Using Sliced Mutual Information to Study Memorization and Generalization in Deep Neural Networks

Shelvia Wongso
National University of Singapore

Rohan Ghosh
National University of Singapore

Mehul Motani
National University of Singapore

Abstract

In this paper, we study the memorization and generalization behaviour of deep neural networks (DNNs) using sliced mutual information (SMI), which is the average of the mutual information (MI) between one-dimensional random projections. We argue that the SMI between features in a DNN (T) and ground truth labels (Y), $SI(T;Y)$, can be seen as a form of *usable information* that the features contain about the labels. We show theoretically that $SI(T;Y)$ can encode geometric properties of the feature distribution, such as its spherical soft-margin and intrinsic dimensionality, in a way that MI cannot. Additionally, we present empirical evidence showing how $SI(T;Y)$ can capture memorization and generalization in DNNs. In particular, we find that, in the presence of label noise, all layers start to memorize but the earlier layers stabilize more quickly than the deeper layers. Finally, we point out that, in the context of Bayesian Neural Networks, the SMI between the penultimate layer and the output represents the worst case uncertainty of the network's output.

1 INTRODUCTION

Over the last decade, there has been a proliferation of deep neural network (DNN) architectures, some of which are known to be capable of generalizing across various domains and tasks (Tay et al., 2020). Despite the significant progress made in the field, a comprehensive theoretical understanding of DNNs remains elusive. The primary focus of our paper is to investigate two key behaviours of DNNs, namely, memorization and generalization, by employing a scalable information measure called the Sliced Mutual Information (SMI), originally introduced in Goldfeld and Greenewald (2021). SMI is defined as the average of the mutual information (MI) measures between one-dimensional random projections of the random variables (RVs). An interesting property of SMI is that it can increase with deterministic transformations of the original RVs (Goldfeld and Greenewald, 2021), which does not hold for MI due to the data processing inequality (DPI). This property aligns well with the characteristic of DNNs which learn a hierarchy of processed feature representations of the input that are increasingly more useful for predicting the labels.

A new notion of informativeness. As observed in Xu et al. (2020), the MI between two RVs X and Y assumes that the means to predict Y from X (or X from Y) are *computationally unbounded*. Xu et al. (2020) proposed a new measure called *predictive $\mathcal{V}$-information*, which considers a predictive family and looks at the change in predictability of Y when given a side information X. This measure limits the set of possible predictive models under computational or statistical constraints. Similar to SMI, *predictive $\mathcal{V}$-information* can also increase with more computation of the RVs. This suggests that there is a growing field of research that considers new notions of informativeness which can increase with processing for compatibility with DNNs.

On memorization. In this paper, we adopt the same operational definition of memorization as (Arpit et al., 2017) which is *the behaviour of DNNs trained on noisy labels*. DNNs are capable of fitting random labels in the training data, resulting in poor test data generalization (Zhang et al., 2017). Standard explicit regularization measures such as dropout and weight decay cannot prevent memorization (Song et al., 2020). In addition, (Arpit et al., 2017) shows that DNNs tend to prioritize learning simple generalizable features first before memorizing. Furthermore, (Stephenson et al., 2021) shows that memorization mainly occurs in deeper layers while earlier layers are not affected as much. In Ghosh and Motani (2021), overall memorization in DNNs was captured via Kolmogorov Growth, a function complexity measure. Here, we show that SMI is able to capture memorization in different DNN layers trained with different degrees of label noise.

Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS) 2023, Valencia, Spain. PMLR: Volume 206. Copyright 2023 by the author(s).

On generalization. Despite ongoing research aimed at understanding why overparameterized DNNs can generalize well (Kawaguchi et al., 2017), no definitive conclusion has been reached (Zhang et al., 2021). While numerous practical complexity measures have been developed to predict generalization, they still fall short in accurately evaluating a wide range of networks and datasets, and often lack theoretical foundations that justify their suitability in predicting generalization (Jiang et al., 2020). In this paper, we demonstrate empirically that the SMI between the features and the labels correlates well with the generalization gap, particularly in the deeper layers. Furthermore, we present a theoretical rationale for why SMI may serve as a predictor of the generalization gap by examining its relationship to margin and intrinsic dimensionality.

Related work. Information-theoretic techniques have been proposed for investigating memorization (Achille and Soatto, 2019) and generalization (Shwartz-Ziv and Tishby, 2017) in DNNs, often employing MI as a key tool. However, current MI estimation methods (Kraskov et al., 2004; Belghazi et al., 2018) are still unable to reliably and efficiently estimate MI for high-dimensional variables (Song and Ermon, 2020; Poole et al., 2019; McAllester and Stratos, 2020) due to the curse of dimensionality. For example, information bottleneck has been suggested as a way to understand how DNNs operate (Shwartz-Ziv and Tishby, 2017; Saxe et al., 2018), yet empirical investigation of this theory has yielded inconsistent results (Geiger, 2020). Similar argument holds for approaches that bound generalization error using MI and its variants (Xu and Raginsky, 2017). In a similar line of work, Wongso et al. (2022) shows that SMI between the features of the hidden layer and the labels encodes information about the network's ability to predict labels correctly. Our paper extends the work to various benchmark architectures and datasets as well as the relation to memorization and generalization.

Contributions. Our contributions are as follows.

1. We propose the use of SMI to study the memorization and generalization behavior in DNNs.
2. We provide theoretical lower bounds on SMI based on margin and intrinsic dimensionality, using them to justify SMI as a metric to study memorization and generalization in DNNs.
3. We empirically show that, in the presence of label noise, the SMI between features and noisy labels of all layers decreases, with earlier layers stabilizing more quickly than the deeper layers.
4. We also observe that SMI between the penultimate features and labels correlates with the generalization gap, i.e., difference between test and training accuracies.
5. We argue for the significance of SMI as an averaging-based information measures in the context of Bayesian Neural Networks by showing that it represents the worst-case uncertainty of the weights.

2 SMI IN DEEP NEURAL NETWORKS

Overview. In this section, we first provide a formal definition and properties of SMI, followed by the motivation of using this metric in studying DNNs as well as its estimation method. Then, we show empirically that SMI can increase with more processing of the input features in various benchmark DNN models and datasets.

Preliminaries. The *Shannon mutual information* between two random variables X and Y is defined as $I(X;Y) = \mathbb{E}_{P_{XY}}[\log \frac{P_{XY}}{P_X P_Y}]$ (Cover and Thomas, 2001). In this paper, all information-theoretic quantities are measured in bits. The d-dimensional unit sphere is $\mathbb{S}^{d-1}$ and its surface area is $S_{d-1} = 2\pi^{d/2}/\Gamma(d/2)$. In DNNs with l layers, we denote X as the input features, Y as the ground truth labels, $\hat{Y}$ as the predicted labels and T_j as the output of each layer where $j = 0, \ldots, l$. Specifically, T_0 is the input layer, $T_{1,\ldots,l-1}$ are the hidden layers, and T_l is the output layer.

2.1 Definition and Properties of SMI

Sliced mutual information (SMI) is proposed by Goldfeld and Greenewald (2021) as an alternative measure of informativeness that is scalable to high dimensions. The SMI between two random variables, X and Y, is formally defined as (Goldfeld and Greenewald, 2021, Definition 1):

Definition 1. *Fix* $(X,Y) \sim P_{X,Y} \in \mathcal{P}(\mathbb{R}^{d_x} \times \mathbb{R}^{d_y})$. *Let* $\Theta \sim \mathrm{Unif}(\mathbb{S}^{d_x-1})$ *and* $\Phi \sim \mathrm{Unif}(\mathbb{S}^{d_y-1})$ *are independent of each other and of* (X,Y). *The SMI is given by:*

$$\begin{aligned} SI(X;Y) &:= I(\Theta^T X; \Phi^T Y | \Theta, \Phi) \\ &= \frac{1}{S_{d_x-1} S_{d_y-1}} \oint_{\mathbb{S}^{d_x-1}} \oint_{\mathbb{S}^{d_y-1}} I(\theta^T X; \phi^T Y) d\theta d\phi \end{aligned} \quad (1)$$

Remark 1. *Similarly, we can define sliced entropy of* X *as* $sh(X) = h(\Theta^T X|\Theta)$ *where* $h(.)$ *represents differential entropy. The conditional sliced entropy of* X *given* Y *is given by* $sh(X|Y) = h(\Theta^T X|\Theta, \Phi, \Phi^T Y)$.

While it is not a proxy for MI itself, SMI shares several basic properties of MI (Goldfeld and Greenewald, 2021, Proposition 1), thus preserving some of the desirable information-theoretic properties. We list several of the useful properties:

1. ***Non-negativity and independence:*** $SI(X;Y) \geq 0$ with equality iff X and Y are independent.
2. ***Bound:*** $SI(X;Y) \leq I(X;Y)$
3. ***Entropy decomposition:*** $SI(X;Y) = sh(X) + sh(Y) - sh(X,Y) = sh(X) - sh(X|Y) = sh(Y) - sh(Y|X)$.
4. ***Chain rule:*** $SI(X,Y;Z) = SI(X;Z) + SI(Y;Z|X)$

Similar to MI, SMI can also be written in terms of relative entropy (Goldfeld and Greenewald, 2021, Proposition 1, Property 3) and has corresponding variational forms (Goldfeld and Greenewald, 2021, Proposition 3). Moreover, SMI can be further extended to k-SMI which considers projections to k-dimensional subspaces Goldfeld et al. (2022). However, it is to be noted that unlike MI, SMI can grow with deterministic processing of the original RVs. The implications of this in the context of DNNs will be discussed further in Section 2.2 and Section 2.4.

Remark 2. *It is worth mentioning that although MI might be infinite in certain scenarios (an example is provided by Goldfeld and Greenewald (2021)), SMI can still have a finite value. This is especially relevant when measuring the quantity between the input features X and the features of hidden layers T in DNNs. As T is often a continuous deterministic function of X, the MI $I(X;T)$ is infinite (Saxe et al., 2018), and any finite value of $I(X;T)$ depends strongly on the MI estimators (Geiger, 2020). Consequently, no definitive conclusion can be drawn from the inconsistent empirical analysis of $I(X;T)$. Instead, future studies may consider investigating $SI(X;T)$ which is more likely to be finite.*

2.2 Motivation for using SMI

Goldfeld and Greenewald (2021) and Goldfeld et al. (2022) demonstrate how SMI can be applied to modern machine learning tasks such as feature extraction and disentanglement of latent factors (using InfoGAN). In this work, we showcase how SMI can also be used to analyze the behaviours of DNNs. Below, we provide concrete arguments which motivate the study of SMI in this context.

1. **Scalability to high dimensions:** Features of hidden layers in DNNs can be very high-dimensional, which can lead to inaccurate estimation of conventional measures of information, such as MI. Let n be the number of given i.i.d samples of $P_{X,Y} \in \mathcal{P}(\mathbb{R}^d \times \mathbb{R})$ and m be the number of projections used in the estimation of SMI. For standard MI, error rates scale as $n^{-1/d}$ for large d, whereas for SMI, the error scales as $m^{-1/2} + n^{-1/2}$ (Goldfeld and Greenewald, 2021). Therefore, SMI is better suited to study high-dimensional variables in DNNs compared to MI.

2. **A form of usable information:** Classic MI $I(X;Y)$ assumes that one can employ arbitrarily complex models to predict Y from X. However, computational complexity is bounded in DNNs, and every layer essentially represents a linear map of the input, followed by some continuous nonlinear activation. As SMI processes the RVs through linear projections, it also represents a form of *usable information* that is directly relevant in DNNs. Furthermore, unlike $\mathcal{V}$-information (Xu et al., 2020) which estimates the *largest* usable information between X and Y, SMI captures the notion of *average usable information* as it averages the MI for all random projections. We hypothesize that this property enables SMI to encode properties such as margin and intrinsic dimensionality of feature representations.

3. **Data Processing Inequality and DNNs:** Consider a Markov chain, $X \rightarrow T \rightarrow Y$, the data processing inequality (DPI) states that $I(X;Y) \geq I(T;Y)$. However, the incompatibility between DPI and DNNs has been highlighted previously in Xu et al. (2020) and Goldfeld and Greenewald (2021). In DNNs, we process the input features through the layers to extract feature representations that are increasingly useful for predicting the label. Thus, ideally, a desirable measure of information in the context of DNNs should be indicative of how *useful* or *usable* a feature representation for inferring the label, and therefore can grow from processing. SMI, similar to $\mathcal{V}$-information, possesses this property (Goldfeld and Greenewald, 2021; Wongso et al., 2022), which further supports its suitability for studying DNNs.

2.3 SMI Estimation

Throughout our experiments, we focus on investigating the SMI between the features X and the labels Y in the training dataset. Since the labels Y is discrete, we only project the features X. SMI can be estimated from high-dimensional samples by combining a scalar MI estimator and a Monte Carlo (MC) integrator. The pseudocode for the SMI estimator is given in Appendix B.1. The estimated SMI can be written as

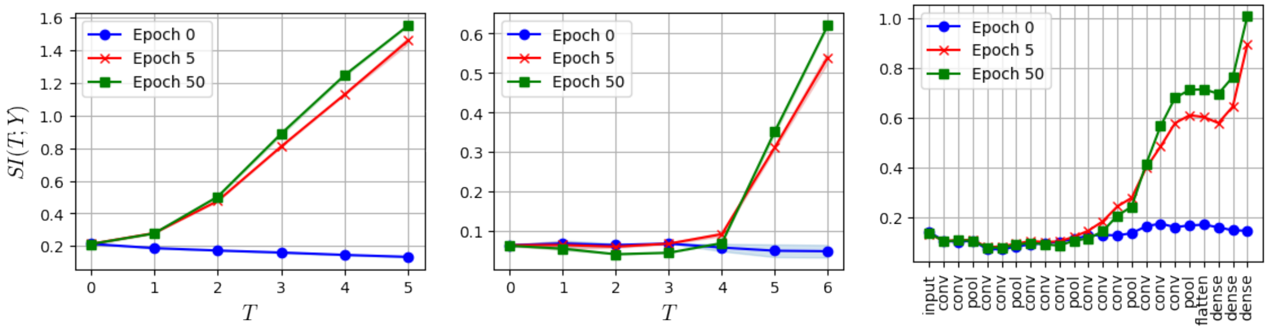
$$\widehat{SI}^{m,n} := \frac{1}{m} \sum_{j=1}^{m} \hat{I} \left(\left(\Theta_j^T X\right)^n ; (Y)^n \right), \qquad (2)$$

where m represents the number of projections and n represents the number of samples.

We adopt the KSG estimator (Kraskov et al., 2004) implemented using Non-parametric Entropy Estimation Toolbox (NPEET) (Krizhevsky, 2000) for the MI computation in (2). As noted in Goldfeld et al. (2022), the estimation error of SMI from the MC sampling is bounded by the variance of the MI projections which scales as $O\left(\sqrt{(1/d_x + 1/d_y)/m}\right)$. We study how the SMI estimates change with m and n, the results for which are shown in Appendix B.2. We find that $m = 1000$ and $n = 10000$ generally yield stable SMI estimates in our experiments.

2.4 SMI Behaviour in DNNs

We study how the SMI between features T of any layer within a neural network and the labels Y, changes during training. Wongso et al. (2022) provides evidence that $SI(T;Y)$ can encode the information shared between features and labels for simple networks and datasets. In this paper, we extend it to more complex architectures and datasets. Similar to Wongso et al. (2022), we observe that a trained DNN shows an overall increasing trend of $SI(T;Y)$ with depth, indicating that the deeper layers learn to become better feature extractors during training.



(a) MLP, MNIST

(b) CNN, CIFAR10

(c) VGG16, CIFAR100

Figure 1: The $SI(T;Y)$ of different layers at different epochs for: (a) MLP trained with MNIST, (b) CNN trained with CIFAR10, and (c) VGG16 trained with CIFAR100. $SI(T;Y)$ generally increases with depth T during training.

Experiment: We train a 5-layer multi-layer perceptron (MLP) on MNIST (LeCun et al., 2010), a 6-layer convolutional neural network (CNN) on CIFAR10 Krizhevsky (2009), and VGG16 Simonyan and Zisserman (2015) on CIFAR100 Krizhevsky (2009) for 50 epochs. The SMI of the different layers is computed and the results are shown in Fig. 1. For convolutional layers, T represents the output of the flattened feature maps. We provide more details of the experiments in Appendix C.1.

Before training: At epoch 0, $SI(T;Y)$ may increase or decrease slightly with depth. This is because the network is initialized with random weights and none of the layers have learnt any useful information for inferring the labels. Therefore, there is very little discrepancy in the amount of usable information present in the various layers, and $SI(T;Y)$ remains close to zero for all T.

During and after training: At epoch 5 and epoch 50, $SI(T;Y)$ generally increases with depth. From this observation, we can imply that the deeper layers of a trained network are better feature extractors compared to earlier layers. Our results support the findings presented in Zeiler and Fergus (2014) which indicate that the deeper layers produce more discriminative features that result in better prediction of the labels. In all the three scenarios, the increase in $SI(T;Y)$ from epoch 0 to epoch 5 is significantly greater compared to that from epoch 5 to epoch 50, particularly for the deeper layers. This might suggest that the network rapidly learns most of the useful features during the initial phase of training, while the increase in useful information learnt becomes progressively smaller as training continues, eventually risking overfitting.

We discuss the major differences in the behaviour of $SI(T;Y)$ in MLP and CNN:

1. **MLP architecture:** In Fig. 1a, the $SI(T;Y)$ values for all layers are higher at epoch 5 and epoch 50 than at epoch 0, indicating that the layers become more effective at predicting labels as they are trained. Moreover, there is a clear upward trend in $SI(T;Y)$ as we move deeper into the layers, suggesting that the amount of usable information progressively grows with depth.

2. **CNN-like architecture:** In both Fig. 1b and Fig. 1c, the $SI(T;Y)$ of earlier layers changes minimally in contrast to the deeper layers. This observation points to the difference in the behaviour of training MLPs and CNNs. Unlike in MLPs, where the amount of usable information tends to steadily increase with depth, we observe that the usable information progressively grows only in the deeper layers in CNNs. This observation suggests that the majority of usable information may be concentrated in the deeper layers of CNNs.

Another interesting observation is that in the 6-layer CNN case (Fig. 1b), the $SI(T;Y)$ of earlier layers drop slightly while the $SI(T;Y)$ of deeper layers is maintained at high values during training. This might suggest that for this particular architecture, the earlier layers may slightly lose their effectiveness as feature extractors, as the deeper layers continue to improve in their ability to extract useful features.

3 CONNECTION TO MARGIN AND ID

Overview. In this section, we show how SMI, unlike MI, can be related to various geometric properties of the feature distribution in DNNs, namely the *spherical-soft* margin and the intrinsic dimensionality (ID). Note that in this section, we mainly interpret X as the feature output of any layer in a DNN. We prove that $SI(X;Y)$ can be lower bounded by a function that depends on the margin and ID. From this bound, we find that $SI(X;Y)$ is likely increases with increasing margin and decreasing ID, and vice-versa. We then provide a rationale for why SMI is a well-suited metric for investigating memorization and generalization in DNNs, through its connection to margin and ID.

3.1 SMI and Margin

We start by asking: what geometrical properties can SMI encode about a distribution $P(X)$ in the context of binary classification, where X has an underlying ground truth label $Y \in \{0, 1\}$? In the context of DNNs, X can be considered to be the output of a specific hidden layer within a DNN, and Y can be considered as the output label. To answer the question, we demonstrate how SMI can be related to the soft margin of the distribution. To that end, we first define the notion of spherical gap and subsequently the spherical soft-margin separation criterion.

Definition 2. ***Spherical gap:*** *Given two non-overlapping spheres of radii R_1 and R_2 in $\mathbb{R}^k$ for some k, with their centre points C_1 and C_2, the spherical gap m_g is defined such that $m_g = d(C_1, C_2) - (R_1 + R_2)$, where $d(C_1, C_2)$ is the Euclidean distance between C_1 and C_2.*

Definition 3. ***Spherical soft-margin separation (SSM-separation):*** *Let $X \in \mathbb{R}^{d_x}$ be the features and $Y \in \{0, 1\}$ be the labels in a binary classification task. Given this, X and Y are said to be $(R_1, R_2, m_g, \epsilon)$-SSM-separated, if there exists two spheres S_1 and S_2 of radii R_1 and R_2, respectively, in $\mathbb{R}^d$, having a spherical gap of $m_g \geq 0$, and it holds that*

$$P(X \in S_1|Y = 0) = P(X \in S_2|Y = 1) = 1 - \epsilon.$$

The quantity ϵ controls the amount of overlap between the features of the two classes in each sphere, i.e., between $P(X \in S_1|Y = 0)$ and $P(X \in S_1|Y = 1)$, and between $P(X \in S_2|Y = 0)$ and $P(X \in S_2|Y = 1)$. Note that in what follows, if RVs X and Y are $(R_1, R_2, m_g, \epsilon)$-*SSM-separated*, we refer to m_g as the spherical soft-margin.

Remark 3. ***(On the SSM-Separation)*** *Note that given the distributions $P(X|Y = 0)$ and $P(X|Y = 1)$, we can find more than one set of $(R_1, R_2, m_g, \epsilon)$ such that X and Y are $(R_1, R_2, m_g, \epsilon)$-SSM-separated. That is because as we increase the radii of the spheres S_1 and S_2, the overlap between $P(X \in S_1|Y = 0)$ and $P(X \in S_1|Y = 1)$, and between $P(X \in S_2|Y = 0)$ and $P(X \in S_2|Y = 1)$, can likely increase, which would yield a larger ϵ. Also, note that when $R_1, R_2 \to \infty$, then the spherical soft-margin m_g becomes the conventional soft-margin used in the context of Support Vector Machines (Hearst et al., 1998).*

With this, we now relate SMI to the spherical soft-margin in the context of binary classification, in the following theorem (refer to Appendix A.1 for the proof).

Theorem 1. ***(Margin-based lower bound)*** *We are given RVs $X \in \mathbb{R}^{d_x}$ and $Y \in \{0, 1\}$. Assume $P(Y = 0) = P(Y = 1) = 0.5$ and that X, Y are $(R_1, R_2, m_g, \epsilon)$-SSM-separated. Given this, we then have*

$$(1 - H(\epsilon, 1 - \epsilon)) B_{\gamma(m_g, R_1, R_2)}\left(\frac{d_x - 1}{2}, \frac{1}{2}\right) \leq SI(X; Y) \leq 1, \quad (3)$$

where $\gamma(m_g, R_1, R_2) = \frac{m_g}{m_g+R_1+R_2}\left(2 - \frac{m_g}{m_g+R_1+R_2}\right)$, $B_x(a, b)$ is the regularized incomplete beta function (Oldham et al., 2008), and $H(p_1, p_2) = -p_1 \log p_1 - p_2 \log p_2$ is the entropy function.

Remark 4. ***(SMI and Margin)*** *The main implication of Theorem 1 is that SMI can be related to the spherical soft-margin between the distributions $P(X|Y = 0)$ and $P(X|Y = 1)$ which in some sense, quantify a degree of overlap between the two classes in the feature space. More precisely, we show in Appendix A.2 that the regularized incomplete beta function in the lower bound increases with increasing margin m_g, when everything else is fixed. Thus, Theorem 1 implies that a larger value of the soft margin will likely yield a larger value of SMI $SI(X; Y)$. Note that when $\epsilon = 0$, this bound reduces to the result in Wongso et al. (2022) where $R_1 = R_2$ was imposed. Furthermore, our result can also be applied in the scenario where there are more than two labels, by considering a one-vs-all binarization of the problem and class-wise margins instead.*

Remark 5. ***(MI and Margin)*** *We also note that unlike SMI, MI is not sensitive to margin. Consider the case where the distributions $P(X|Y = 0)$ and $P(X|Y = 1)$ have no overlap ($\epsilon = 0$). In that case, we would then have that MI $I(X; Y) = H(Y) = 1$ where $H(Y)$ represent the discrete entropy operator for Y. This is due to our assumption that $P(Y = 0) = P(Y = 1) = 0.5$ and X will provide complete knowledge of Y. This also implies that when the distributions are $(R_1, R_2, m_g, 0)$ SSM-separated, MI does not change with the margin m_g, as $I(X; Y) = 1$ in this case. We perform proof-of-concept experiments to illustrate this point in Appendix A.3. Therefore, SMI may prove to be a more desirable metric when the margin of the feature distributions can change.*

SMI, Margin and Memorization: As mentioned previously, memorization in the context of DNNs is most often studied in the context of label noise. With larger label noise, one expects the decision boundary to get more complex (Garcia et al., 2015), and yield smaller margins in the feature space (Lin and Bradic, 2021). This property has also been used to actively filter out the data with noisy labels, by identifying the data samples for which the margin is smaller (Lin and Bradic, 2021). As Theorem 1 shows that SMI and margin are related, this points to a potential connection between SMI and memorization as well. In the context of Theorem 1, as the margin between $P(X|Y = 0)$ and $P(X|Y = 1)$ reduces with larger label noise, this should also lead to a reduction of SMI between the features and the noisy labels. To verify whether SMI can indeed capture memorization, we study the impact of label-noise on SMI in our experiments in Section 4.1.

SMI, Margin and Generalization: There exists many works in literature (Koltchinskii and Panchenko, 2002; Bartlett and Shawe-Taylor, 1999; Montanari et al., 2019) which find that the soft-margin of a classifier can relate to its ability to generalize. Margin-based generalization bounds have been proposed over the last two decades, and many of them find that a larger margin can be associated with a potentially smaller generalization gap and vice-versa Bartlett et al. (2017); Neyshabur et al. (2018); Jiang et al. (2019). As Theorem 1 connects SMI to a variant of soft-margin and finds that SMI will likely increase as the soft-margin grows, we hypothesize that SMI should therefore also increase when generalization gap decreases. We perform experiments to test our hypothesis in Section 4.2.

3.2 SMI and Intrinsic Dimensionality

The intrinsic dimensionality (ID) of X is defined as the minimum number of variables required to represent X. We find that SMI can encode the ID of the distribution of X, when we consider ID as the dimensionality of the smallest linear subspace W that contains X. We provide theoretical results below (refer to Appendix A.4 and Appendix A.5 for the proofs). For what follows, recall Definition 2 and Definition 3.

Theorem 2. *Assume that the support of $P(X)$ lies within a linear subspace W of K dimensions. Let W be represented by the orthonormal basis set $\{u_i\}_{i=1}^K$, and the center of W be at a distance of $\boldsymbol{\mu}$ from the origin. Let $\boldsymbol{U}$ be a matrix with columns $\{u_i\}_{i=1}^K$. Then, we have*

$$SI(X;Y) = SI\left(\boldsymbol{U}^T\left(X-\boldsymbol{\mu}\right);Y\right). \tag{4}$$

Corollary 2.1. ***(Margin- and ID-based lower bound)*** *Assume we are given RVs $X \in \mathbb{R}^{d_x}$ and $Y \in \{0,1\}$, such that $P(Y=0) = P(Y=1) = 0.5$. Assume that K is the dimensionality of the smallest subspace W that contains the support of $P(X)$ ($K \leq d_x$). Furthermore, consider X and Y that are $(R_1, R_2, m_g, \epsilon)$-SSM-separated, via spheres of radius R_1 and R_2, whose centers lie in W. We then have,*

$$\left(1-H\left(\epsilon,1-\epsilon\right)\right)B_{\gamma(m_g,R_1,R_2)}\left(\frac{K-1}{2},\frac{1}{2}\right) \leq SI(X;Y) \leq 1, \tag{5}$$

where $\gamma(m_g, R_1, R_2)$ is as defined in Theorem 1.

Remark 6. ***(SMI and ID)*** *Corollary 2.1 is a consequence of Theorem 2, and it states that the SMI between X and Y is lower-bounded by the regularized incomplete beta function that depends not only on the margin, but also the ID (represented by K). More precisely, this result implies that SMI can potentially decrease in response to increasing K (refer to the Appendix A.2 for the dependence between the regularized incomplete beta function and its parameters).*

Remark 7. ***(MI and ID)*** *We also note that unlike SMI and similar to the margin case, MI is not sensitive to changes in the ID. Consider the case where the distributions $P(X|Y=0)$ and $P(X|Y=1)$ have no overlap ($\epsilon = 0$). In that case, we would then have that MI $I(X;Y) = H(Y) = 1$. We perform a proof-of-concept experiment in Appendix A.6 to show that MI does not change with ID while SMI decreases with increasing ID as expected. Therefore, SMI may prove to be a more desirable metric when the ID of the feature distributions is not fixed.*

SMI, ID and Memorization: It has been shown in the literature that in the presence of label noise, DNNs undergo dimensionality compression (measured using local ID) in the early stage of training, followed by dimensionality expansion as they overfit to the noisy labels (Ma et al., 2018). They show that by implementing dimensionality-driven learning strategy that avoids the dimensionality expansion stage, the networks can generalize better. As Corollary 2.1 shows that SMI and ID are related, this points to a potential connection between SMI and memorization as well. As the ID increases with larger label noise, this should also lead to a reduction of SMI between the features and the noisy labels, and vice-versa. This will be verified in the experiments in Section 4.1.

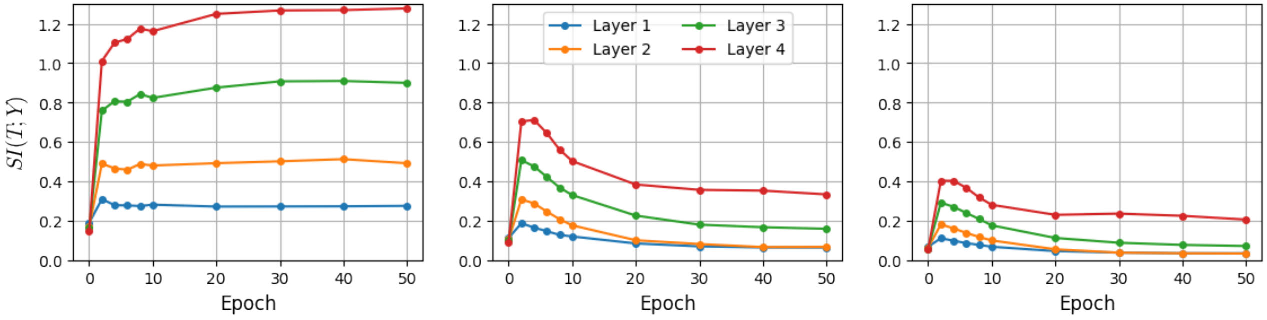
SMI, ID and Generalization: Recent works have shown that larger ID of the features can adversely impact generalization (Ansuini et al., 2019; Nakada and Imaizumi, 2020). As Corollary 2.1 finds that SMI will likely increase with decreasing ID, we hypothesize that SMI should therefore also increase when generalization gap decreases, and vice-versa. This is verified in the experiments in Section 4.2.

4 EXPERIMENTS

Overview. In the previous section, we discuss briefly how SMI can be indicative of memorization and generalization through the connection to margin and ID. In this section, we perform experiments to show how SMI relates to memorization (Section 4.1) and generalization (Section 4.2) in DNNs to confirm our hypothesis. In Section 4.1, we find that $SI(T;Y)$ decreases for all hidden layers when the networks are trained with higher label noise. In Section 4.2, we find that the SMI for penultimate layer, $SI(T_{l-1};Y)$, increases with decreasing generalization gap, vice-versa. Both sets of findings support our argument that SMI can indeed encode geometric properties of feature distribution such as the margin and the ID.

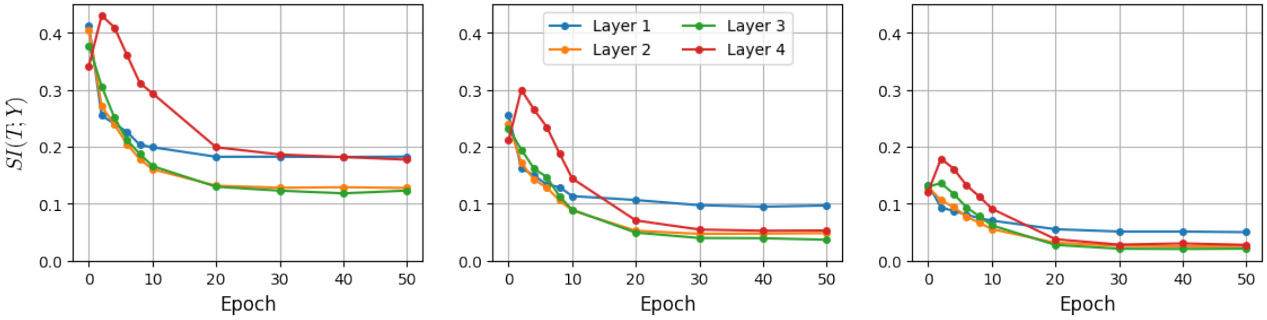
4.1 Memorization

Our goal is to investigate how SMI relates to memorization for different layers in DNNs by training them with different degrees of label noise. The label noise is induced by randomly changing the training labels for a fraction ϵ (referred to as the label noise ratio) of the dataset. Note that here we only apply Symmetric Label Noise (Van Rooyen et al. (2015)). We provide more details on the experiment in Appendix C.2. We estimate the $SI(T;Y)$ of each hidden layer during training, with Y representing the corrupted la-



(a) label noise ratio = 0 (b) label noise ratio = 0.2 (c) label noise ratio = 0.4

Figure 2: The $SI(T;Y)$ in 5-layer MLP trained with MNIST of different label noise ratio for 50 epochs. The $SI(T;Y)$ decreases with more label noise and it decreases faster for deeper layers.



(a) label noise ratio = 0 (b) label noise ratio = 0.2 (c) label noise ratio = 0.4

Figure 3: The $SI(T;Y)$ in 5-convolutional layer CNN trained with Fashion MNIST of different label noise ratio for 50 epochs. The $SI(T;Y)$ decreases with more label noise and it decreases faster for deeper layers.

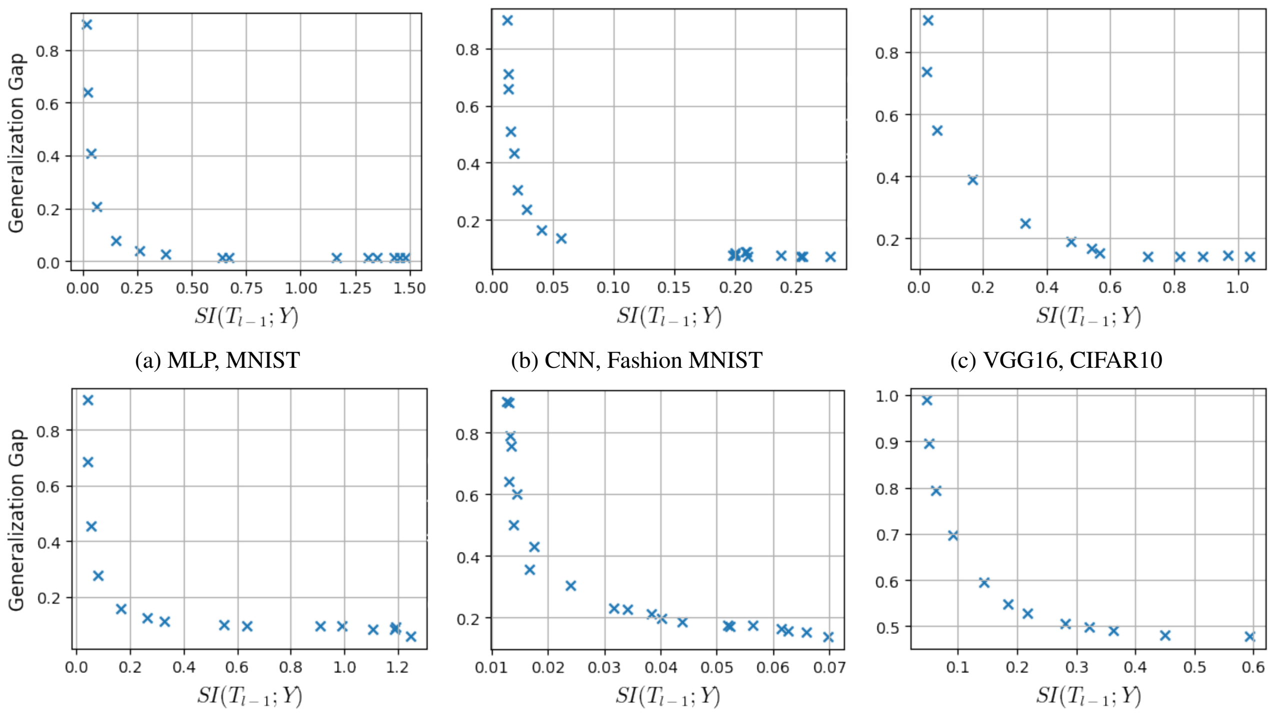
bels. Across varying degrees of label noise, we present the results for a 5-layer MLP trained on MNIST in Fig. 2 and a 5-layer CNN trained on Fashion MNIST (Xiao et al., 2017) in Fig. 3.

As the degree of label noise increases, the amount of usable information in all hidden layers decreases. In the presence of label noise, neural networks are prone to memorizing the noisy labels and consequently fail to generalize to test data with clean labels. In our experiments, we observe that $SI(T;Y)$ of all the hidden layers decreases with more label noise, as reflected in both Fig. 2 and Fig. 3. These findings confirm our hypothesis in Section 3 as more label noise leads to decreasing margin (Lin and Bradic, 2021) and increasing ID (Ma et al., 2018). As margin and ID are related to SMI through the lower bound shown in Theorem 1 and Corollary 2.1, SMI is expected to decrease as well with higher label noise.

Below, we provide a detailed discussion of the differences in SMI behavior observed between the MLP and CNN:

1. **MLP, MNIST:** In the presence of label noise (Fig. 2b and Fig. 2c), all the hidden layers experience an increase in $SI(T;Y)$ first before decreasing. This may indicate that the networks learn relevant and generalizable features first before proceeding to memorize. This observation agrees with the results from Arpit et al. (2017) and Stephenson et al. (2021). Furthermore, we observe that the $SI(T;Y)$ of deeper layers drop more compared to the earlier layers when the network is memorizing. Nevertheless, the SMI of deeper layers remain higher than that of earlier layers.
2. **CNN, Fashion MNIST:** In the absence of label noise (Fig. 3a), we observe that $SI(T;Y)$ of all hidden layers eventually decrease during training due to overfitting. Only in the last hidden layer, the $SI(T;Y)$ increases first before decreasing. The trends are maintained even in the presence of label noise (Fig. 3b and Fig. 3c). We also observe that the $SI(T;Y)$ of deeper layers drop more compared to the earlier layers (mainly layer 1) when the network is memorizing. Unlike the MLP case, the SMI of layer 4 is eventually lower than that of layer 1 during training.

These results support the argument in Stephenson et al. (2021) which states that memorization predominately occurs in deeper layers. We see that in both cases presented here, the earlier layers (e.g., layer 1) are less affected compared to the deeper layers (e.g., layer 4) by the presence of label noise. In Appendix C.2, we show results with MLP trained on Fashion MNIST, where we find that our current observations still hold. We also include the performance of



(a) MLP, MNIST

(b) CNN, Fashion MNIST

(c) VGG16, CIFAR10

(d) MLP, Fashion MNIST

(e) CNN, CIFAR10

(f) ResNet50, CIFAR100

Figure 4: The generalization gap vs $SI(T_{l-1};Y)$ where T_{l-1} is the penultimate layer for various combinations of architectures and datasets. In general, $SI(T_{l-1};Y)$ increases with decreasing generalization gap, and vice versa.

the networks during training in the same Appendix. Our empirical analysis also suggests that the deeper layers continue to memorize until the training accuracy reaches 100% while the earlier layers stop memorizing at an earlier stage in the training process.

4.2 Generalization

Our goal is to investigate how the SMI between features of the hidden layer and the labels relates to generalization gap. The generalization gap of a model is computed by taking the difference between training accuracy and test accuracy. SMI for different networks and datasets can have very different scales and thus cannot be compared directly. Instead, to obtain different values of generalization gap for the same architecture and dataset, we vary the dropout rate (between 0.1 and 0.5), the amount of label noise (between 0.005 and 1), and the use of batch normalization. In general, we have observed that the SMI of the deeper layers tends to be more correlated to generalization gap compared to that of the earlier layers. We present our results for the penultimate layer T_{l-1} in Fig. 4 for various combinations of networks (MLP, CNN, VGG16, and ResNet50 (He et al., 2016)) and datasets (MNIST, Fashion MNIST, CIFAR10, and CIFAR100). More details on the experiment can be found in Appendix C.3.

In general, we observe that the $SI(T_{l-1};Y)$ has an inverse relationship with generalization gap i.e., $SI(T_{l-1};Y)$ increases with decreasing generalization gap and vice versa. As the generalization gap decreases, the margin is likely to increase Bartlett et al. (2017); Neyshabur et al. (2018); Jiang et al. (2019) and the ID is likely to decrease (Nakada and Imaizumi, 2020) and thus SMI will likely increase as well (by Theorem 1 and Corollary 2.1). Since our results show that the SMI behaves as expected with generalization gap, it points to the possibility of refining the metric further (which will be discussed in Section 5. We hope that it would enable the metric to accurately predict the generalization gap and facilitate comparisons across various models and datasets.

5 REFLECTIONS

Summary of Contributions: As hypothesized in the motivation in Section 2.2, we indeed see across our experiments in Section 4 that SMI is an interesting and useful tool to analyze DNNs, and can capture both memorization and generalization. We argue that this is possible because SMI can encode geometric properties of feature distributions as discussed in Section 3. One interesting observation from our experiment in Section 4.1 is that in the presence of label noise, the earlier layers tends to stop memorizing at an earlier stage while the deeper layers continue to memorize until 100% training accuracy is achieved.

At its core, the SMI $SI(X;Y)$, like the MI $I(X;Y)$, is a measure of shared information between X and Y, and they share common properties (Goldfeld and Greenewald, 2021). However, as we see in Section 2.4, unlike MI, SMI can increase with more processing of the features as portrayed by the increase in SMI with depth in a trained DNN. This characteristic is consistent with observations made using visualization methods on the hidden layers Zeiler and Fergus (2014), which reveal that deeper layers generate more discriminative features that lead to better prediction of the labels.

SMI and $\mathcal{V}$-Information: As described in Section 2.2, SMI can be categorized as a measure of *average usable information* in the presence of computational constraints, similar to $\mathcal{V}$-information proposed in Xu et al. (2020), which looks at the *largest usable information* instead. We hypothesize that this property of SMI allows it to encode more geometrical aspects of the distribution of the features X, such as the spherical soft-margin (Theorem 1) and its ID (Corollary 2.1). It is important to note that, as experiments in Appendix A.3 and Appendix A.6 show, MI does not encode these properties and stays fixed in response to any changes in margin and ID. While estimating margin and ID directly is challenging in high dimensions, SMI is a measure that is scalable to high dimensions and thus we argue that it is more suitable for the analysis of DNNs.

Furthermore, as $SI(X;Y)$ computes the average SMI across all linear projections of X and Y, it is fundamentally different from $\mathcal{V}$-information, which encodes the *largest* MI among all projections. Note that this also prevents $\mathcal{V}$-information from encoding properties such as the classifier's hard-margin and its intrinsic dimensionality, when $P(X|Y=0)$ and $P(X|Y=1)$ can be separated via a hyperplane. As SMI considers the *average* usable information instead of the *largest*, we see that it can additionally encode such geometrical properties of the feature distribution.

SMI and Bayesian DNNs: In addition to the prior discussed motivation for SMI, we find that there lies a deeper and more general significance to averaging the MI of projections, in the context of Bayesian Deep Learning (Goan and Fookes, 2020). There, the weights W of a neural network are modelled as a random variable with a posterior distribution $W \sim P(W|D)$, where D is the training dataset $\{(X_1,Y_1),(X_2,Y_2),...,(X_m,Y_m)\}$, where $X \in \mathbb{R}^d$ and $Y \in \{0,1\}$. Here, we find that the following result holds.

Proposition 1. *We consider a Bayesian Neural Network (BNN) which can be represented as the feedforward graph $X \to T \to \widehat{y}$, where X is the input data, T is the penultimate layer of the network, and $\widehat{y} \in \mathbb{R}$ is the one-dimensional network output logit. Let $W_{opt} \in \mathbb{R}^{dim(W)}$ represent the fixed trained weights for the penultimate layer. Let us assume that the posterior $P(W|D) \sim \mathcal{N}(W_{opt}, diag(\boldsymbol{\sigma}))$, for some $\boldsymbol{\sigma} \in \mathbb{R}^{dim(W)}$ and $diag(.)$ is the diagonal matrix operator. Then, we have that*

$$I(\widehat{y};Y) \geq SI(T;Y). \tag{6}$$

The above result shows us that a larger SMI between the penultimate layer and the last layer of a neural network can potentially lead to a larger $I(\widehat{Y};Y)$, which points to better generalization. This is indeed observed across our experiments, as larger SMI usually accompanies better generalization performance and vice-versa (see Section 4.2). Intuitively, this follows from the fact that SMI averages the MI for all projections (see (1)). As the uniform distribution over the projections is equivalent to the uniform distribution over the penultimate layer weights W, SMI indirectly connects to BNNs by considering the *worst-case* uncertainty over the weights.

SMI with non-linear functions: Note that without the averaging over the uniform distribution, (6) will not hold. This shows the significance of averaging over all projections, which represents a linear function class. Naturally, an extension of this approach is to consider more complex, non-linear function classes. An interesting generalization of SMI between X and discrete Y is thus estimating $\mathbb{E}_{f\in\mathcal{F}}\left[I(f(X);Y)\right]$ for $\mathcal{F}$ which can represent non-linear function classes. Using appropriate $\mathcal{F}$ that better emulates the network function structure between the input and the output, these SMI variants will more readily relate to label uncertainty (see Proposition 1). The theoretical and empirical observations in this work indicate that averaging-based MI measures like SMI, and its generalizations, represents a promising approach to studying DNNs.

Dimensionality-reduced SMI variants: A potential avenue of future research lies in exploring variants of SMI for convolutional layers, as flattened convolutional layers usually are of very high dimensionality, which biases the SMI to have lower values. One way to alleviate this issue is to reduce the dimensionality of the layers before computing the SMI, e.g., one such example is pooling. This leads to a plethora of possible approaches for defining SMI for convolutional layers and other equivariant CNN layers (e.g., scale and rotation equivariant CNNs), which can be explored in future work.

SMI and Network Regularization: Finally, throughout our experiments in Section 4, we see that the SMI between the network layers (T) and the outputs (Y) can be indicative of the generalization performance. These observations overall point to a potential actionable manner in which we can regularize neural networks, by forcing their layers to have larger $SI(T;Y)$. Therefore, a possible extension of this work is to actively regularize the network layers during training, to prevent their $SI(T;Y)$ from decreasing. The results in our work motivate the investigation of novel methods for regularizing networks as a promising avenue for further research.

Acknowledgements

This research is supported by A*STAR, CISCO Systems (USA) Pte. Ltd and the National University of Singapore under its Cisco-NUS Accelerated Digital Economy Corporate Laboratory (Award I21001E0002), and by the Singapore Ministry of Education under Grant MOE2019-T2-2-171. Additionally, we would like to thank the members of the Kent Ridge AI research group at the National University of Singapore for helpful feedback and interesting discussions on this work.

References

A. Achille and S. Soatto. Where is the information in a deep neural network? *CoRR*, abs/1905.12213, 2019.

A. Ansuini, A. Laio, J. H. Macke, et al. Intrinsic dimension of data representations in deep neural networks. In *Advances in Neural Information Processing Systems 32*, 2019.

D. Arpit, S. Jastrzebski, N. Ballas, et al. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 233–242, 2017.

P. Bartlett and J. Shawe-Taylor. Generalization performance of support vector machines and other pattern classifiers. *Advances in Kernel methods—support vector learning*, pages 43–54, 1999.

P. L. Bartlett, D. J. Foster, and M. Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems 30*, pages 6240–6249, 2017.

M. I. Belghazi, A. Baratin, S. Rajeswar, et al. Mutual information neural estimation. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 530–539, 2018.

T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, 2001. ISBN 9780471062592. doi: 10.1002/0471200611.

L. P. F. Garcia, A. C. P. L. F. de Carvalho, and A. C. Lorena. Effect of label noise in the complexity of classification problems. *Neurocomputing*, 160:108–119, 2015.

B. C. Geiger. On information plane analyses of neural network classifiers - A review. *CoRR*, abs/2003.09671, 2020.

R. Ghosh and M. Motani. Network-to-network regularization: Enforcing occam's razor to improve generalization. *Advances in Neural Information Processing Systems*, 34: 6341–6352, 2021.

E. Goan and C. Fookes. Bayesian neural networks: An introduction and survey. *CoRR*, abs/2006.12024, 2020.

Z. Goldfeld and K. H. Greenewald. Sliced mutual information: A scalable measure of statistical dependence. In *Advances in Neural Information Processing Systems 34*, pages 17567–17578, 2021.

Z. Goldfeld, K. H. Greenewald, T. Nuradha, et al. k-sliced mutual information: A quantitative study of scalability with dimension. *CoRR*, abs/2206.08526, 2022.

K. He, X. Zhang, S. Ren, et al. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognitio*, pages 770–778, 2016.

M. A. Hearst, S. T. Dumais, E. Osuna, et al. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28, 1998.

Y. Jiang, D. Krishnan, H. Mobahi, et al. Predicting the generalization gap in deep networks with margin distributions. In *7th International Conference on Learning Representations*, 2019.

Y. Jiang, B. Neyshabur, H. Mobahi, et al. Fantastic generalization measures and where to find them. In *8th International Conference on Learning Representations*, 2020.

K. Kawaguchi, L. P. Kaelbling, and Y. Bengio. Generalization in deep learning. *CoRR*, abs/1710.05468, 2017.

V. Koltchinskii and D. Panchenko. Empirical margin distributions and bounding the generalization error of combined classifiers. *The Annals of Statistics*, 30(1):1–50, 2002.

A. Kraskov, H. Stögbauer, and P. Grassberger. Estimating mutual information. *Physical review E*, 69(6):066138, 2004.

A. Krizhevsky. Non-parametric entropy estimation toolbox (npeet). Technical report, 2000.

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

Y. LeCun, C. Cortes, and C. Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

J. Z. Lin and J. Bradic. Learning to combat noisy labels via classification margins. *CoRR*, abs/2102.00751, 2021.

X. Ma, Y. Wang, M. E. Houle, et al. Dimensionality-driven learning with noisy labels. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 3361–3370, 2018.

D. McAllester and K. Stratos. Formal limitations on the measurement of mutual information. In *The 23rd International Conference on Artificial Intelligence and Statistics*, volume 108, pages 875–884, 2020.

A. Montanari, F. Ruan, Y. Sohn, et al. The generalization error of max-margin linear classifiers: High-dimensional asymptotics in the overparametrized regime. *arXiv preprint arXiv:1911.01544*, 2019.

R. Nakada and M. Imaizumi. Adaptive approximation and generalization of deep neural network with intrinsic dimensionality. *J. Mach. Learn. Res.*, 21(174):1–38, 2020.

B. Neyshabur, S. Bhojanapalli, and N. Srebro. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. In *6th International Conference on Learning Representations*, 2018.

K. B. Oldham, J. C. Myland, and J. Spanier. The incomplete beta function b (v, μ, x). In *An Atlas of Functions*, pages 603–609. 2008.

B. Poole, S. Ozair, A. van den Oord, et al. On variational bounds of mutual information. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 5171–5180, 2019.

A. M. Saxe, Y. Bansal, J. Dapello, et al. On the information bottleneck theory of deep learning. In *6th International Conference on Learning Representations*, 2018.

R. Shwartz-Ziv and N. Tishby. Opening the black box of deep neural networks via information. *CoRR*, abs/1703.00810, 2017.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations*, 2015.

H. Song, M. Kim, D. Park, et al. Learning from noisy labels with deep neural networks: A survey. *CoRR*, abs/2007.08199, 2020.

J. Song and S. Ermon. Understanding the limitations of variational mutual information estimators. In *8th International Conference on Learning Representations*, 2020.

C. Stephenson, S. Padhy, A. Ganesh, et al. On the geometry of generalization and memorization in deep neural networks. In *9th International Conference on Learning Representations*, 2021.

Y. Tay, M. Dehghani, D. Bahri, et al. Efficient transformers: A survey. *CoRR*, abs/2009.06732, 2020.

B. Van Rooyen, A. Menon, and R. C. Williamson. Learning with symmetric label noise: The importance of being unhinged. *Advances in neural information processing systems*, 28, 2015.

Wikipedia contributors. Spherical cap — Wikipedia, the free encyclopedia, 2022. URL `https://en.wikipedia.org/w/index.php?title=Spherical_cap&oldid=1088721584`. [Online; accessed 26-May-2022].

S. Wongso, R. Ghosh, and M. Motani. Understanding deep neural networks using sliced mutual information. In *IEEE International Symposium on Information Theory*, pages 133–138, 2022.

H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017.

A. Xu and M. Raginsky. Information-theoretic analysis of generalization capability of learning algorithms. In *Advances in Neural Information Processing Systems 30*, pages 2524–2533, 2017.

Y. Xu, S. Zhao, J. Song, et al. A theory of usable information under computational constraints. In *8th International Conference on Learning Representations*, 2020.

M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *Computer Vision - ECCV 2014 - 13th European Conference*, volume 8689 of *Lecture Notes in Computer Science*, pages 818–833, 2014.

C. Zhang, S. Bengio, M. Hardt, et al. Understanding deep learning requires rethinking generalization. In *5th International Conference on Learning Representations*, 2017.

C. Zhang, S. Bengio, M. Hardt, et al. Understanding deep learning (still) requires rethinking generalization. *Commun. ACM*, 64(3):107–115, 2021.

SUPPLEMENTARY MATERIAL

To allow ease of access and improve readability, we present a short summary of our supplementary materials. All citations in this appendix are to the reference list in the main paper. The supplementary materials include:

A ON THE THEORY

A.1 Proof of Theorem 1

Theorem 1. ***(Margin-based lower bound)*** *We are given RVs $X \in \mathbb{R}^{d_x}$ and $Y \in \{0,1\}$. Assume $P(Y=0) = P(Y=1) = 0.5$ and that X, Y are $(R_1, R_2, m_g, \epsilon)$-SSM-separated. Given this, we then have*

$$(1 - H(\epsilon, 1-\epsilon))\, B_{\gamma(m_g,R_1,R_2)}\left(\frac{d_x-1}{2}, \frac{1}{2}\right) \le SI(X;Y) \le 1, \tag{7}$$

where $\gamma(m_g, R_1, R_2) = \frac{m_g}{m_g+R_1+R_2}\left(2 - \frac{m_g}{m_g+R_1+R_2}\right)$, $B_x(a,b)$ is the regularized incomplete beta function (Oldham et al., 2008), and $H(p_1,p_2) = -p_1 \log p_1 - p_2 \log p_2$ is the entropy function.

Proof. Let us denote the encapsulating spheres via S_1 and S_2, as per the SSM-separation definition. Consider the line segment connecting the center of S_1 denoted by C_1 to the center of S_2 denoted by C_2. Translation invariance of SMI follows from the fact that MI $I(x;y)$ itself is shift invariant when y is discrete. We additionally apply the rotation invariant property of SMI (Goldfeld and Greenewald, 2021) to re-position the origin at C, such that $||C - C_1|| = R_1 + m_g/2$ and $||C - C_2|| = R_2 + m_g/2$.

Now, with the origin at C, let us denote the vector representation of C_1 by $-(R_1 + m_g/2)\boldsymbol{u_1}$, which implies that the vector representation of C_2 would be $(R_2 + m_g/2)\boldsymbol{u_1}$. Here $\boldsymbol{u_1}$ represents a unit vector. Next, consider any point $\boldsymbol{P_1}$ inside S_1, which has a vector representation of

$$\boldsymbol{P_1} = -(R_1 + m_g/2)\boldsymbol{u_1} + \tau \boldsymbol{u_1'}, \tag{8}$$

where $0 \le \tau \le R_1$, and $\boldsymbol{u_1'}$ represents any unit vector. Similarly, we consider any point $\boldsymbol{P_2}$ inside S_2, which has a vector representation of

$$\boldsymbol{P_2} = (R_2 + m_g/2)\boldsymbol{u_1} + \tau \boldsymbol{u_2'}, \tag{9}$$

where $0 \le \tau \le R_2$, and $\boldsymbol{u_2'}$ represents any unit vector. Now, as SMI projects the points onto a unit vector sampled from the d_x dimensional sphere $\mathbb{S}^{d_x-1}$ (as per convention in Goldfeld and Greenewald (2021)), let us represent this unit vector random variable via $\boldsymbol{u_w}$. Given $\boldsymbol{u_w}$, we estimate the maximum value of the projection of $\boldsymbol{P_1}$ onto $\boldsymbol{u_w}$, represented by γ_1, when iterating through all points within S_1, as follows

$$\begin{aligned}\gamma_1 &= \max_{\boldsymbol{u_1'},\tau} -(R_1 + m_g/2)\boldsymbol{u_1^T u_w} + \tau \boldsymbol{u_1^{\prime T} u_w} && (10)\\ &= \max_{\tau} -(R_1 + m_g/2)\boldsymbol{u_1^T u_w} + \tau && (11)\\ &= -(R_1 + m_g/2)\boldsymbol{u_1^T u_w} + R_1. && (12)\end{aligned}$$

Similarly, we estimate the minimum value of the projection of $\boldsymbol{P_2}$ onto $\boldsymbol{u_w}$, represented by γ_2, when iterating through all points within S_2, as follows

$$\begin{aligned}\gamma_2 &= \min_{\boldsymbol{u_2'},\tau} (R_2 + m_g/2)\boldsymbol{u_1^T u_w} + \tau \boldsymbol{u_2^{\prime T} u_w} && (13)\\ &= \min_{\tau} (R_2 + m_g/2)\boldsymbol{u_1^T u_w} - \tau && (14)\\ &= (R_2 + m_g/2)\boldsymbol{u_1^T u_w} - R_2. && (15)\end{aligned}$$

Now, if $\gamma_1 \leq \gamma_2$, there is no overlap between the projections of the sphere S_1 and S_2, when projected using the unit vector u_w. This yields the constraint,

$$\begin{aligned}-(R_1 + m_g/2)\boldsymbol{u_1^T u_w} + R_1 &\leq (R_2 + m_g/2)\boldsymbol{u_1^T u_w} - R_2 && (16)\\ (R_1 + R_2 + m_g)\boldsymbol{u_1^T u_w} &\geq R_1 + R_2 && (17)\\ \boldsymbol{u_1^T u_w} &\geq \frac{R_1 + R_2}{R_1 + R_2 + m_g}. && (18)\end{aligned}$$

Thus, as both u_1 and u_w are unit vectors, we can substitute $\cos\theta_w = \boldsymbol{u_1^T u_w}$, which, combined with (18), yields

$$\theta_w \leq \cos^{-1}\left(\frac{R_1 + R_2}{R_1 + R_2 + m_g}\right). \tag{19}$$

Thus, when $\theta_w \leq \cos^{-1}\left(\frac{R_1+R_2}{R_1+R_2+m_g}\right)$, there would not be any overlap between the projections of the points in S_1 and S_2. As $\boldsymbol{u_w}$ is sampled uniformly in estimating the SMI, probability that the projections of S_1 and S_2 do not overlap, is the same as ratio the surface area of a hyperspherical cap (Wikipedia contributors, 2022) of radius R_1, and a height of $h = R_1 - R_1\cos\theta_w = R_1\left(1 - \frac{R_1+R_2}{R_1+R_2+m_g}\right)$ to half the area of S_1. Let us then denote this probability via $P(S_1^T\boldsymbol{u_w} \cap S_2^T\boldsymbol{u_w} = \emptyset)$. Using the expression of the area of the hyperspherical cap (Wikipedia contributors, 2022), we have that

$$P(S_1^T\boldsymbol{u_w} \cap S_2^T\boldsymbol{u_w} = \emptyset) = B_{\frac{2R_1h-h^2}{R_1^2}}\left(\frac{d_x - 1}{2}, \frac{1}{2}\right), \tag{20}$$

and substituting for $h = R_1\left(1 - \frac{R_1+R_2}{R_1+R_2+m_g}\right)$, we ultimately have

$$P(S_1^T\boldsymbol{u_w} \cap S_2^T\boldsymbol{u_w} = \emptyset) = B_{\gamma(m_g,R_1,R_2)}\left(\frac{d_x - 1}{2}, \frac{1}{2}\right), \tag{21}$$

where $\gamma(m_g, R_1, R_2) = \frac{m_g}{m_g+R_1+R_2}\left(2 - \frac{m_g}{m_g+R_1+R_2}\right)$. Next, we can re-write the SMI $SI(X;Y)$ as,

$$\begin{aligned}SI(X;Y) &= SI(X;Y|S_1^T\boldsymbol{u_w} \cap S_2^T\boldsymbol{u_w} = \emptyset)P(S_1^T\boldsymbol{u_w} \cap S_2^T\boldsymbol{u_w} = \emptyset) && (22)\\ &\quad + SI(X;Y|S_1^T\boldsymbol{u_w} \cap S_2^T\boldsymbol{u_w} \neq \emptyset)P(S_1^T\boldsymbol{u_w} \cap S_2^T\boldsymbol{u_w} \neq \emptyset) && (23)\\ &\geq SI(X;Y|S_1^T\boldsymbol{u_w} \cap S_2^T\boldsymbol{u_w} = \emptyset)B_{\gamma(m_g,R_1,R_2)}\left(\frac{d_x - 1}{2}, \frac{1}{2}\right) + 0 && (24)\end{aligned}$$

Next, we use the constraints imposed by the SSM-separation criterion, which states that $P(X \in S_1|Y = 0) = P(X \in S_2|Y = 1) = 1 - \epsilon$. Let us consider any $\boldsymbol{u_w}$ such that $S_1^T\boldsymbol{u_w} \cap S_2^T\boldsymbol{u_w} = \emptyset$. Note that, this implies that the maximum of S_1's projection (γ_1) is less than the minimum of S_2's projection. This implies that we can separate the distribution of their projections w.r.t a point $p_c \in \mathbb{R}$ in the real line, such that all of S_1's projections are to the left of P_c and all of S_2's projections are to the right.

Let us denote the random variable which is the projection of X onto u_w, by x_w. Next, we define another random variable ρ as follows:

$$\rho = \begin{cases} 0 & \text{if } X^T \boldsymbol{u_w} < p_c \\ 1 & \text{if } X^T \boldsymbol{u_w} \geq p_c \end{cases}$$

Given the definition of SSM-separation, note that $P(\rho = 0) = P(\rho = 1) = 0.5$. Furthermore, note that $h(Y|\rho) = H(\epsilon, 1-\epsilon)$, where $H(\epsilon, 1-\epsilon)$ has the same definition as given in the Theorem. Now, we denote the mutual information between $X^T \boldsymbol{u_w}$ and Y, by $I(X^T \boldsymbol{u_w}; Y)$. Let us denote h as the entropy operator. We can write

$$h(X^T \boldsymbol{u_w}|Y, \rho) + h(Y|\rho) = h(Y|X^T \boldsymbol{u_w}, \rho) + h(X^T \boldsymbol{u_w}|\rho) \tag{25}$$

Note that $h(Y|X^T \boldsymbol{u_w}, \rho) = h(Y|X^T \boldsymbol{u_w}))$, and thus we can write,

$$\begin{aligned} h(Y) - h(Y|X) &= h(Y) + h(X^T \boldsymbol{u_w}|\rho) - h(X^T \boldsymbol{u_w}|Y, \rho) - h(Y|\rho) && (26)\\ I(X^T \boldsymbol{u_w}; Y) &= h(Y) - h(Y|\rho) + I(X^T \boldsymbol{u_w}; Y|\rho) && (27)\\ I(X^T \boldsymbol{u_w}; Y) &\geq h(Y) - h(Y|\rho) = 1 - H(\epsilon, 1-\epsilon) && (28) \end{aligned}$$

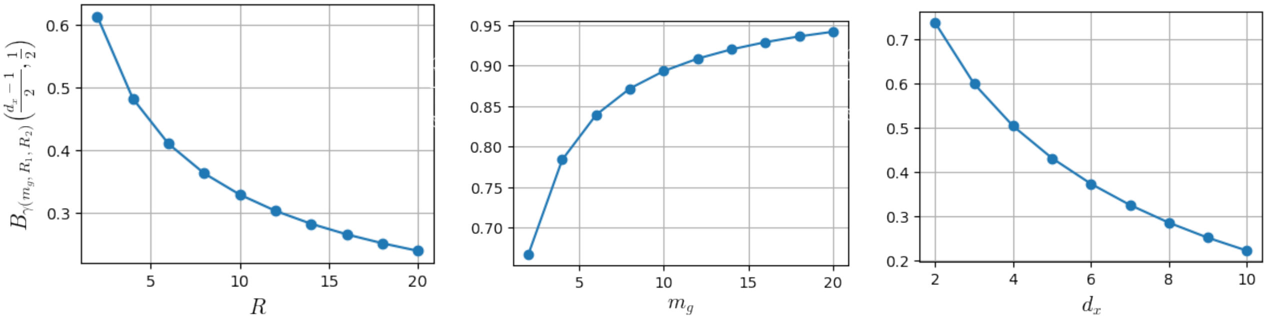
Thus, for all $\boldsymbol{u_w}$ such that $S_1^T \boldsymbol{u_w} \cap S_2^T \boldsymbol{u_w} = \emptyset$, we have that $I(X^T \boldsymbol{u_w}; Y) \geq 1 - H(\epsilon, 1-\epsilon)$. As SMI represents the average of $I(X^T \boldsymbol{u_w}; Y)$ over $\boldsymbol{u_w}$, using the result in (24), we can finally write

$$\begin{aligned} SI(X;Y) &\geq SI(X;Y|S_1^T \boldsymbol{u_w} \cap S_2^T \boldsymbol{u_w} = \emptyset) B_{\gamma(m_g, R_1, R_2)} \left(\frac{d_x - 1}{2}, \frac{1}{2} \right) && (29)\\ &= \mathbb{E}_{\boldsymbol{u_w}} \left[I(X^T \boldsymbol{u_w}; Y) | S_1^T \boldsymbol{u_w} \cap S_2^T \boldsymbol{u_w} = \emptyset \right] B_{\gamma(m_g, R_1, R_2)} \left(\frac{d_x - 1}{2}, \frac{1}{2} \right) && (30)\\ &\geq \left(1 - H(\epsilon, 1-\epsilon)\right) B_{\gamma(m_g, R_1, R_2)} \left(\frac{d_x - 1}{2}, \frac{1}{2} \right). && (31) \end{aligned}$$

Furthermore, for any $\boldsymbol{u_w}$, we have that $I(X^T \boldsymbol{u_w}; Y) = h(Y) - h(Y|X^T \boldsymbol{u_w}) \leq h(Y) = 1$, and thus $SI(X;Y) \leq 1$. This yields the result.

A.2 On the Regularized Incomplete Beta Function

In Fig. 5, we show the dependence between the regularized incomplete beta function $B_x(a, b)$ in Theorem 1 and its parameters. In Fig. 5a, we fix the margin $m_g = 3$ and data dimensionality $d_x = 2$ and assume that the radii of two hyperspheres are the same i.e., $R_1 = R_2 = R$. We show that $B_x(a, b)$ decreases with increasing R. In Fig. 5b, we fix the radii of the hyperspheres $R_1 = R_2 = 1$ and data dimensionality $d_x = 2$. We show that $B_x(a, b)$ increases with increasing margin m_g. In Fig. 5c, we fix the radii of the hyperspheres $R_1 = R_2 = 1$ and margin $m_g = 3$. We show that $B_x(a, b)$ decreases with increasing data dimensionality d_x.



(a) vary radius of hypersphere

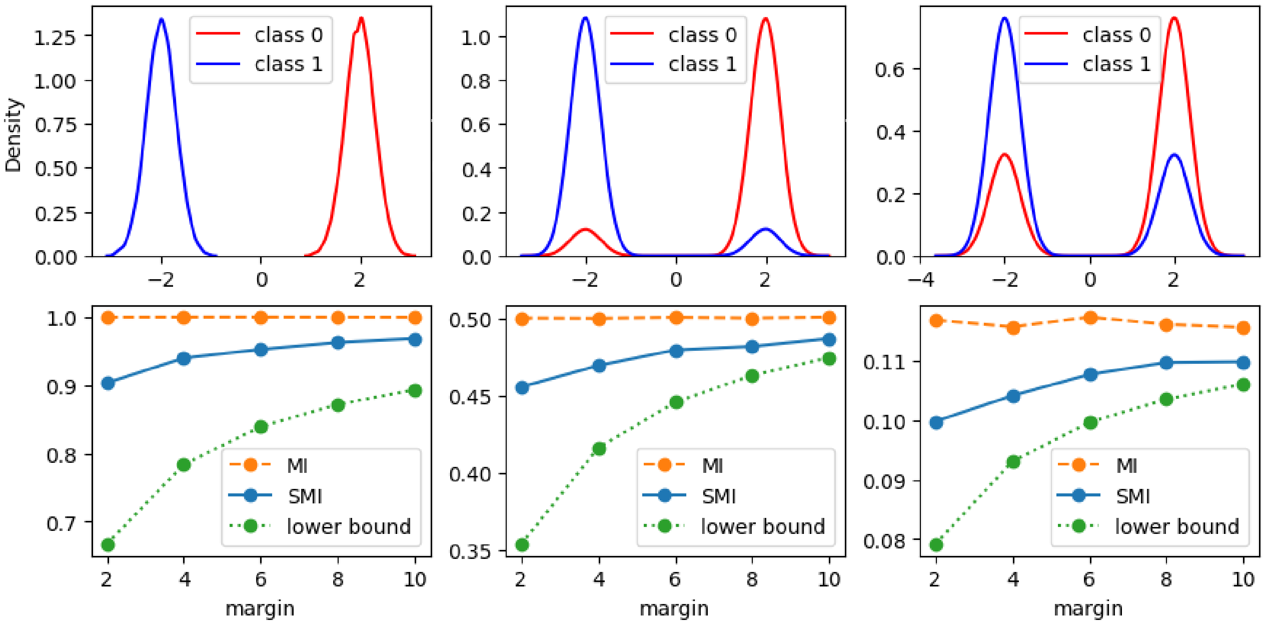
(b) vary margin

(c) vary data dimensionality

Figure 5: Plots to show how the regularized incomplete beta function in Theorem 1 vary with its parameters. It increases with decreasing radius of the hypersphere, increasing margin and decreasing data dimensionality.

A.3 Empirical Results for SMI and Margin

We conduct an experiment to show how MI, SMI, and the margin-based lower bound (in Theorem 1) vary with margin m_g and degree of overlap ϵ. We generate 100,000 data samples X from truncated 2D Gaussian variables, and assign labels Y of 0 or 1 to them. We assume $P(Y = 0) = P(Y = 1) = 0.5$ as before. We illustrate the 1D version of the data distribution in Fig. 6 (top) for different ϵ values. In this case, X and Y are $(1, 1, m_g, \epsilon)$-SSM-separated. We then compute the MI $I(X; Y)$, the SMI $SI(X; Y)$ and the lower bound in Theorem 1. In Fig. 6a, there is no overlap in the feature distribution of the two classes ($\epsilon = 0$) while in Fig. 6b and Fig. 6c, we allow some overlap in the feature distribution of the two classes ($\epsilon = 0.1$ and $\epsilon = 0.3$ respectively). We present the results for the 3 cases in Fig. 6 (bottom). In all the cases, $I(X; Y)$ stays about the same but $SI(X; Y)$ and the lower bound clearly increases with margin. This empirically proves our argument that SMI is sensitive to margin while MI is not.



(a) 0 overlap

(b) 0.1 overlap

(c) 0.3 overlap

Figure 6: (Top) Illustrations of feature distributions of 1D truncated gaussian variables with different degrees of overlap between the feature distributions of the two different classes. (Bottom) For the different degrees of overlap, MI $I(X; Y)$ stays constant while both the SMI $SI(X; Y)$ and the margin-based lower bound increase with margin.

A.4 Proof of Theorem 2

Theorem 2. *Assume that the support of $P(X)$ lies within a linear subspace W of K dimensions. Let W be represented by the orthonormal basis set $\{u_i\}_{i=1}^K$, and the center of W be at a distance of $\boldsymbol{\mu}$ from the origin. Let $\boldsymbol{U}$ be a matrix with columns $\{u_i\}_{i=1}^K$. Then, we have*

$$SI(X;Y) = SI\left(\boldsymbol{U}^T\left(X-\boldsymbol{\mu}\right);Y\right). \tag{32}$$

Proof. Let $\boldsymbol{U_\phi} \in \mathbb{R}^d \times \mathbb{R}^{d-K}$ be the null space matrix of the linear subspace W, which is represented by $\boldsymbol{U} \in \mathbb{R}^d \times \mathbb{R}^K$. Then, the concatenated matrix $\boldsymbol{U}_c = [\boldsymbol{U}, \boldsymbol{U_\phi}] \in \mathbb{R}^d \times \mathbb{R}^d$ represents a rotation of the axes in $\mathbb{R}^d$. Next, we note that, as shown in Goldfeld and Greenewald (2021), $SI(X;Y)$ is invariant to rotations to the co-ordinate axes w.r.t X. Furthermore, in what follows we also make use of the property that $SI(\alpha X;Y) = SI(X;Y)$, when Y is discrete Goldfeld and Greenewald (2021). We have,

$$SI(X;Y) = SI\left(\boldsymbol{U}_c^T\left(X-\boldsymbol{\mu}\right);Y\right) \tag{33}$$

$$= SI\left(\left[\boldsymbol{U}^T\left(X-\boldsymbol{\mu}\right), \mathbf{0}_{(1\times d-K)}\right];Y\right), \tag{34}$$

where $[\mathbf{0}]_{1\times d-K}$ represents the null matrix of size $(1\times d-K)$. Note that the above follows from the fact that $\boldsymbol{U_\phi}$ represents the null space of W, and X lies within W.

Next, we re-iterate the definition of $SI(X;Y)$ as follows. Let us define independent RVs $\Theta \sim$ Unif$(\mathbb{S}^{d_x-1})$ and $\Phi \sim$ Unif$(\mathbb{S}^{d_y-1})$. Note that in this context, $d_x = d$ and $d_y = 1$. Also, in what follows, we use the convention $E_{A\in B}[.]$ to represent $E_{A\sim Unif(B)}[.]$, where $Unif(B)$ represents the uniform distribution over the set B. The SMI between X and Y, for discrete Y then can be expressed as:

$$SI(X;Y) = \mathbb{E}_{\theta\in\mathbb{S}^{d_x-1},\phi\in\mathbb{S}^{d_y-1}}\left[I(\theta^T X;\phi^T Y)\right] \tag{35}$$

$$= \mathbb{E}_{\theta\in\mathbb{S}^{d_x-1},\phi\in\mathbb{S}^{d_y-1}}\left[I(\theta^T X;Y)\right] = \mathbb{E}_{\theta\in\mathbb{S}^{d_x-1}}\left[I(\theta^T X;Y)\right] \tag{36}$$

$$\tag{37}$$

For what follows, let us represent the set of all points in $\mathbb{R}^d$ which have a distance of R to the origin, via the set $S_{d-1}(R)$. Also, in what follows, we make use of the fact that $I(\alpha X;Y) = I(X;Y)$, for discrete Y, as MI is scale invariant w.r.t X when Y is discrete. Note that $S_{d_x-1} = S_{d_x-1}(1)$. Now, using (34), we have

$$SI(X;Y) = SI\left(\left[\boldsymbol{U}^T\left(X-\boldsymbol{\mu}\right), \mathbf{0}_{(1\times d-K)}\right];Y\right) \tag{38}$$

$$= \mathbb{E}_{\theta\in\mathbb{S}^{d_x-1}}\left[I(\theta^T\left[\boldsymbol{U}^T\left(X-\boldsymbol{\mu}\right), \mathbf{0}_{(1\times d_x-K)}\right];Y)\right] \tag{39}$$

$$= \mathbb{E}_{\theta\in\mathbb{S}^{d_x-1}}\left[I(\theta_{trunc}^T\boldsymbol{U}^T\left(X-\boldsymbol{\mu}\right);Y)\right] \tag{40}$$

$$= \mathbb{E}_R\mathbb{E}_{\theta_{trunc}\in\mathbb{S}^{d_x-K-1}(R)}\left[I(\theta_{trunc}^T\boldsymbol{U}^T\left(X-\boldsymbol{\mu}\right);Y)\right] \tag{41}$$

$$= \mathbb{E}_R\mathbb{E}_{\theta'_{trunc}\in\mathbb{S}^{d_x-K-1}(1)}\left[I(R(\theta'_{trunc})^T\boldsymbol{U}^T\left(X-\boldsymbol{\mu}\right);Y)\right] \tag{42}$$

$$= \mathbb{E}_R\mathbb{E}_{\theta'_{trunc}\in\mathbb{S}^{d_x-K-1}(1)}\left[I((\theta'_{trunc})^T\boldsymbol{U}^T\left(X-\boldsymbol{\mu}\right);Y)\right] \tag{43}$$

$$= \mathbb{E}_{\theta'_{trunc}\in\mathbb{S}^{d_x-K-1}}\left[I((\theta'_{trunc})^T\boldsymbol{U}^T\left(X-\boldsymbol{\mu}\right);Y)\right] = SI\left(\boldsymbol{U}^T\left(X-\boldsymbol{\mu}\right);Y\right) \tag{44}$$

$$\tag{45}$$

This completes the proof. □

A.5 Proof of Corollary 2.1

Corollary 2.1. ***(Margin- and ID-based lower bound)*** *We consider the same setting as in Theorem 1. However, we additionally assume that the support of $P(X)$ lies within a linear subspace W of K dimensions ($K \le d_x$). Furthermore, consider X and Y that are $(R_1, R_2, m_g, \epsilon)$ SSM-separated, via spheres of radius R_1 and R_2, whose centers lie in W. We then have,*

$$\left(1 - H\left(\epsilon, 1-\epsilon\right)\right) B_{\gamma(m_g,R_1,R_2)}\left(\frac{K-1}{2}, \frac{1}{2}\right) \le SI(X;Y) \le 1, \tag{46}$$

Proof. The result directly follows from Theorem 2, noting that when the center spheres of radii R_1 and R_2 lie on the linear subspace W, their projection on W have the same radii of R_1 and R_2 respectively. Next, we apply Theorem 2, which essentially states that SMI for the dimensionality-reduced data within the subspace W is the same as the original SMI. As the dimensionality of W is K instead of d_x, this implies that Theorem 1's result directly holds for $d_x = K$ instead, giving us:

$$(1 - H(\epsilon, 1-\epsilon)) B_{\gamma(m_g,R_1,R_2)}\left(\frac{K-1}{2}, \frac{1}{2}\right) \leq SI(X;Y) \leq 1. \tag{47}$$

This completes the proof. □

A.6 Empirical results for SMI and Intrinsic Dimensionality

We conduct an experiment to show how MI, SMI, and the lower bound (in Corollary 2.1) vary with intrinsic dimensionality (ID) K. We generate 100,000 data samples X from 10-dimensional gaussian mixtures, and assign labels Y of 0 or 1 to them. These gaussian mixtures are obtained from 10-dimensional gaussian variables with mean and variance only along the first dimension. The intrinsic dimensionality K is then varied by adding gaussian noise to the first K dimensions. We vary K from 2 to 10. We assume $P(Y=0) = P(Y=1) = 0.5$ as before and 0 overlap (can be extended to overlapping cases too). Note that X and Y are $(2.5, 2.5, 5, 0)$-SSM-separated in this case. We then compute the MI $I(X;Y)$, the SMI $SI(X;Y)$ and the lower bound in Corollary 2.1. We present the results in Fig. 7. We observe that $I(X;Y)$ stays constant at 1 while $SI(X;Y)$ and the lower bound decreases with increasing ID. This empirically proves our argument that SMI is sensitive to changes in ID while MI is not.

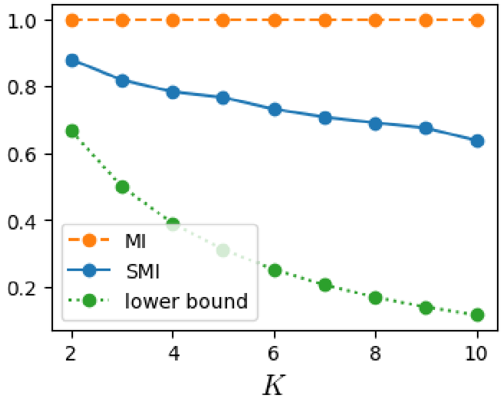


Figure 7: MI $I(X;Y)$ stays constant while both the SMI $SI(X;Y)$ and the lower bound in Corollary 2.1 decrease with increasing ID K.

A.7 Proof of Proposition 1

Proposition 1 *We consider a Bayesian Neural Network (BNN) which can be represented as the feedforward graph $X \rightarrow T \rightarrow \widehat{y}$, where X is the input data, T is the penultimate layer of the network, and $\widehat{y} \in \mathbb{R}$ is the one-dimensional network output logit. Let $W_{opt} \in \mathbb{R}^{dim(W)}$ represent the fixed trained weights for the penultimate layer. Let us assume that the posterior $P(W|D) \sim \mathcal{N}(W_{opt}, diag(\boldsymbol{\sigma}))$, for some $\boldsymbol{\sigma} \in \mathbb{R}^{dim(W)}$ and $diag(.)$ is the diagonal matrix operator. Then, we have that*

$$I(\widehat{y};Y) \geq SI(T;Y). \tag{48}$$

Proof. We note that for a single logit network with output $\widehat{y} = W^T T$, the with the posterior $P(W|D)$, we will have that

$$I(\widehat{y};Y) = \mathbb{E}_{W \sim P(W|D)}\left[I(W^T T;Y)\right] \tag{49}$$

Next, we note that as the uncertainty in W increases to the limiting case when $\boldsymbol{\sigma} \rightarrow \{\infty, \infty, ..., \infty\}$, this would provide a lower bound for $I(\widehat{y};Y)$. Furthermore, in the limiting case when $\boldsymbol{\sigma} \rightarrow \{\infty, \infty, ..., \infty\}$, we note that as $P(W|D)$ converges to the uniform distribution, it is independent of the center of the Gaussian $P(W|D) \sim \mathcal{N}(W_{opt}, diag(\boldsymbol{\sigma}))$. Thus, as $\boldsymbol{\sigma} \rightarrow \{\infty, \infty, ..., \infty\}$, we can express $P(W|D) \sim \mathcal{N}(0, diag(\boldsymbol{\sigma}))$.

For what follows, let us represent the set of all points in $\mathbb{R}^d$ which have a distance of τ to the origin, via the set $S_{d-1}(\tau)$. Also, we will make use of the fact that $I(\alpha X; Y) = I(X; Y)$, for discrete Y, as MI is scale invariant w.r.t X when Y is discrete. Lastly, we use the convention $E_{A \in B}[.]$ to represent $E_{A \sim Unif(B)}[.]$, where $Unif(B)$ represents the uniform distribution over B. The observations from the previous paragraph then yield:

$$\begin{aligned}
\mathbb{E}_{W \sim P(W|D)}\left[I(W^T T; Y)\right] &\geq \lim_{\boldsymbol{\sigma} \to \{\infty, \infty, \ldots, \infty\}} \mathbb{E}_{W \sim P(W|D)}\left[I(W^T T; Y)\right] && (50)\\
&= \lim_{\boldsymbol{\sigma} \to \{\infty, \infty, \ldots, \infty\}} \mathbb{E}_{W \sim \mathcal{N}(0, diag(\boldsymbol{\sigma}))}\left[I(W^T T; Y)\right] && (51)\\
&= \lim_{\sigma \to \infty} \mathbb{E}_{W \sim \mathcal{N}\left(0, \sigma \times diag(\boldsymbol{I}_{dim(W)})\right)}\left[I(W^T T; Y)\right] && (52)
\end{aligned}$$

where $dim(W)$ is the dimensionality of W and $\boldsymbol{I}_k$ represents the identity vector of dimensionality k. Note that $dim(T) = dim(W)$. The above can be subsequently represented as

$$\begin{aligned}
\mathbb{E}_{W \sim P(W|D)}\left[I(W^T T; Y)\right] &\geq \mathbb{E}_\tau \mathbb{E}_{W \in S_{dim(T)-1}(\tau)}\left[I(W^T T; Y)\right] && (53)\\
&= \mathbb{E}_\tau \mathbb{E}_{W' \in S_{dim(T)-1}(1)}\left[I(\tau(W')^T T; Y)\right] = \mathbb{E}_\tau\left[SI(T; Y)\right] = SI(T; Y). && (54)
\end{aligned}$$

This completes the proof. □

B ON SMI ESTIMATION

B.1 Psuedocode of the SMI Estimator

Algorithm 1 shows the pseudocode of our SMI estimator. Since in all our experiments Y is the discrete labels, we do not project the Y and only project X into one-dimensional variables. We adopt the KSG estimator (Kraskov et al., 2004) implemented using Non-parametric Entropy Estimation Toolbox (NPEET) with $k = 3$ (Krizhevsky, 2000) for the MI computation of SMI. The complexity of our SMI estimator, $\widehat{SI}(X, Y)$, is of the form $O(mn(d_x + \log n))$ where m is the number of slices, n is the number of samples, and d_x is the dimensions of X. For all our experiments, the SMI is computed using the training dataset.

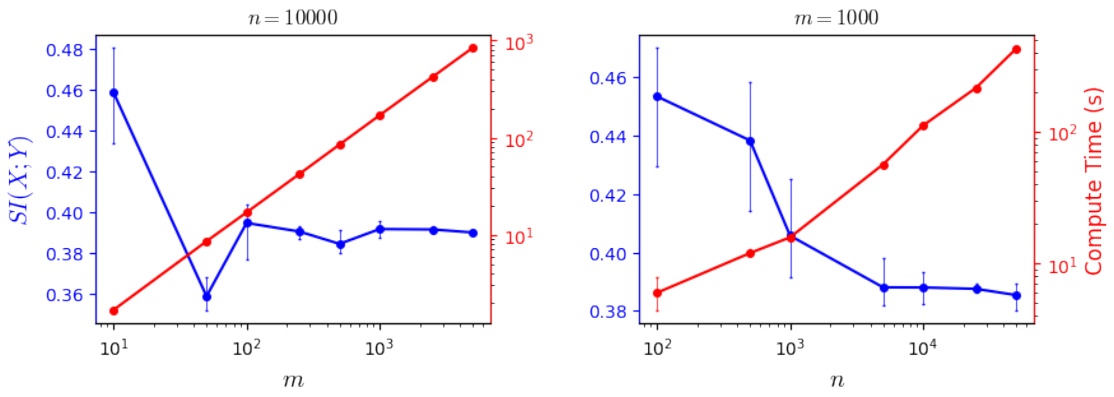
Algorithm 1 SMI Estimator (Goldfeld and Greenewald, 2021, Appendix B)

```
Require: $n$ (pairs of) samples $(X^n, Y^n)$ i.i.d. according to $P_{X,Y} \in \mathcal{P}(\mathbb{R}^{d_x} \times \mathbb{R})$, a scalar MI estimator $\hat{I}(\cdot;\cdot)$, and a chosen number of slices $m$.
  for $i = 1 : m$ do
      Sample $\Theta_i$ uniform on the sphere $\mathbb{S}^{d_x-1}$[1]
      Compute the MI estimate: $S_i \leftarrow \hat{I}\left(\left(\Theta_i^T X\right)^n; (Y)^n\right)$
  end for
  $\widehat{SI}_{n,m} \leftarrow \frac{1}{m}\sum_{i=1}^{m} S_i$
```

B.2 Convergence Behaviour of SMI Estimator

We study how our SMI estimator behaves with different number of slices m and different number of samples n used to estimate it. We compute the SMI between the inputs and the true labels for Fashion MNIST training dataset. For the SMI computation, we vary m and n. We show in Fig. 8, that the $SI(X; Y)$ converges as m and n increase (as expected). We also show that the compute time increases linearly with increasing m and n. In Fig. 8a, we observe that for m larger than 1000, the SMI estimation is stable and has converged. In Fig. 8b, we show that for n larger than 10000, the SMI estimation is relatively stable and has converged. For all of our experiments, we use $m = 1000$ and $n = 10000$.

[1] A uniform sample from $\mathbb{S}^{d-1}$ can be found by sampling a vector Z from a d-dimensional isotropic Gaussian and forming $Z/||Z||_2$.



(a) varying m

(b) varying n

Figure 8: The convergence behaviour of $SI(X;Y)$ where X and Y are the input features and labels of the Fashion MNIST training dataset respectively. The number of slices m and the number of samples n used to estimate the SMI are varied. The time spent to compute the SMI in each case is also plotted to show that it behaves linearly with m and n.

C EXPERIMENT DETAILS

C.1 SMI Behaviour in DNNs

Here, we provide the training details and for Section 2.4 in the main paper.

Experiment Details: We consider three different types of networks: 5-layer MLP (architecture is shown in Table 1), 6-convolutional layer CNN (architecture is shown in Table 2), and pre-trained VGG16 (architecture is shown in Table 3). For VGG16, all the weights are trained during training and the units of the last layer depends on the number of classes in the dataset. We consider 3 different datasets: MNIST, CIFAR10 and CIFAR100. The network is trained for 50 epochs using SGD optimizer with 0.01 learning rate and 0.9 momentum. The batch size is set to 32. The SMI is computed with $m = 500$ and $n = 10000$ on the training dataset.

Table 1: The architecture of the 5-layer MLP.

Layer Type	**Parameters**
Fully-Connected	1024 units, ReLU
Fully-Connected	1024 units, ReLU
Fully-Connected	1024 units, ReLU
Fully-Connected	1024 units, ReLU
Fully-Connected	10 units, Linear

Table 2: The architecture of the 6-convolutional layer CNN.

Layer Type	**Parameters**
Convolutional	32 filters, 3 × 3 kernels, strides=2, ReLU
Convolutional	32 filters, 3 × 3 kernels, strides=1, ReLU
Convolutional	32 filters, 3 × 3 kernels, strides=2, ReLU
Convolutional	32 filters, 3 × 3 kernels, strides=1, ReLU
Convolutional	32 filters, 3 × 3 kernels, strides=2, ReLU
Convolutional	10 filters, 1 × 1 kernels
Global Average Pooling	-

Table 3: The architecture of VGG16 for experiments in Section 2.4.

Layer Type	Parameters
VGG16 base network	Pre-trained on ImageNet
Fully-Connected	4096 units, ReLU
Fully-Connected	4096 units, ReLU
Fully-Connected	100 units, Linear

C.2 Memorization

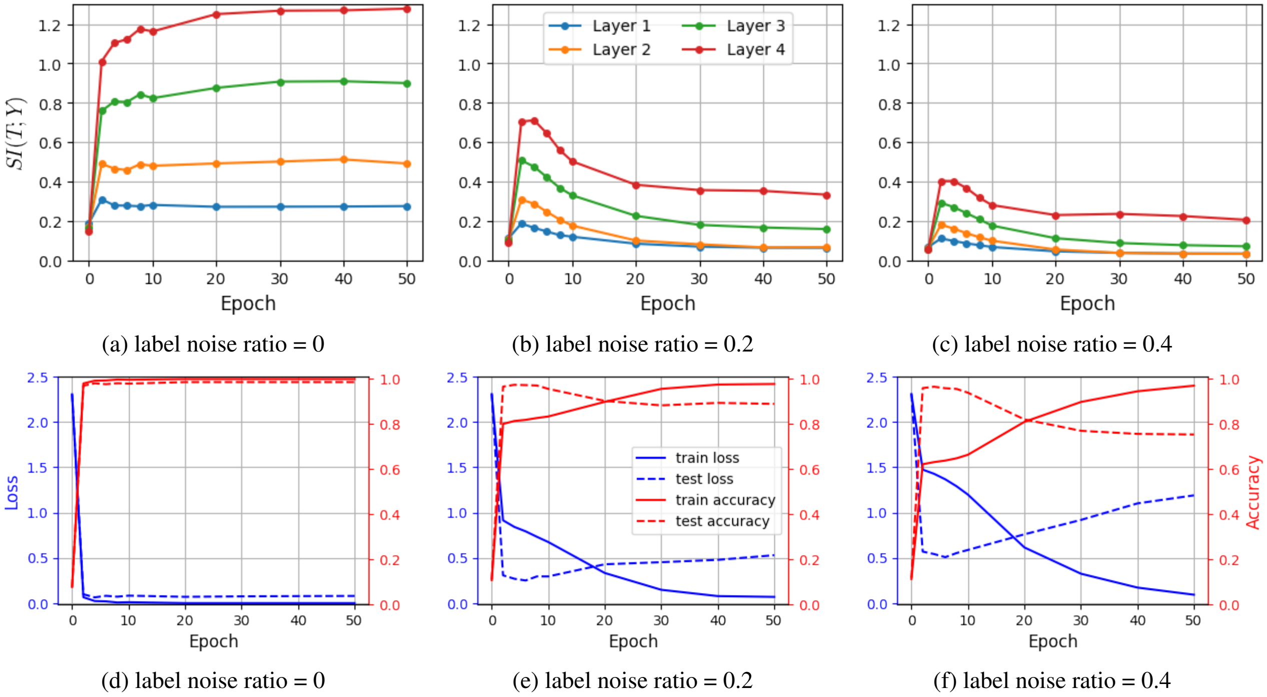
Here, we provide the training details and additional results for Section 4.1 in the main paper.

Experiment Details: We consider two different types of networks: 5-layer MLP (architecture is shown in Table 1) and 5-convolutional layer CNN (architecture is shown in Table 4). We consider 2 different datasets: MNIST and Fashion MNIST. The network is trained for 50 epochs (100 epochs for MLP, Fashion MNIST experiment) using SGD optimizer with 0.01 learning rate and 0.9 momentum. The learning rate decays by a factor of 0.9 when the training accuracy does not improve for the next 10 epochs. The batch size is set to 32. The label noise is induced by randomly changing the training labels for a fraction ϵ (referred to as the label noise ratio) of the dataset. The SMI is computed with $m = 500$ and $n = 10000$ on the training dataset (with corrupted labels).

Table 4: The architecture of the 5-convolutional layer CNN.

Layer Type	Parameters
Convolutional	512 filters, 3×3 kernels, strides=2, batch normalization, ReLU
Convolutional	512 filters, 3×3 kernels, strides=1, batch normalization, ReLU
Convolutional	512 filters, 3×3 kernels, strides=2, batch normalization, ReLU
Convolutional	512 filters, 3×3 kernels, strides=1, batch normalization, ReLU
Convolutional	10 filters, 1×1 kernels
Global Average Pooling	-

In Fig. 9, we consider MLP model with MNIST and show the $SI(T;Y)$ for all the layers as well as the model performance during training for different degrees of label noise. In Figure 10, we consider CNN model with Fashion MNIST and show the $SI(T;Y)$ for all the layers as well as the model performance during training for different degrees of label noise. These are the experiments considered in Section 4.1 in the main paper. Please refer to Section 4.1 for discussion of results. Additionally, we consider another case: MLP trained with Fashion MNIST (Fig. 11). We arrive at the same conclusions as the ones discussed in the main paper.



(a) label noise ratio = 0

(b) label noise ratio = 0.2

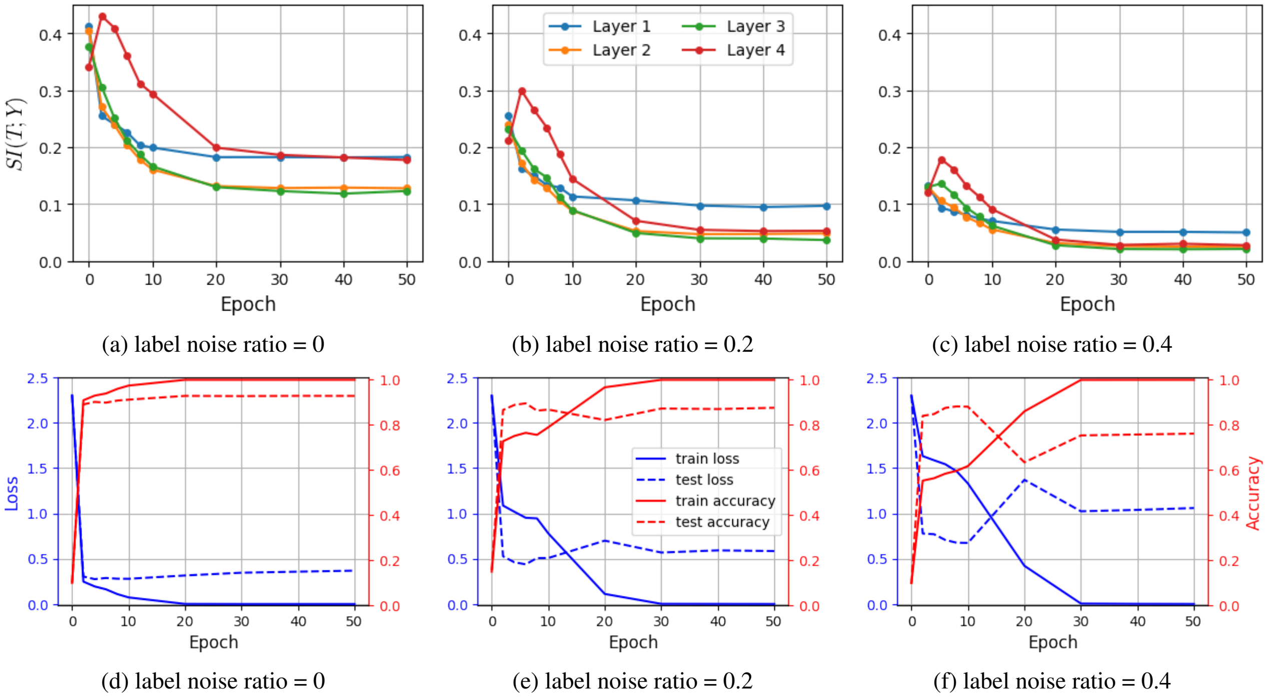
(c) label noise ratio = 0.4

(d) label noise ratio = 0

(e) label noise ratio = 0.2

(f) label noise ratio = 0.4

Figure 9: The $SI(T;Y)$ and model performance in 5-layer MLP trained with MNIST of different label noise ratio for 50 epochs.



(a) label noise ratio = 0

(b) label noise ratio = 0.2

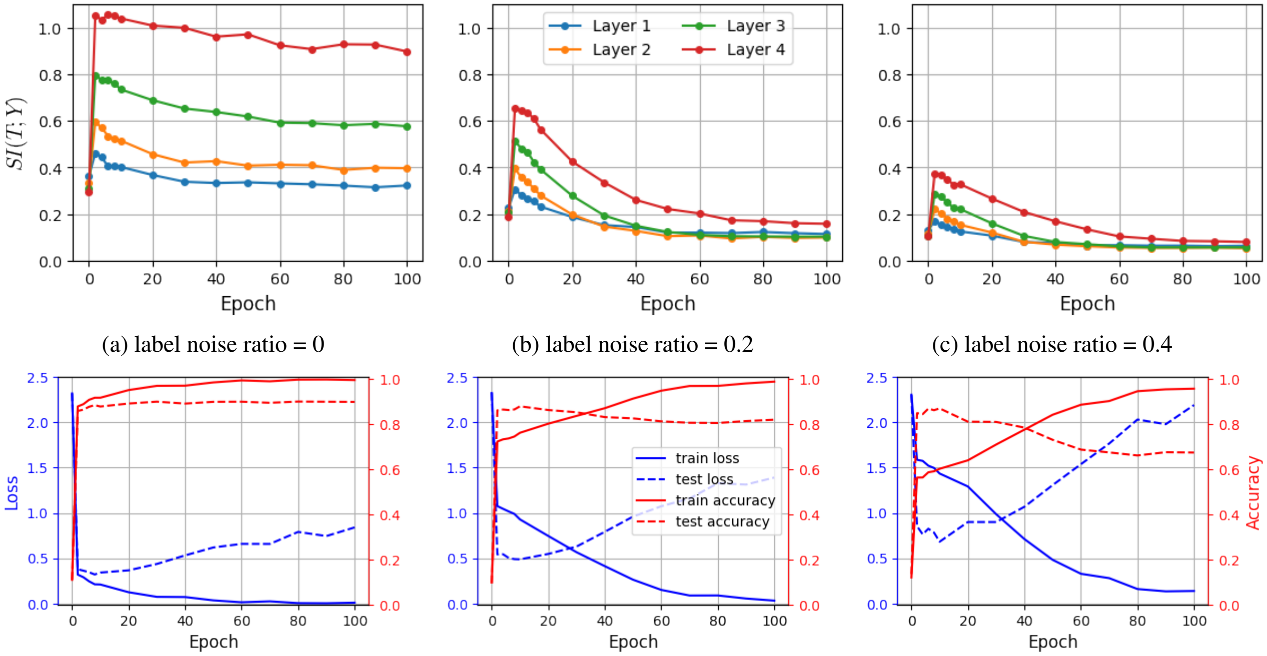
(c) label noise ratio = 0.4

(d) label noise ratio = 0

(e) label noise ratio = 0.2

(f) label noise ratio = 0.4

Figure 10: The $SI(T;Y)$ and model performance in 5-convolutional layer CNN trained with Fashion MNIST of different label noise ratio for 50 epochs.



(a) label noise ratio = 0

(b) label noise ratio = 0.2

(c) label noise ratio = 0.4

(d) label noise ratio = 0

(e) label noise ratio = 0.2

(f) label noise ratio = 0.4

Figure 11: The $SI(T;Y)$ and model performance in 5-layer MLP trained with Fashion MNIST of different label noise ratio for 100 epochs.

C.3 Generalization

Here, we provide the training details and additional results for Section 4.2 in the main paper.

Experiment Details: We consider four different types of networks: 5-layer MLP (architecture is shown in Table 1), 5-convolutional layer CNN (architecture is shown in Table 4), pre-trained VGG16 and pre-trained ResNet50 (both pre-trained architectures are shown in Table 5). For VGG16 and ResNet50, all the weights are trained during training. We consider 4 different datasets: MNIST, Fashion MNIST, CIFAR10 and CIFAR100. For MLP and CNN, the network is trained with batch size 32 using SGD optimizer with 0.01 learning rate and 0.9 momentum. For VGG16 and ResNet50, the network is trained with batch size 256 using SGD optimizer with 0.001 learning rate and 0.9 momentum. The SMI is computed with $m = 1000$ and $n = 10000$ on the training dataset. The stopping criterion is when the training accuracy does not improve in the next 20 epochs. For all the experiments here, the training accuracy is close to 100%. To obtain different generalization gap, we train the network with dropout of varying probability and different degrees of label noise. For the MLP setting, we consider dropout probability in the range $[0.1, 0.2, 0.3, 0.4, 0.5]$ and label noise ratio in the range $[0.005, 0.01, 0.05, 0.1, 0.2, 0.4, 0.6, 0.8, 1.0]$. For the CNN setting, we consider the dropout probability in the range $[0.1, 0.2, 0.3, 0.4, 0.5]$ and label noise ratio in the range $[0.2, 0.4, 0.6, 0.8, 1.0]$ as well as with and without batch normalization. For the VGG16 and ResNet50 cases, we consider dropout probability in the range $[0.1, 0.2, 0.3, 0.4]$ and label noise ratio in the range $[0.05, 0.1, 0.2, 0.4, 0.6, 0.8, 1.0]$. For MLP and CNN, the dropout is applied for each hidden layer while for VGG16 and ResNet50, the dropout is only applied at the last two fully-connected hidden layers.

Table 5: The architecture of pre-trained VGG16/ResNet50.

Layer Type	**Parameters**
VGG16/ResNet50 base network	Pre-trained on ImageNet
Fully-Connected	4096 units, ReLU
Fully-Connected	4096 units, ReLU
Fully-Connected	10 or 100 units, Linear