and Tiny ImageNet (Moustafa, 2017). We have trained these models with training data having true labels ("generalized models") as well as separately using training data with labels shuffled to varing degrees ("memorized models") (Zhang et al., 2017; 2021).

For memorized models, when we say we train it with corruption degree p, we mean that with probability p, we attempt changing the label for each training datapoint. Changing the labels happens uniformly at random to any of the class labels. Note that this may result in the label remaining the same; therefore the expected fraction of datapoints whose labels changed are $p - p/K$, where K is the number of class labels. So, e.g. for a dataset with 10 classes, this would mean that for corruption degrees of 20%, 40%, 60%, 80%

[2]The CNN models were built along the lines of (Tran et al., 2022).

and 100%, the expected percentage of training datapoints with changed labels is 18%, 36%, 54%, 72% and 90% respectively. We have run experiments for values of p being 0% (generalized model), and memorized models with p being 20% , 40%, 60%, 80% and 100% .

A summary of the models, datasets, training set sizes, and number of parameters is provided in Table 1 of the Supplementary Material. Tables 2 and 3 in the Supplementary Material report the average training and test accuracies of all models over three runs. Additional details on the models, hyperparameters, and training procedures are also included in the Supplementary Material. The general terminology used in this work is also explained in the Supplementary Material.

Following standard practice in studying memorized models (e.g. Stephenson et al. (2021)), we do not use explicit regularizers such as Dropout or batchnorm, or early stopping, unless otherwise mentioned, as a result of which our baseline test accuracy numbers are often much lower than what is usually found with standard training of these models. All the models are trained to either reach very high training accuracies (i.e. $99\% - 100\%$) or trained until 500 epochs. Some models did not reach such high accuracies, in which case, results have been shown on the model obtained at epoch 500. We trained 3 instances of each model and results displayed are averaged over these instances with the shaded region indicating the range of results also indicated in the plots.

Once the model is trained, we apply MASC on each layer of the network with respect to different subspaces. For MLP models, all the MASC experiments were performed for all the layers in the network including on the input (after it is pre-processed). For CNN models and AlexNet models, the experiments were performed on flatten layer (Flat) and fully connected layers (FC). While we ran the experiments on the input layer for CNNs, we did not do so for AlexNet.

3.3 Leveraging MASC to retrain the model

Here, the idea is to use one of the layerwise MASC classifiers in order to relabel the corrupted training set. This relabeled training set is then used to retrain the existing model. To determine the layer whose MASC classifier we will use, we find the layer whose MASC classifier generalizes best. To this end, we first split the test data set into 80%-20%. We use the MASC accuracy on the corrupted subspaces in the 20% of the test dataset to identify the model's best-layer. Then, using the best-layer MASC predictions, we relabel the corrupted training dataset. We train with the relabeled corrupted training dataset for upto 30 epochs and perform early stopping with patience of 3 by considering the 20% test dataset as a validation dataset. A similar process was followed while working with subspaces corresponding to true labels. The test accuracy on the models is calculated with respect to the 80% test dataset, obtained in the aforementioned split. A schematic of the retraining process using MASC is shown in Figure 2 in the Supplementary Material.

4 Enhanced innate generalization ability in memorized models

Models trained with corrupted labels have high training accuracy (on corrupted labels) while also having low accuracy on the test set with true labels (Zhang et al., 2017; 2021). We ask if we can decode the representations of the hidden layers of these memorized models to obtain better generalization.

To do so, we build a probe that we call Minimum Angle Subspace Classifier (MASC) using class-conditioned corrupted training subspaces obtained from the memorized models' hidden layer outputs. MASC is performed layer-wise for the layers of the network independently. More details on MASC are available in the Methods section. MASC accuracy on corrupted training data, MASC accuracy on original training data (with true labels), and MASC accuracy on test data (with true labels) over the layers of MLP trained on MNIST, CNN trained on Fashion-MNIST and AlexNet trained on CIFAR-100, for various randomly-chosen fractions of label corruption in training data (i.e. corruption degrees) are shown in Figure 2. Likewise, results for MLP trained on CIFAR-10, CNN trained on MNIST & CIFAR-10 and AlexNet trained on Tiny ImageNet are presented in Figure 12 (Supplementary Material).

Importantly, for every corrupted model we have (with non-zero corruption degree), except those with 100% corruption degree, we find that our Minimum Angle Subspace Classifier (MASC) in at least one layer

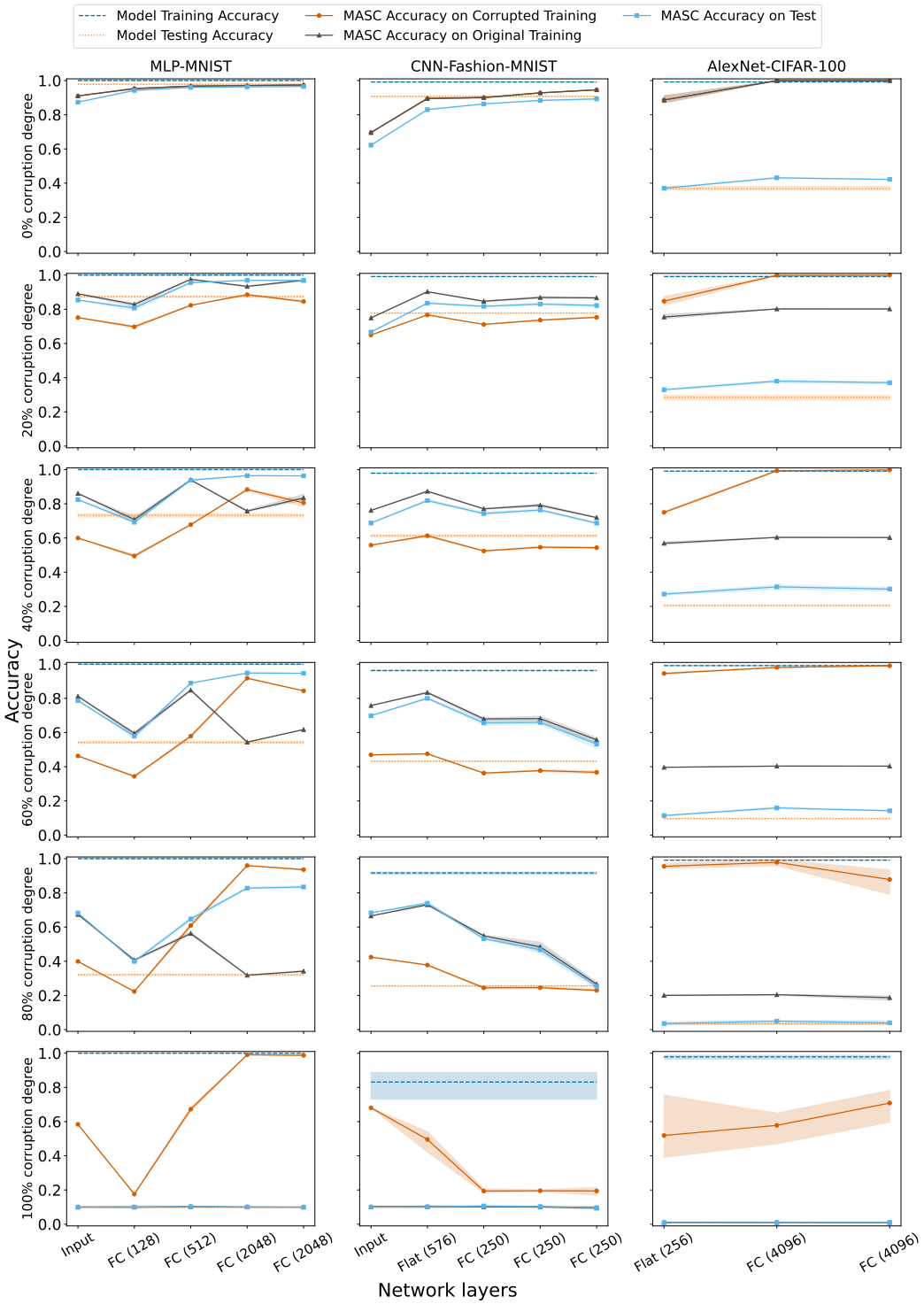

Figure 2: Minimum Angle Subspace Classifier (MASC) accuracy over the layers of the network when the data is projected onto corrupted training subspaces with the indicated corruption degree, for multiple models/datasets. Rows corresponds to plots with the same corruption degree & the columns correspond to the models, as noted. Training accuracy (dashed line) & test accuracy (dotted line) of the model is shown. FC corresponds to fully connected layer with *ReLU* activation whereas Flat corresponds to flatten layer without *ReLU* activation. The number of class-wise PCA components of these models are shown in Figure 15 the Supplementary Material. *SGD* optimizer (Qian, 1999) was used for training MLP models, whereas *Adam* optimizer (Kingma, 2014) was used for other models.

has better test accuracy than the corresponding model itself. Table 1 reports by what percentage the MASC classifier outperformed the model for the best such layer, for each model. In many cases, the MASC test accuracy is dramatically better than that of the model. This is remarkable, because, in addition to the layerwise outputs, MASC used precisely the same information (including the same corrupted training dataset) that was available to the model itself, and yet is able to extract better generalization. This suggests that the model retains significant latent generalization ability, which is not captured in its own test-set performance. In many models, the same MASC, especially on the later layers, also approaches perfect accuracy on the corrupted training set, indicating that this improved generalization can happen concurrently with memorization of training data points with shuffled labels. Below, we make more specific observations on the performance of the models.

Table 1: Percentage by which the MASC classifier (run on the best layer) outperformed the model's test accuracy when the data is projected onto corrupted training subspaces. The best layer corresponds to the one that has the highest measured MASC test accuracy among the layers for the said model/dataset. The accuracies in each case are averaged over three runs. Some of the detailed results are available in Supplementary Material, as indicated.

Corruption degree	**20%**	**40%**	**60%**	**80%**
MLP-MNIST	10.93%	31.63%	75.04%	159.93%
MLP-CIFAR-10 (Supplementary)	9.90%	24.42%	46.97%	64.75%
CNN-MNIST (Supplementary)	9.81%	37.03%	98.69%	201.06%
CNN-Fashion-MNIST	7.49%	33.50%	84.93%	189.00%
CNN-CIFAR-10 (Supplementary)	2.29%	6.26%	27.03%	60.17%
AlexNet-CIFAR-100	33.58%	53.10%	64.86%	45.00%
AlexNet-Tiny ImageNet (Supplementary)	27.50%	53.46%	45.38%	14.16%

With generalized models i.e. those with 0% corruption degree, at the later layers of the network, it is observed that in most of the cases MASC accuracy on training data approaches the models training accuracy. Similarly, MASC accuracy on test dataset is comparable to or performed better than the models' test accuracy.

Even for high corruption degrees, we find that MASC performs well. For example, with 80% corruption degree, which implies that approximately 72% of the training labels have been changed, we observed good MASC test accuracy in many cases. Notably, the MASC test accuracy on the later layers is over 80% on MLP-MNIST, in comparison to 34% test accuracy by the model. Similarly, MASC test accuracy on one of the layers is about 75% for CNN-Fashion-MNIST, in contrast to 25% model test accuracy. Even for larger models/datasets such as AlexNet-CIFAR-100, MASC test accuracy outperforms the model test accuracy by 45%, for training sets with 80% corruption degree.

Not only does MASC have better accuracy than the model on the test data but, when applied to some layers, it also does well on the training data with the true labels. Although the model has memorized the training data with corrupted labels, outputs from certain layers have the ability to predict the trained true labels. For example, in MLP-MNIST, for low to moderate degrees of corruption, MASC on the middle layer (FC (512)) has good accuracy on the true training labels, while also retaining good accuracy on the test set. With 40% corruption degree, approximately 36% are changed labels and yet the model has good accuracy on the true training labels in at least one layer of the network. e.g. MLP-MNIST has over 90% true training accuracy at layer FC(512), CNN-Fashion-MNIST has approximately 85% in Flat (576) layer & AlexNet-CIFAR-100 has approximately 60% in FC (4096) layer. This means that almost 20% of those labels are predicted correctly even though the model was trained for 500 epochs or has reached high training accuracy on corrupted labels. In the process of doing this, the model does not have any direct information about the true labels and neither does MASC.

One way to think about a Deep Network, is as one that successively transforms input representations in a manner that aids in good prediction performance. Therefore, performance of MASC on the input is a good baseline measure to assess if subsequent layers have favorable accuracies. Naively, for models trained with

corrupted data, one would expect layered representations that enable the model to do well on the corrupted training data, but not do well on the test/training data that have true labels. While this expectation seems to hold with respect to the model itself, we find that the layer-wise representations do not necessarily follow this expectation. That is, MASC applied to subsequent layers, often have better true training accuracy and test accuracy than MASC applied to the input, suggesting that the Deep Network does indeed transform the data in a manner amenable to better generalization, even if its labels are dominated by noise.

5 Generalization via true training labels with memorized models

While the previous section demonstrated improved generalization performance by MASC, here, we investigate if there exist subspaces that can offer even better generalization performance. To this end, we consider the setting where the true label identities of the training set are known, after training with corrupted labels is complete. Can we extract significantly high training as well as test performance in this case from the layerwise outputs of the network? To do so, we build MASC using subspaces obtained from training data with true labels. It is a priori unclear if MASCs trained in this manner will have high accuracy. Since the network trained assuming different labels for many of the datapoints, it is conceivable that class-wise subspaces corresponding to true labels lack structure and predictive power. We find, however, that these possibilities do not bear out.

MASC accuracy on original training data and on test data projected on true training label subspace over the layers of the same networks is shown in Figure 3 and Figure 13 (Supplementary Material). For comparison, MASC accuracy on corrupted training data and test data projected on corrupted training subspace is also shown. We find that, in many cases, accuracies on the true training labels, as well as the test set are dramatically better here than with the experiments where subspaces were determined for the corrupted training data. In Table 5 (Supplementary Material), we show by what percentage the MASC classifier outperformed the model for the best layer for corruption degrees 20%, 40%, 60% and 80%. In fact, the MASC test accuracies for the corrupted models (with non-zero corruption degree) are sometimes fairly close to the test accuracy of the uncorrupted model.

Notably, even for models trained with 100% corruption degree, in most cases, the MASC retains significant accuracy on the true training labels as well as the test set. This is in spite of the fact that the model itself has chance-level test-set accuracy. For example, MASC classifier has 95% test accuracy in the last FC(2048) layer for MLP-MNIST, 69% test accuracy for Flat(576) layer in CNN-Fashion-MNIST, and 4% test accuracy for Flat(256) layer in AlexNet-CIFAR-100.

The results here are proof of principle that suggest the existence of subspaces which allow one to extract significantly high generalization performance on models trained with datapoints whose labels are shuffled to a remarkably high degree. This has two implications. On the one hand, it demonstrates that models trained with very high label noise, surprisingly, retain the latent ability to generalize very well. On the other hand, it suggests that development of new techniques to identify favorable subspaces could help markedly boost generalization performance of models, whose training data is known to have label noise.

The Supplementary Material also describes a control experiment with MASC accuracies on a random initialization of the network, as well as comparison with early stopping test accuracies. We have results corresponding to MLP trained on CIFAR-10, CNN trained on MNIST and CIFAR-10, and AlexNet trained on Tiny ImageNet in the Supplementary Material for all the experiments. We also have a section comparing MLP models trained on MNIST and CIFAR-10 with SGD and Adam optimizer.

In section 6 of the Supplementary Material, we also investigating the latent memorization capabilities of uncorrupted models. Here, conversely, we ask how well a network trained on true labels can manifest memorization of an arbitrary relabeling of its training data. More specifically, we built a MASC classifier on a model trained on true training labels, with the goal of memorizing training data whose labels are corrupted to varying degrees post hoc. Interestingly, we find a dichotomy in model behavior here, with some models trained on specific datasets having the propensity to memorize to a high degree, whereas others not demonstrating such ability.

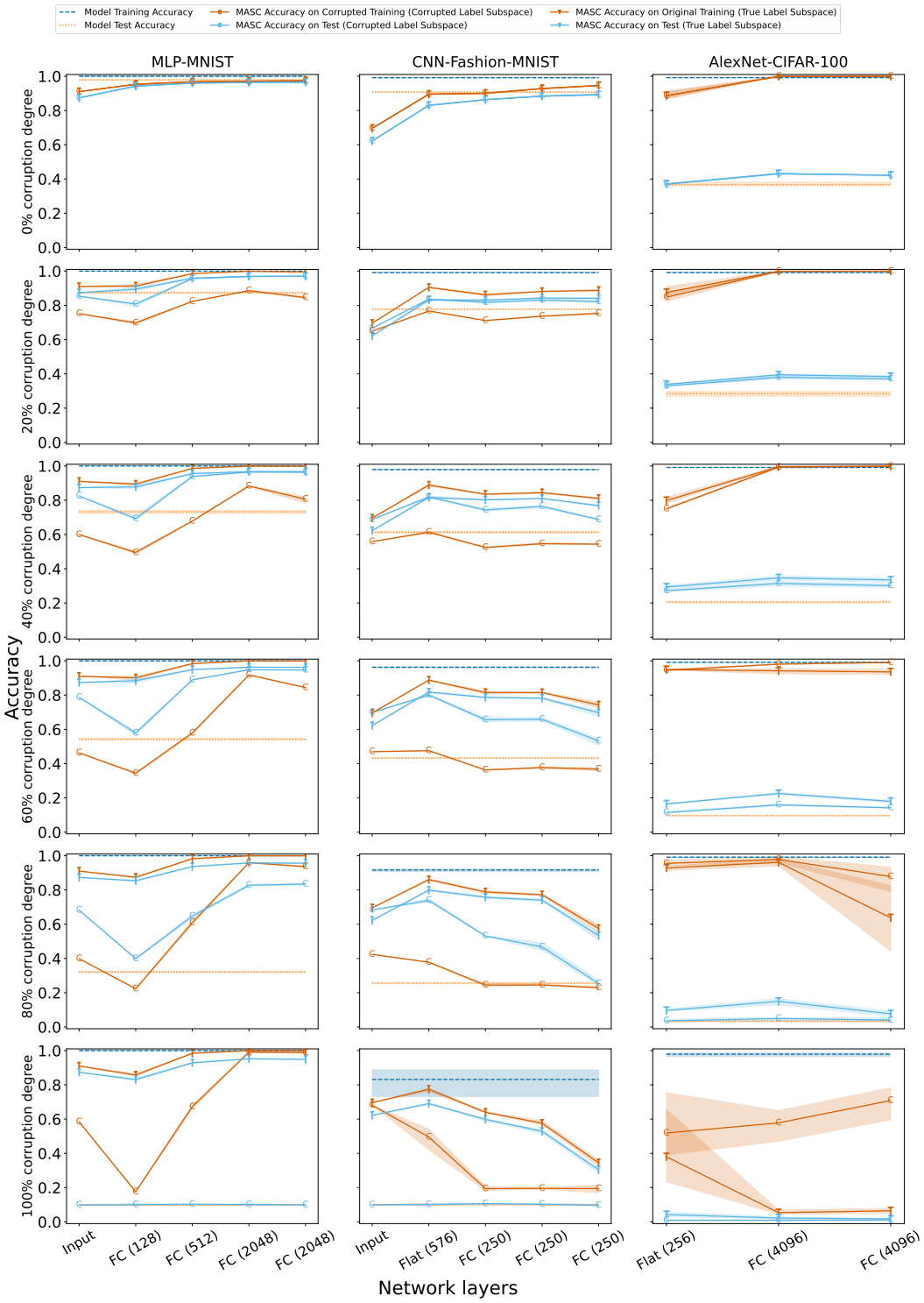

Figure 3: Minimum Angle Subspace Classifier (MASC) accuracy over the layers of the network when the data set is projected onto corrupted subspace and subspace corresponding to true training labels. Rows corresponds to plots which have the same corruption degree and the columns correspond to the models as noted. Training and test accuracy of the model is shown. FC corresponds to fully connected layer with ReLU activation whereas Flat corresponds to flatten layer without ReLU activation. The respective number of class-wise PCA components for true training label subspaces of the models is shown in Figure 17 of the Supplementary Material. *SGD* optimizer was used for training MLP models, whereas *Adam* optimizer was used for other models.

6 Leveraging MASC to retrain the base memorized model for improved generalization

Taking into account the better generalization ability using MASC on memorized models, in this section, we ask the following question. Can we use the MASC classifier to retrain the existing model to achieve better generalization?

Details of the pipeline for retraining existing models, leveraging MASC are already presented before. For different corruption degrees, test accuracy before and after retraining with relabeled data using MASC (corrupted and true subspaces) for MLP-MNIST, MLP-CIFAR-10, CNN-MNIST, CNN-Fashion-MNIST, CNN-CIFAR-10 models are shown in Figure 4. Similar results for AlexNet-CIFAR-100, AlexNet-Tiny ImageNet models are shown in Figure 27 of the Supplementary Material.

In order to study the dynamics of accuracy during retraining, unencumbered by the early stopping criterion, we also performed a similar experiment without using early stopping, for 10 epochs. The results for model before training, model after retraining on MASC corrupted subspace, and model after retraining on MASC subspace with true labels over the 10 epochs for all model-dataset pairs with various corruption degrees are shown in Figure 31 of the Supplementary Material.

In Figure 4 and 27 (Supplementary Material), we find that for some models (MLP-MNIST, CNN-MNIST, CNN-Fashion-MNIST, CNN-CIFAR-10) with non-zero corruption degrees, there is an improvement in the test accuracy of models retrained using relabelling with MASC on corrupted subspaces, in comparison to the models' test accuracy before retraining (existing models). Indeed, in some cases, the improvement is quite significant, especially for larger corruption degrees (that are below 100% corruption degree).

However, for some models (MLP-CIFAR-10, AlexNet-CIFAR100), the accuracy gains due to such retraining appear marginal. Indeed, in some cases (MLP-CIFAR10, AlexNet-Tiny ImageNet for 20% corruption degree), there is a decrease in the test accuracy with such retraining. In order to study why, we checked the fraction of incorrect labels in the relabeled training dataset and compared it with the same measure for the existing corrupted training dataset. For the corruption degrees 20%, 40%, 60%, and 80%, these results are plotted in Figure 28 in the Supplementary Material. Table 6 (Supplementary Material) lists the exact values of the same.

In particular, it turns out that for MLP-CIFAR-10 and AlexNet-CIFAR-100 this fraction is almost equal to the fraction on the existing corrupted dataset; for MLP-CIFAR-10 the fraction is marginally higher for the relabeled dataset and for AlexNet-CIFAR-100, it is marginally lower. This simply implies that the MASC classifier on the best layer that uses corrupted subspaces does roughly as well on the training set with true labels as the existing model, while surprisingly, the same MASC classifier is able to perform significantly better than the existing model on the test-set with true labels (See light orange bar in Figure 4 and 27 (Supplementary Material)). With regard to retraining, it would seem that the relabeled training isn't more effective than the existing corrupted training set in training the model, which possibly reflects in the lack of significant improvement in test accuracy. More broadly, this suggests that MASC's better generalization isn't necessarily accompanied by better training set performance on true labels. This phenomenon requires a more detailed future investigation.

Secondly, for AlexNet-Tiny ImageNet with 20% corruption degree, the relabeled training set has significantly fewer fraction of correct labels (92.46% lower) than in the existing corrupted training set. We wanted to determine to what extent this lower fraction is driven by previously incorrect label predictions (in the existing corrupted training dataset) being predicted correctly (in the relabeled set), versus previously correct predictions being relabeled incorrectly. For all models, these numbers are visualized in Figures 29 & 30 and Tables 7 & 8 (Supplementary Material) list corresponding numbers. We find for the case of AlexNet-Tiny ImageNet with 20% corruption degree that there is a small fraction (5.94%) of previously incorrect labels that are correctly relabeled and a large fraction (24.46%) of correctly labeled points that are incorrectly labeled by the MASC classifier trained on the corrupted label subspaces. Notably, even though this MASC classifier while doing significantly worse on the training data than the model happens to do markedly better than the model in test accuracy. As before, we think that the poor performance of the retrained model is driven by the fact that relabeling results in a dataset with larger fraction of incorrect labels.

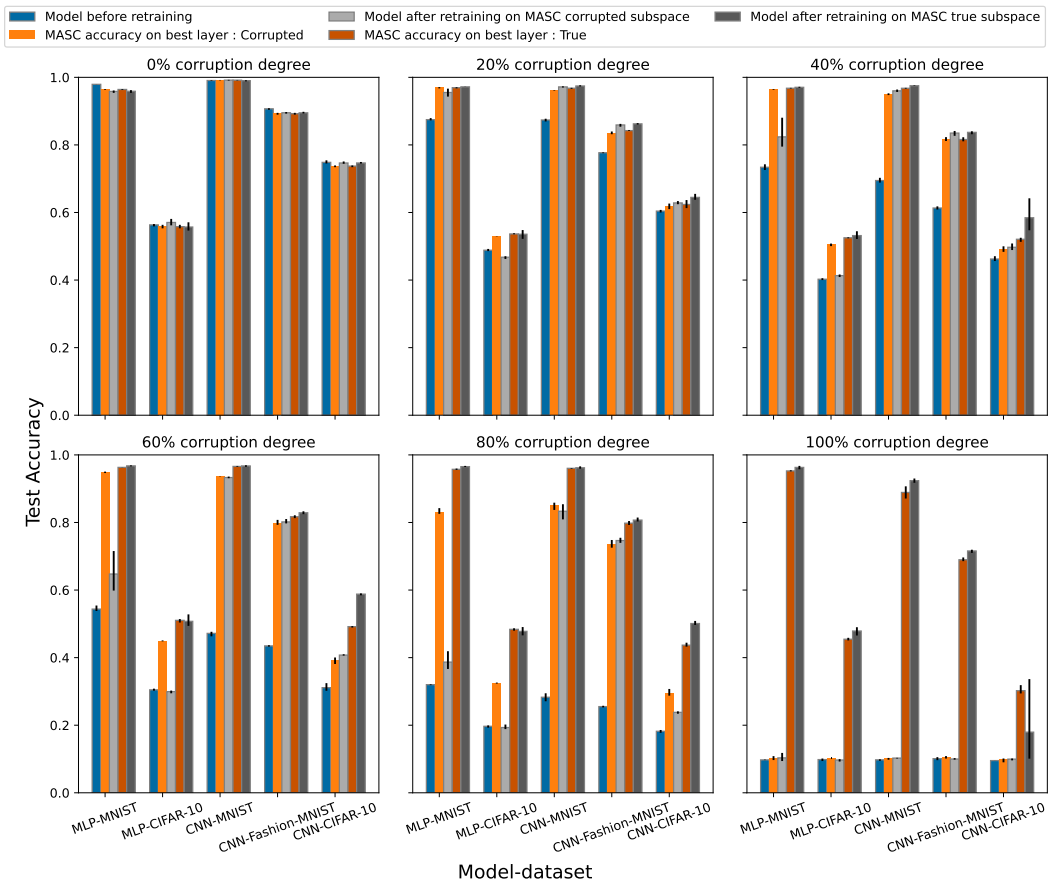

Figure 4: Test accuracies averaged over three runs on the 80% test dataset is plotted for different model-dataset pairs for various corruption degrees and for various models/MASC classifiers. Model before retraining corresponds to the existing memorized model. Model after retraining on MASC corrupted subspace corresponds to model trained with training dataset relabeled using MASC corrupted subspace predictions on the best layer. Model after retraining on MASC true subspace corresponds to model trained with training dataset relabeled using MASC subspaces corresponding to true label predictions on the best layer. MASC test accuracy on the best layers for corrupted and true label subspaces on existing corrupted models (before retraining) are shown for comparison. The best layer was identified using the 20% test dataset. Error bar represents the range on three different runs.

Thirdly, in some cases (e.g., AlexNet-Tiny ImageNet with 40%, 60%, 80% corruption degree), we observe that even though relabeling results in a somewhat larger fraction of incorrect labels, the test accuracy of the retrained model is slightly better. This suggests that retraining performance is not simply driven by fraction of correct labels, but that specifics of which points are relabeled can drive retraining performance in ways that remain to be investigated.

With respect to models retrained on MASC true label subspaces, it was observed that the test accuracy of such models usually performed significantly better than the models before retraining.

By-and-large, we find that the retrained models that use MASC corrupted subspace relabeling don't have better test performance than the corresponding MASC classifiers. However, it would be interesting to apply MASC on the retrained models to see if that would further improve generalization performance.

7 Discussion

In this work, we investigated the phenomenon of memorized networks not generalizing well, asking why the ability to generalize is apparently diminished due to the act of memorizing. We find, surprisingly, that the intrinsic ability to generalize remains present to a degree not previously recognized, and this ability can be decoded from the internals of the network by straightforward means. On the one hand, we design probes that use the subspace geometry of the corrupted training data to decode such better generalization. We also demonstrate using true labels post hoc that there exist subspaces that allow for an even more improved decoding. Furthermore, we show that such decoding can be leveraged to retrain the models to have better generalization. We also show (in Supplementary Material) that the internal representations of some Deep Networks trained on true labels, possess the ability to substantially memorize relabelings of its training data.

An interesting question is about why this phenomenon even occurs; naïvely one would expect that Deep Networks, on being trained with noisy data, discard the ability to generalize in favor of learning noise. Are there specific inductive biases that promote such generalization? And, do such mechanisms also promote generalization in networks whose training data isn't corrupted significantly by such noise? It would also be instructive to study the dynamics of this form of generalization during training. It is known (Arpit et al., 2017) that the model's test accuracy transiently peaks in the early epochs of training with corrupted data, before dropping while training accuracy of the corrupted training data rises. It is unclear whether this transient rise in model generalization is caused by the subspace organization seen here, & if so, why such subspace organization isn't degraded as much as the model's test accuracy over further epochs of training. Additionally, certain Deep Networks trained on specific uncorrupted datasets seem to possess internal representations that are amenable to significant memorization, whereas others aren't. The mechanistic basis of this ability is unclear & its possible connections to generalization in the such models merit further investigation.

The work has a number of implications. On the one-hand, it suggests that the ability to memorize and generalize may not be antithetical. Indeed, in multiple cases, we are able to construct single MASC classifiers that perform well both on the shuffled training labels as well as on the held-out test data that has true labels. Secondly, theories proposed to explain generalization in Deep Networks have traditionally argued for the setting where the data distribution is well-behaved, i.e. corresponding to real-world data, but not for data with shuffled labels. We suggest, in light of the present results, that such theories also ought to be able to explain why networks retain the ability to generalize even in the face of noisy training data. That is, a satisfactory understanding of generalization in Deep Networks should also cover settings where the training data is noisy and its distribution is not well-behaved. Thirdly and more pragmatically, techniques such as the MASC classifier might suggest a way of boosting generalization in trained Deep Networks, whose training data intrinsically contains varying degrees of label noise. While this has been beyond the scope of the present paper, possibilities of designing new techniques for learning subspaces that have good generalization ability could be explored. Indeed, it is possible that significantly better subspaces exist than the ones uncovered here, & it would be interesting to see how much the generalization accuracy can be improved by pursuing this direction. Relatedly, it is possible that other classifiers operating on layerwise outputs have better performance than MASC – a possibility that merits further exploration. Fourthly, it would be interesting to formulate a measure to study representational similarity between memorized & generalized networks to see if they use similar mechanisms. Does the answer depend on the particular class of networks (e.g. MLPs vs. CNNs)?

Finally, the results here are reminiscent of a puzzling phenomenon observed in Neuroscience. In multiple settings (Miura et al., 2012; Stringer et al., 2021), in the rat olfactory system and the mouse visual system, it has been shown that a decoder using data from a subset of neurons from specific areas in the brain of a well-trained behaving animal has accuracy significantly better than the behavioral accuracy of the animal on novel trials, even though the animal is motivated to do well on the task. This implies that those animals have better innate generalization ability on that task – which can be easily decoded from a subset of their neurons – than is manifested by their behavior. It may therefore be that this is a phenomenon shared between brains and machines, whose underlying mechanisms and potential trade-offs remain to be investigated.

References

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. 2018. *arXiv preprint arXiv:1610.01644*, 2018.

Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *International conference on machine learning*, pp. 233–242. PMLR, 2017.

Amr Bakry, Mohamed Elhoseiny, Tarek El-Gaaly, and Ahmed Elgammal. Digging deep into the layers of cnns: In search of how cnns achieve view invariance. *arXiv preprint arXiv:1508.01983*, 2015.

N Alex Cayco-Gajic and R Angus Silver. Re-evaluating circuit mechanisms underlying pattern separation. *Neuron*, 101(4):584–602, 2019.

SueYeon Chung, Daniel D Lee, and Haim Sompolinsky. Linear readout of object manifolds. *Physical Review E*, 93(6):060301, 2016.

Uri Cohen, SueYeon Chung, Daniel D Lee, and Haim Sompolinsky. Separability and geometry of object manifolds in deep neural networks. *Nature communications*, 11(1):746, 2020.

Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

Freeman Dyson et al. A meeting with enrico fermi. *Nature*, 427(6972):297–297, 2004.

Matthew S Farrell, Stefano Recanatesi, Guillaume Lajoie, and Eric Shea-Brown. Dynamic compression and expansion in a classifying recurrent network. *bioRxiv*, pp. 564476, 2019.

Benoît Frénay, Ata Kabán, et al. A comprehensive introduction to label noise. In *ESANN*. Citeseer, 2014.

Peiran Gao and Surya Ganguli. On simplicity and complexity in the brave new world of large-scale neuroscience. *Current opinion in neurobiology*, 32:148–155, 2015.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in neural information processing systems*, 31, 2018.

Olivier J Hénaff, Robbe LT Goris, and Eero P Simoncelli. Perceptual straightening of natural videos. *Nature neuroscience*, 22(6):984–991, 2019.

Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International conference on machine learning*, pp. 2304–2313. PMLR, 2018.

Ashish Khetan, Zachary C Lipton, and Anima Anandkumar. Learning from noisy singly-labeled data. *arXiv preprint arXiv:1712.04577*, 2017.

Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pp. 3519–3529. PMLR, 2019.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Andrew K Lampinen and Surya Ganguli. An analytic theory of generalization dynamics and transfer learning in deep linear networks. *arXiv preprint arXiv:1809.10374*, 2018.

Mingchen Li, Mahdi Soltanolkotabi, and Samet Oymak. Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. In *International conference on artificial intelligence and statistics*, pp. 4313–4324. PMLR, 2020.

Ashok Litwin-Kumar, Kameron Decker Harris, Richard Axel, Haim Sompolinsky, and LF Abbott. Optimal degrees of synaptic connectivity. *Neuron*, 93(5):1153–1164, 2017.

Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. *Advances in neural information processing systems*, 33: 20331–20342, 2020.

Eran Malach and Shai Shalev-Shwartz. Decoupling" when to update" from" how to update". *Advances in neural information processing systems*, 30, 2017.

Aditya Krishna Menon, Brendan Van Rooyen, and Nagarajan Natarajan. Learning from binary labels with instance-dependent noise. *Machine Learning*, 107:1561–1595, 2018.

Keiji Miura, Zachary F Mainen, and Naoshige Uchida. Odor representations in olfactory cortex: distributed rate coding and decorrelated population activity. *Neuron*, 74(6):1087–1098, 2012.

Grégoire Montavon, Mikio L Braun, and Klaus-Robert Müller. Kernel analysis of deep networks. *Journal of Machine Learning Research*, 12(9), 2011.

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.

Mohammed Ali Moustafa. Tiny imagenet, 2017. URL `https://kaggle.com/competitions/tiny-imagenet`.

Ning Qian. On the momentum term in gradient descent learning algorithms. *Neural networks*, 12(1):145–151, 1999.

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.

Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*, 2014.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *International conference on machine learning*, pp. 4334–4343. PMLR, 2018.

Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

Clayton Scott, Gilles Blanchard, and Gregory Handy. Classification with asymmetric label noise: Consistency and maximal denoising. In *Conference on learning theory*, pp. 489–511. PMLR, 2013.

Yanyao Shen and Sujay Sanghavi. Learning with bad training data via iterative trimmed loss minimization. In *International conference on machine learning*, pp. 5739–5748. PMLR, 2019.

Cory Stephenson and Tyler Lee. When and how epochwise double descent happens. *arXiv preprint arXiv:2108.12006*, 2021.

Cory Stephenson, Abhinav Ganesh, Yue Hui, Hanlin Tang, SueYeon Chung, et al. On the geometry of generalization and memorization in deep neural networks. In *International Conference on Learning Representations*, 2021.

Carsen Stringer, Marius Pachitariu, Nicholas Steinmetz, Matteo Carandini, and Kenneth D Harris. High-dimensional geometry of population responses in visual cortex. *Nature*, 571(7765):361–365, 2019.

Carsen Stringer, Michalis Michaelos, Dmitri Tsyboulski, Sarah E Lindo, and Marius Pachitariu. High-precision coding in visual cortex. *Cell*, 184(10):2767–2778, 2021.

David Sussillo and Larry F Abbott. Generating coherent patterns of activity from chaotic neural networks. *Neuron*, 63(4):544–557, 2009.

Lina M Tran, Adam Santoro, Lulu Liu, Sheena A Josselyn, Blake A Richards, and Paul W Frankland. Adult neurogenesis acts as a neural regularizer. *Proceedings of the National Academy of Sciences*, 119(45): e2206704119, 2022.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *Advances in neural information processing systems*, 27, 2014.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization, 2017. URL `https://arxiv.org/abs/1611.03530`.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31, 2018.