Estimating Information Flow in Deep Neural Networks

Ziv Goldfeld [1 2] **Ewout van den Berg** [2 3] **Kristjan Greenewald** [2 3] **Igor Melnyk** [2 3] **Nam Nguyen** [2 3]
Brian Kingsbury [2 3] **Yury Polyanskiy** [1 2]

Abstract

We study the estimation of the mutual information $I(X;T_\ell)$ between the input X to a deep neural network (DNN) and the output vector T_ℓ of its ℓ^{th} hidden layer (an "internal representation"). Focusing on feedforward networks with fixed weights and noisy internal representations, we develop a rigorous framework for accurate estimation of $I(X;T_\ell)$. By relating $I(X;T_\ell)$ to information transmission over additive white Gaussian noise channels, we reveal that compression, i.e. reduction in $I(X;T_\ell)$ over the course of training, is driven by progressive geometric clustering of the representations of samples from the same class. Experimental results verify this connection. Finally, we shift focus to purely deterministic DNNs, where $I(X;T_\ell)$ is provably vacuous, and show that nevertheless, these models also cluster inputs belonging to the same class. The binning-based approximation of $I(X;T_\ell)$ employed in past works to measure compression is identified as a measure of clustering, thus clarifying that these experiments were in fact tracking the same clustering phenomenon. Leveraging the clustering perspective, we provide new evidence that compression and generalization may *not* be causally related and discuss potential future research ideas.

1. Introduction

Measuring the mutual information $I(X;T_\ell)$ between the input feature X to a deep neural network (DNN) and the output T_ℓ of its ℓ^{th} layer has long been a topic of research, with applications to unsupervised feature learning (Linsker, 1988; van den Oord et al., 2018; Hjelm et al., 2019) and deep learning analysis (Shwartz-Ziv & Tishby, 2017; Saxe et al., 2018; Achille & Soatto, 2018). Mutual information is an appealing measure due to its invariance to smooth, invertible transformations and the fact that it has meaningful units (bits or nats). However, these benefits come at a price: mutual information is often impossible to compute analytically, and its estimation from samples is inherently difficult (Paninski, 2003). A variety of approaches for entropy, and thereby mutual information, estimation have been developed over the years, including k-nearest neighbors (kNN) techniques (Kozachenko & Leonenko, 1987; Kraskov et al., 2004), kernel density estimation techniques (Kandasamy et al., 2015; Han et al., 2017) and trainable neural estimators (Belghazi et al., 2018). However, most previous information-theoretic studies of deep learning (Shwartz-Ziv & Tishby, 2017; Saxe et al., 2018) approximate the mutual information by discretizing neurons' outputs, an operation called 'binning'.

The binning-based approach is attractive because of its computational efficiency when the number of bins is not too large; however, even mildly coarse discretizations can yield inaccurate estimates.[1] This fact is illustrated by the empirical mutual information plots from (Shwartz-Ziv & Tishby, 2017; Saxe et al., 2018) where the *compression* phenomenon of the Information Bottleneck theory, i.e., a long-term decrease of $I(X;T_\ell)$ during DNN training, was studied. Both works plotted the binned mutual information $I\big(X;\mathsf{Bin}(T_\ell)\big)$ for deterministic DNNs (namely, networks that deterministically map the input to the hidden representation) with $\mathsf{Bin}(T_\ell)$ being a per-neuron discretization of T_ℓ. But, in deterministic DNNs with strictly monotone nonlinearities (e.g., tanh or sigmoid) the true mutual information $I(X;T_\ell)$ is provably either infinite (continuous X) or a constant (discrete X). Therefore, the fluctuations of $I\big(X;\mathsf{Bin}(T_\ell)\big)$ observed during DNN training by (Shwartz-Ziv & Tishby, 2017; Saxe et al., 2018) must be due to estimation errors rather than changes in mutual information. Indeed, Fig. 1 illustrates how larger bin sizes can easily cause errors in $I\big(X;\mathsf{Bin}(T_\ell)\big)$ trajectories.

The degeneracy of $I(X;T_\ell)$ in deterministic DNNs is a consequence of T_ℓ being a deterministic function of X. If the

[*]Equal contribution [1]Massachusetts Institute of Technology [2]MIT-IBM Watson AI Lab [3]IBM Research. Correspondence to: Ziv Goldfeld <zivg@mit.edu>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

[1] The binning-based proxy approaches the true value when the bin sizes are shrunk to zero by definition (Cover & Thomas, 2006).

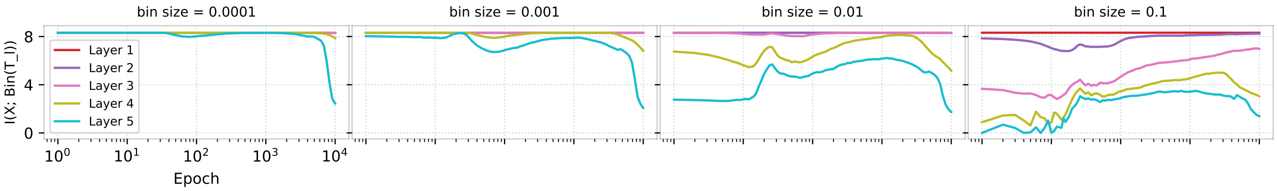

Figure 1. $I\big(X;\mathsf{Bin}(T_\ell)\big)$ vs. epochs for different bin sizes and the model in (Shwartz-Ziv & Tishby, 2017), where X is uniformly distributed over a 2^{12}-sized empirical dataset. The curves converge to $H(X) = \ln(2^{12}) \approx 8.3$ for small bins.

DNN has continuous nonlinearities and P_X is continuous, then so is T_ℓ, and thus $I(X;T_\ell) = \infty$ (cf. Theorem 2.4 of Polyanskiy & Wu (2012–2017)). When P_X is discrete (e.g., when the features are discrete or if X adheres to an empirical distribution over the dataset), the mutual information equals the entropy $H(X)$, a constant that is independent of the DNN parameters. This follows because the mapping from a discrete X to the support of T_ℓ is injective for strictly monotone nonlinearities for any but a measure-zero set of weights. Both continuous and discrete degeneracies were previously observed (Amjad & Geiger, 2018; Kolchinsky et al., 2019). These are a consequence of the fact that deterministic DNNs can encode information about X in arbitrarily fine variations of T_ℓ essentially without loss, even if deeper layers have fewer neurons.

That said, the estimate $I\big(X;\mathsf{Bin}(T_\ell)\big)$ is system dependent and its compression observed in past works seems meaningful. What mechanism drives this compression? To answer this question, we develop a rigorous framework for tracking the flow of information in DNNs. To ensure $I(X;T_\ell)$ is useful for studying the learned representations, the map $X \mapsto T_\ell$ must be a stochastic parameterized channel whose parameters are the DNN's weights and biases. To obtain pertinent insights into practical systems, we impose the following requirements on the framework. (R1) The stochasticity should be intrinsic to the operation of the DNN, so that the characteristics of mutual information measures are related to the learned internal representations. (R2) The stochasticity should relate the mutual information to the deterministic binned version $I\big(X;\mathsf{Bin}(T_\ell)\big)$, since this is the object whose compression was observed; this requires the injected noise to be isotropic over the domain of T_ℓ analogously to the per-neuron binning operation. (R3) Most importantly, the network trained under this stochastic model should be closely related to those trained in practice.

In Section 2 we propose a stochastic DNN framework in which independently and identically distributed (i.i.d.) Gaussian noise is added to the output of each of the DNN's hidden layer neurons. This makes the map $X \mapsto T_\ell$ stochastic, ensures the data processing inequality is satisfied, and makes $I(X;T_\ell)$ reflect the DNN's true operating conditions, per (R1). Since the noise is centered and isotropic, (R2) holds. As for (R3), experiments show that the DNN's learned representations and performance are not meaningfully affected by the addition of noise, for variances β^2 not too large.

Under the stochastic model, $I(X;T_\ell)$ has no exact analytic expression and is intractable to evaluate numerically. In Section 3 we therefore construct a provably accurate estimator for it. The estimator employs a sampling technique that decomposes the estimation problem into several instances of the differential entropy estimation setup studied in (Goldfeld et al., 2019). Leveraging the risk bounds derived therein, we prove that for any dimension of the hidden layer, the risk of our mutual information estimator converges at the parametric rate of estimation (see Section 3). We then introduce a method for efficient implementation of our mutual information estimator and derive theoretical guarantees on its accuracy. An implementation of the estimation toolkit is available at `http://anonymized`.

Having a tool for accurately tracking $I(X;T_\ell)$ over the course of stochastic DNN training, we focus on the geometric phenomenon that drives its fluctuations. We relate $I(X;T_\ell)$ to the well-understood notion of data transmission over additive white Gaussian noise (AWGN) channels. Namely, $I(X;T_\ell)$ is the aggregate information transmitted over the channel $P_{T_\ell|X}$ with input X drawn from a constellation defined by the data samples and the noisy DNN parameters. As training progresses, the representations of inputs from the same class tend to cluster together and become increasingly indistinguishable at the channel's output, thereby decreasing $I(X;T_\ell)$. Furthermore, these clusters tighten as one moves into deeper layers, providing evidence that the DNN's layered structure progressively improves the representation of X to increase its relevance for Y.

In Section 5.1 we experimentally demonstrate that, in some cases, $I(X;T_\ell)$ exhibits compression during training of noisy DNNs. It is further shown that regardless of whether $I(X;T_\ell)$ compresses or not, its fluctuations always track the degree of clustering in the internal representation space. Finally, in Section 5.2, we examine clustering in a deterministic DNN. Several methods for measuring clustering (valid for both noisy and deterministic systems) are identified and used to show that clusters of inputs in learned representations form during deterministic DNN training as well. We complete the circle back to $I\big(X;\mathsf{Bin}(T_\ell)\big)$ by demonstrating that this quantity measures clustering. This explains what previous works were actually observing in those determinis-

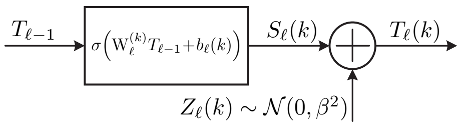
$T_{\ell-1}$
$\sigma\Big(\mathrm{W}_\ell^{(k)}T_{\ell-1}+b_\ell(k)\Big)$
$S_\ell(k)$
$T_\ell(k)$
$Z_\ell(k)\sim\mathcal{N}(0,\beta^2)$

Figure 2. kth noisy neuron in layer ℓ: $\mathrm{W}_\ell^{(k)}$ and $b_\ell(k)$ are the kth row/entry of the weight matrix and the bias, respectively.

tic systems: not (the theoretically impossible) compression of mutual information, but increased clustering of hidden representations. Leveraging the clustering perspective we then provide new evidence that compression of $I(X;T_\ell)$ and generalization may *not* be causally related. The geometric clustering of internal representations is thus the fundamental phenomenon of interest, and we aim to test its connection to generalization performance, theoretically and experimentally, in future work.

2. Preliminary Definitions

Noisy DNNs: For integers $k\leq\ell$, let $[k\!:\!\ell]\triangleq\big\{i\in\mathbb{Z}\big|k\leq i\leq\ell\big\}$ and use $[\ell]$ when $k=1$. Consider a noisy DNN with $L+1$ layers $\{T_\ell\}_{\ell\in[0:L]}$, with input $T_0=X$ and output T_L. The ℓth hidden layer, $\ell\in[L-1]$, is described by $T_\ell=f_\ell(T_{\ell-1})+Z_\ell$, where $f_\ell:\mathbb{R}^{d_{\ell-1}}\to\mathbb{R}^{d_\ell}$ is a deterministic function of the previous layer and $Z_\ell\sim\mathcal{N}\big(0,\beta^2\mathrm{I}_{d_\ell}\big)$; no noise is injected to the output, i.e., $T_L=f_L(T_{L-1})$. We set $S_\ell\triangleq f_\ell(T_{\ell-1})$ and use φ_β for the probability density function (PDF) of Z_ℓ. The functions $\{f_\ell\}_{\ell\in[L]}$ can represent any type of layer (fully connected, convolutional, max-pooling, etc.). Fig. 2 shows a neuron in a noisy DNN.

To explore the relation between noisy and deterministic DNNs under conditions representative of current machine learning practices, we trained four-layer convolutional neural networks (CNNs) on MNIST (LeCun et al., 1999). The CNNs used different internal noise levels (including $\beta=0$) and one used dropout instead of additive noise. We measured their performance on the validation set and characterized the cosine similarities between their internal representations (see Supplement 8.3 for full details of architecture and training). The results in Table 1 show small amounts of internal noise ($\beta\leq 0.1$) have a minimal impact on classification performance, while dropout strongly improves it. The histograms in Fig. 3 show that the noisy (for small β) and dropout models learn internal representations similar to those learned by the deterministic model. In this high-dimensional space, unrelated representations would create cosine similarity histograms with zero mean and standard deviation between 0.02–0.3, so the observed values are quite large. As expected, dissimilarity increases as β increases, and similarity is lower for the internal layers (2 and 3).

Table 1. MNIST validation errors for different models: mean ± standard deviation over eight initial random seeds.

Model	**# Errors**
Deterministic	50 ± 4.6
Noisy ($\beta=0.05$)	50 ± 5.0
Noisy ($\beta=0.1$)	51 ± 6.9
Noisy ($\beta=0.2$)	86 ± 9.8
Noisy ($\beta=0.5$)	2200 ± 520
Dropout ($p=0.2$)	39 ± 3.9

Mutual Information: Noisy DNNs induce a stochastic map from X to the rest of the network, described by the conditional distribution $P_{T_1,\ldots,T_L|X}$. The corresponding PDF[2] is $p_{T_1,\ldots,T_L|X=x}$. Assuming $X\sim P_X$, the system is described by the joint distribution $P_{X,T_1,\ldots,T_L}\triangleq P_XP_{T_1,\ldots,T_L|X}$, under which $X-T_1-\ldots-T_{L-1}-T_L$ forms a Markov chain. For each $\ell\in[L-1]$, we study the mutual information $I(X;T_\ell)\triangleq h(T_\ell)-\int\mathsf{d}P_X(x)h(T_\ell|X=x)$. The composition of Gaussian noises and nonlinearities renders the stochastic map $X\mapsto T_\ell$ too complicated to analytically compute $I(X;T_\ell)$. Therefore, we focus on estimating $I(X;T_\ell)$ from samples.

3. Mutual Information Estimation

We design a provably accurate estimator of $I(X;T_\ell)$ inspired by our recent work on differential entropy estimation (Goldfeld et al., 2019). Given a feature dataset $\mathcal{X}=\{x_i\}_{i\in[n]}$ i.i.d. according to P_X, our $I(X;T_\ell)$ estimator is constructed from estimators of $h(T_\ell)$ and $h(p_{T_\ell|X=x})$, $\forall x\in\mathcal{X}$. We propose a sampling method that reduces the estimation of each entropy to the framework from (Goldfeld et al., 2019). Using entropy estimation risk bounds derived therein we control the error of our *sample propagation* (SP) estimator $\hat{I}_{\mathsf{SP}}$ of $I(X;T_\ell)$.

Each differential entropy is estimated and computed via a two-step process. First, we approximate each true entropy by the differential entropy of a *known* Gaussian mixture (defined only through the available resources: the obtained samples and the noise parameter). This estimate converges to the true value at the *parametric* estimation rate, uniformly in the dimension. However, since the Gaussian mixture entropy has no closed-form expression, in the second (computational) step we develop a Monte Carlo integration (MCI) method to numerically evaluate it. Mean squared error (MSE) bounds on the MCI computed value are established.

3.1. Sampling Procedure and the Estimator

Unconditional Entropy: Since $T_\ell=S_\ell+Z_\ell$, where S_ℓ and Z_ℓ are independent, we have $p_{T_\ell}=p_{S_\ell}*\varphi_\beta$. To estimate $h(p_{T_\ell})$ feed each $x\in\mathcal{X}$ into the DNN and collect

[2] $P_{T_1,\ldots,T_L|X=x}$ is absolutely continuous with respect to (w.r.t.) the Lebesgue measure for all $x\in\mathcal{X}$.

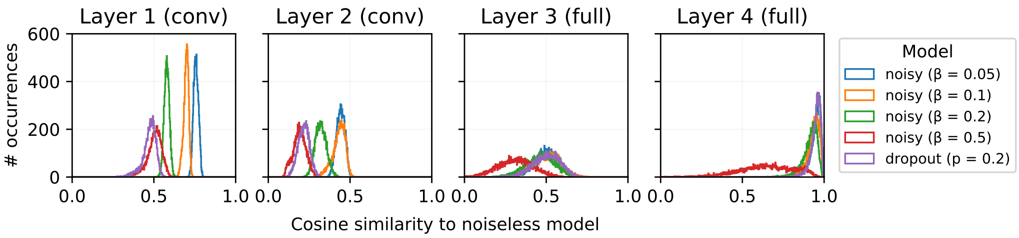

Figure 3. Cosine similarity histograms between internal representations of deterministic, noisy, and dropout MNIST CNNs.

the output it produces at the $(\ell-1)$-th layer. The function f_ℓ is then applied on each collected output to obtain $\mathcal{S}_\ell \triangleq \{s_{\ell,i}\}_{i\in[n]}$, which is a set of n i.i.d. samples from p_{S_ℓ}. Denoting by $\hat{p}_\mathcal{A}$ the empirical probability mass function of a set $\mathcal{A} = \{a_i\}_{i\in[n]}$, we estimate $h(p_{T_\ell})$ via $h(\hat{p}_{\mathcal{S}_\ell} * \varphi_\beta)$, which is the differential entropy of a Gaussian mixture with centers $\{s_{\ell,i}\}_{i\in[n]}$.

Conditional Entropies: Fix $x \in \mathcal{X}$ and consider estimating $h(p_{T_\ell|X=x})$, where $p_{T_\ell|X=x} = p_{S_\ell|X=x} * \varphi_\beta$. We sample from $p_{S_\ell|X=x}$ by feeding x into the DNN n_x times, collecting $T_{\ell-1}$ outputs that correspond to different noise realizations, and applying f_ℓ on each. The obtained samples $\mathcal{S}_\ell^{(x)} \triangleq \{s_{\ell,i}^{(x)}\}_{i\in[n_x]}$ are i.i.d. according to $p_{S_\ell|X=x}$. Each $h(p_{T_\ell|X=x})$ is estimated by $h\big(\hat{p}_{\mathcal{S}_\ell^{(x)}} * \varphi_\beta\big)$.[3]

Mutual Information Estimator: We estimate $I(X;T_\ell)$ by

$$\hat{I}_{\mathsf{SP}} \triangleq h(\hat{p}_{\mathcal{S}_\ell} * \varphi_\beta) - \frac{1}{n}\sum_{x\in\mathcal{X}} h\big(\hat{p}_{\mathcal{S}_\ell^{(x)}} * \varphi_\beta\big). \quad (1)$$

3.2. Theoretical Guarantees and Computation

This sampling procedure unifies the estimation of $h(p_{T_\ell})$ and $\big\{h(p_{T_\ell|X=x})\big\}_{x\in\mathcal{X}}$ into the problem of differential entropy estimation under Gaussian convolutions (Goldfeld et al., 2019): estimate $h(p_S * \varphi_\beta)$ based on i.i.d. samples $\mathcal{S}_n \triangleq \{S_i\}_{i\in[n]}$ from p_S and knowledge of φ_β. Our $\hat{I}_{\mathsf{SP}}$ is inspired by the differential entropy estimator proposed in (Goldfeld et al., 2019), which approximates $h(p_S * \varphi_\beta)$ by $h(\hat{p}_{\mathcal{S}_n} * \varphi_\beta)$. Before analyzing $\hat{I}_{\mathsf{SP}}$ performance, we note that its estimation is statistically difficult since any good estimator of $h(p_S * \varphi_\beta)$ using $\mathcal{S}_n$ and φ_β requires exponentially many samples in d (Theorem 1 of (Goldfeld et al., 2019)). Nonetheless, Theorem 2 therein shows that the absolute-error risk of $h(\hat{p}_{\mathcal{S}_n} * \varphi_\beta)$ converges at the best possible rate of $O\big(c^d/\sqrt{n}\big)$, for a constant c and all d. These results are restated and discussed in Supplement 9.

We now bound the estimation risk of $\hat{I}_{\mathsf{SP}}$ (the result is stated for bounded nonlinearities for simplicity of presentation; see Supplement 9 for corresponding ReLU results).

Theorem 1. *Fix $\ell \in [L-1]$ and assume $\|f_\ell\|_\infty \le 1$. For $\hat{I}_{\mathsf{SP}}$ from (1) with $n = n_x$, for all $x \in \mathcal{X}$, we have*

$$\sup_{P_X} \mathbb{E}\left|I(X;T_\ell) - \hat{I}_{\mathsf{SP}}\right| \le \frac{8c^{d_\ell} + d_\ell \log\left(1+\frac{1}{\sigma^2}\right)}{4\sqrt{n}}, \quad (2)$$

where c is a numerical constant explicitly given in the right-hand side (RHS) of (7) in Supplement 9.3.

Theorem 1 is proven in Supplement 10.1. Interestingly, the quantity $\frac{1}{\sigma^2}$ is the signal-to-noise ratio (SNR) between S and Z. The larger σ is, the easier estimation becomes, since the noise smooths out the complicated P_X distribution.

Evaluating the SP estimator requires computing differential entropies of (known) Gaussian mixture distributions (see (1)). Let $\hat{p}_{s^n} * \varphi_\beta$ be such a mixture and $G \sim \hat{p}_{s^n} * \varphi_\beta$. Noting that $h(\hat{p}_{s^n} * \varphi_\beta) = -\mathbb{E}\Big[\log\big((\hat{p}_{s^n} * \varphi_\beta)(G)\big)\Big]$, we numerically approximate the RHS via efficient MCI (Robert, 2004). Specifically, we generate n_{MC} i.i.d. samples from $\hat{p}_{s^n} * \varphi_\beta$ and approximate the expectation by an empirical average. This unbiased proxy achieves an MSE of $O\big((n\cdot n_{\mathsf{MC}})^{-1}\big)$ (Supplement 9), and thus only adds a negligible error to the estimation process.[4]

Choosing β and n: We describe practical guidelines for selecting β and n for estimating $I(X;T_\ell)$ in actual classifiers. Ideally, β is treated as a hyperparameter tuned to optimize the DNN's performance on held-out data, since internal noise serves as a regularizer similar to dropout. In practice, we sometimes back off from this optimal β to a higher value to ensure accurate estimation of mutual information ($\hat{I}_{\mathsf{SP}}$ requires more samples for smaller β values), depending on factors like the dimensionality of T_ℓ and the number of available data samples.

While n can be selected from the risk bound in (2), this value can be quite large since Theorem 1 is a worst-case result. Furthermore, generating the estimated $I(X;T_\ell)$ curves shown in Section 5 requires repeatedly[5] running the differential entropy estimator, which makes the n dictated by

[3] For $\ell = 1$, we have $h(T_1|X) = h(Z_1) = \frac{d_1}{2}\log(2\pi e\beta^2)$.

[4] There are other ways to numerically evaluate $h(\hat{p}_{s^n} * \varphi_\beta)$, e.g., the Gaussian mixture bounds from Kolchinsky & Tracey (2017); however, our proposed method is the fastest of which we are aware.

[5] Each $I(X;T_\ell)$, for a given set of DNN parameters, involves computing $n+1$ differential entropy estimates, and our experiments estimate the trajectory of $I(X;T_\ell)$ across training epochs.

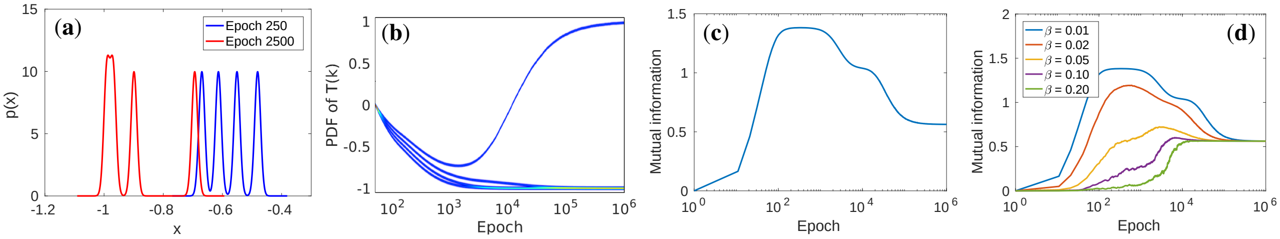

Figure 4. Single-layer tanh network: (a) the density $p_{T(k)}$ at epochs $k = 250, 2500$; (b) $p_{T(k)}$ and (c) $I\big(X;T(k)\big)$ as a function of k; and (d) mutual information as a function of k, for different β values..

Theorem 1 infeasible. To overcome this computational burden while adhering to the theory, we tested the value of n given by the theorem on a few points of each curve and reduced it until the overall computation cost became reasonable. To ensure estimation accuracy was not compromised we empirically tested that the estimate remained stable.

As a concrete example, for an error bound of 10% of Fig. 5 plot's vertical scale ($\approx$ 0.8 absolute error) Theorem 1 requires $n = 4 \cdot 10^9$ samples, which is beyond our computational budget. Performing the above procedure to lower n, we find good accuracy is achieved for $n = 4 \cdot 10^6$. Adding more samples beyond this value does not change the results.

4. Compression and Clustering

We use a minimal example to connect compression and clustering via an information-theoretic perspective. Consider one noisy neuron with a scalar input X. Let $T(k) = S(k) + Z = \sigma(w_k X + b_k) + Z$ be the neuron's output at epoch k, where σ is strictly monotone and $Z \sim \mathcal{N}(0, \beta^2)$. Invariance of $I\big(X;T(k)\big)$ to invertible operations implies $I\big(X;T(k)\big) = I\big(S(k);S(k)+Z\big)$. From an information-theoretic perspective, $I\big(S(k);S(k)+Z\big)$ is the aggregate number of nats transmitted over an AWGN channel with input constellation $\mathcal{S}_k \triangleq \big\{\sigma(w_k x + b_k) \mid x \in \mathcal{X}\big\}$. In other words, $I\big(S(k);S(k)+Z\big)$ measures how distinguishable the symbols of $\mathcal{S}_k$ are when composed with Gaussian noise (it roughly equals the log of the number of resolvable clusters under noise level β). Since the distribution of $T(k)$ is a Gaussian mixture with means $s \in \mathcal{S}_k$, the closer two constellation points s and s' are, the more the Gaussians centered on them overlap. Hence reducing point spacing in $\mathcal{S}_k$ (by changing w_k and b_k) directly reduces $I\big(X;T(k)\big)$.

Let $\sigma = \tanh$, $\beta = 0.01$ and $\mathcal{X} = \mathcal{X}_{-1} \cup \mathcal{X}_1$, with $\mathcal{X}_{-1} = \{-3,-1,1\}$ and $\mathcal{X}_1 = \{3\}$, labeled -1 and 1, respectively. We train the neuron using mean squared loss and gradient descent with learning rate 0.01 to best illustrate $I\big(X;T(k)\big)$ trends. The Gaussian mixture $p_{T(k)}$ is plotted across epochs k in Fig. 4(a)-(b). The learned bias is approximately $-2.3w$, ensuring that the tanh transition region correctly divides the two classes. Initially $w = 0$, so all four Gaussians in $p_{T(0)}$ are superimposed. As k increases, the Gaussians initially diverge, with the three from $\mathcal{X}_{-1}$ eventually re-converging as they each meet the tanh boundary. This is reflected in the mutual information trend in Fig. 4(c), with the dips in $I\big(X;T(k)\big)$ around $k = 10^3$ and $k = 10^4$ corresponding to the second and third Gaussians respectively merging into the first. Thus, there is a direct connection between clustering and compression. Fig. 4(d) shows $I\big(X;T(k)\big)$ for different noise levels β as a function of k. For small β (as above) the $\mathcal{X}_{-1}$ Gaussians are distinct and merge in two stages as w grows. For larger β, however, the $\mathcal{X}_{-1}$ Gaussians are indistinguishable for any w, making $I\big(X;T(k)\big)$ only increase as the two classes gradually separate. Similar behavior and trends are observed for a two-neuron leaky-ReLU network in Supplement 7.

5. Empirical Results

We show that the observations from Section 4 hold for two larger networks. The experiments demonstrate that $I(X;T_\ell)$ compression in noisy DNNs is driven by clustering of internal representations, and that deterministic DNNs cluster samples as well. The considered DNNs are (1) the fully connected network (FCN) from (Shwartz-Ziv & Tishby, 2017; Saxe et al., 2018), dubbed the *SZT model*, and (2) a convolutional network for MNIST classification, called *MNIST CNN*. We present selected results; see supplement for additional experiments.

5.1. Noisy SZT Model

Consider the data and model of (Shwartz-Ziv & Tishby, 2017) for binary classification of 12-dimensional inputs using a fully connected 12–10–7–5–4–3–2 architecture. The FCN was tested with tanh and ReLU nonlinearities as well as a linear model. Fig. 5(a) presents results for the tanh SZT model with $\beta = 0.005$ (test classification accuracy 97%), showing the relation across training epochs between the estimated $I(X;T_\ell)$, train/test losses and the evolving internal representations. The rise and fall of $I(X;T_\ell)$ corresponds to how spread out or clustered the representations in each layer are. For example, $I(X;T_5)$ grows until epoch 28, when the Gaussians start to move away from each other along a curve (see scatter plots on the right). Around epoch

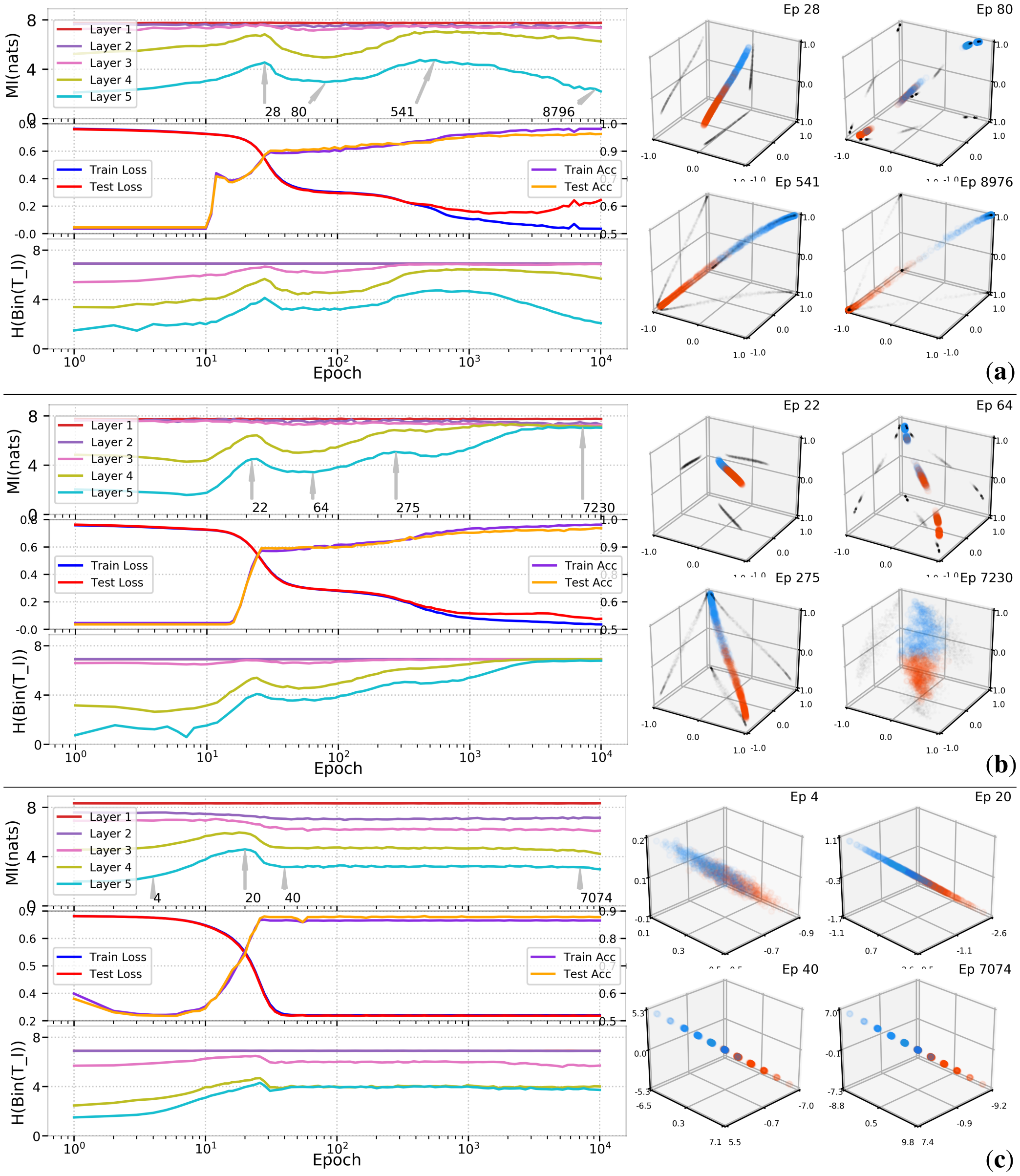

Figure 5. (a) Evolution of $I(X;T_\ell)$ and training/test losses across training epochs for the SZT model with $\beta = 0.005$ and tanh nonlinearities. The scatter plots show the values of Layer 5 ($d_5 = 3$) at the arrow-marked epochs on the mutual information plot. The bottom plot shows $H\big(\mathsf{Bin}(T_\ell)\big)$ across epochs for bin size $B = 10\beta$. (b) Same setup as in (a) but with regularization that encourages orthonormal weight matrices. (c) SZT model with $\beta = 0.01$ and *linear* activations.

80 they start clustering and $I(X;T_5)$ drops. As training progresses, the saturating tanh units push the Gaussians to two furthest corners of the cube, reducing $I(X;T_5)$ even more.

To confirm that clustering (via saturation) was central to the compression observed in Fig. 5(a), we also trained the model using the regularization from (Cisse et al., 2017) (test classification accuracy 98%), which encourages orthonormal weight matrices. The results are shown in Fig. 5(b). Apart from minor initial fluctuations, compression is completely gone. The scatter plots show that the vast majority of neurons do not saturate and no clustering is observed at the later stages of training. Saturation is not the only mechanism that can cause clustering and consequently reduce $I(X;T_\ell)$. For example, in Fig. 5(c) we illustrate the clustering behavior in a linear SZT model (test classification accuracy 89%). As seen from the scatter plots, due to the formation of several clusters and projection to a lower dimensional space, $I(X;T_\ell)$ drops even without the nonlinearities.

The results in Fig. 5(a) and (b) also show that the relation-

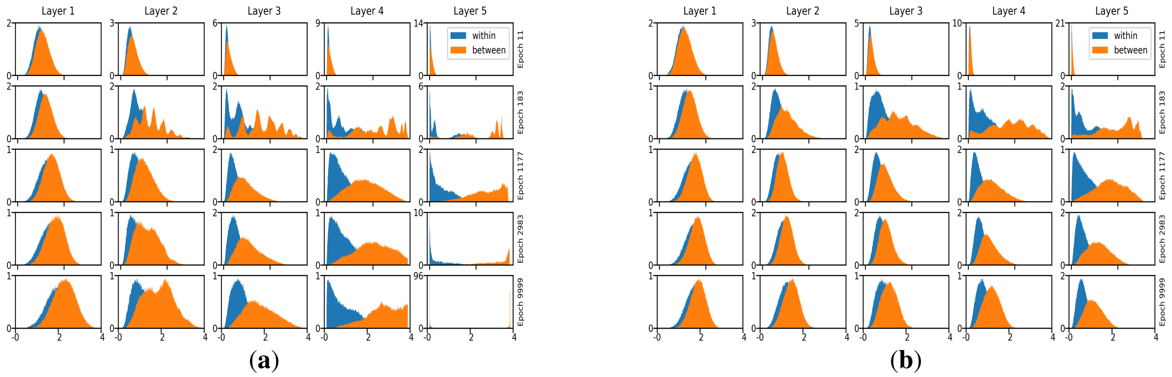

Figure 6. (a) Histogram of within- and between-class pairwise distances for SZT model with tanh non-linearities and additive noise $\beta = 0.005$. (b) Same as (a) but training with weight normalization.

ship between compression and generalization performance is not a simple one. In Fig. 5(a), the test loss begins to increase at roughly epoch 3200 and continues to grow until training ends. At the same time compression occurs in layers 4 and 5. In contrast, the test loss in Fig. 5(b) does not increase, and compression does not occur. We believe this subject deserves further examination in future work.

To provide another perspective on clustering that is sensitive to class membership, we compute histograms of pairwise distances between representations of samples, distinguishing within-class distances from between-class distances. Fig. 6 shows histograms for the SZT models from Figs. 5(a) and (b). As training progresses, the formation of clusters is clearly seen (layer 3 and beyond) for the unnormalized SZT model in Fig. 5(a). In the normalized model (Fig. 5(b)), however, no tight clustering is apparent, supporting the connection between clustering and compression.

Having identified clustering as the source of compression, we focus on it as the point of interest. To measure clustering we consider the discrete entropy $H\big(\mathsf{Bin}(T_\ell)\big)$, with the number of equal-sized bins, B, as a tuning parameter. Note that $\mathsf{Bin}(T_\ell)$ partitions the dynamic range (e.g., $[-1,1]^{d_\ell}$ for a tanh layer) into B^{d_ℓ} cells or bins. When hidden representations are spread out, many bins are non-empty, each holding a positive probability mass. Conversely, for clustered representations, the distribution is concentrated on a few bins, each with relatively high probability. Recalling that the uniform distribution maximizes discrete entropy, we see why reduction in $H\big(\mathsf{Bin}(T_\ell)\big)$ measures clustering.

To illustrate, the bottom plots in Figs. 5(a), (b) and (c) show $H\big(\mathsf{Bin}(T_\ell)\big)$ for each SZT model using $B = 10\beta$. Its precise values differ from those of $I(X;T_\ell)$, suggesting $H\big(\mathsf{Bin}(T_\ell)\big)$ is formally not an estimator of the mutual information. Nonetheless, a clear correspondence between the trends of $H\big(\mathsf{Bin}(T_\ell)\big)$ and $I(X;T_\ell)$ is seen, indicating that $H\big(\mathsf{Bin}(T_\ell)\big)$ measures clustering in a manner similar to $I(X;T_\ell)$. This is particularly important when moving back to *deterministic DNNs*, where $I(X;T_\ell)$ is no longer informative (being independent of the system parameters). Fig. 1 shows $H\big(\mathsf{Bin}(T_\ell)\big)$ for the deterministic SZT model ($\beta = 0$). The bin size is a free parameter, and depending on its value, $H\big(\mathsf{Bin}(T_\ell)\big)$ reveals different clustering granularities. Moreover, since in deterministic networks $T_\ell = f_\ell(X)$, for a deterministic map f_ℓ we have $I\big(X;\mathsf{Bin}(T_\ell)\big) = H\big(\mathsf{Bin}(T_\ell)\big)$. Thus, the plots from (Shwartz-Ziv & Tishby, 2017), (Saxe et al., 2018), and our Fig. 1 all show that $H\big(\mathsf{Bin}(T_\ell)\big)$ evolution during training of deterministic DNNs *tracks the degree of clustering in the internal representation spaces*, rather than the theoretically impossible compression of $I(X;T_\ell)$.

5.2. Deterministic MNIST CNN

We also examine a deterministic model that is more representative of current machine learning practice: the MNIST CNN trained with dropout from Section 2. Fig. 7 portrays the near-injective behavior of this model. Even when only two bins are used to compute $H\big(\mathsf{Bin}(T_\ell)\big)$, it takes values that are approximately $\ln(10000) = 9.210$, for all layers and training epochs, even though the two convolutional layers use max-pooling. While Fig. 7 does not show compression at the level of entire layers, computing $H\big(\mathsf{Bin}(T_\ell(k))\big)$ for individual units k in layer 3 reveals a gradual decrease over epochs 1–128. To quantify this trend, we computed linear regressions predicting $H\big(\mathsf{Bin}(T_\ell(k))\big)$ from the epoch index, for all units k in layer 3, and determined the mean and standard deviation of the slope of the linear predictions. If most slopes are negative, then compression occurs during training at the level of individual units. For a range of bin sizes from 10^{-4}–10^{-1} the least negative mean slope was -0.002 nats/epoch with a maximum standard deviation of 0.001, showing that most units undergo compression.

In Fig. 8 we show histograms of pairwise distances between MNIST validation set samples in the input (pixel) space and in the four layers of the CNN. The histograms were

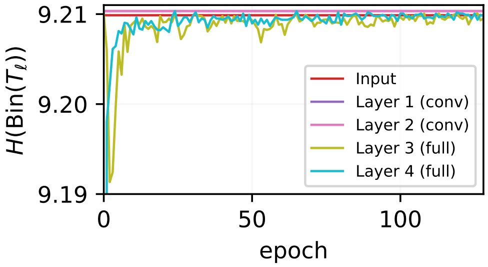

Figure 7. $H\big(\mathsf{Bin}(T_\ell)\big)$ for the MNIST CNN, computed using two bins: $[-1, 0]$ and $(0, 1]$. The tiny range of the y axis shows the near injectivity of the model.

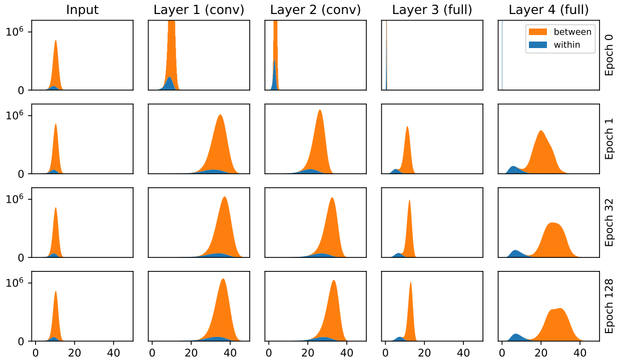

Figure 8. Histograms of within-class and between-class pairwise distances from the MNIST CNN.

computed for epochs 0, 1, 32, and 128, where epoch 0 is the initial random weights and epoch 128 is the final weights. The histogram for the input shows that the mode of within-class pairwise distances is lower than the mode of between-class pairwise distances, but that there is substantial overlap. Layers 1 and 2, which are convolutional and therefore do not contain any units that receive the full input, do little to reduce this overlap, suggesting that the features learned in these layers are somewhat generic. In contrast, even after one epoch of training, layers 3 and 4, which are fully connected, separate the distribution of within-class distances from the distribution of between-class distances.

5.3. Summary of Experiments

To summarize, we made the following observations in our experiments. (1) Compression can occur in noisy networks, e.g., the noisy SZT model inspired by the deterministic network from (Shwartz-Ziv & Tishby, 2017), for which compression was first observed (upper left plot in Fig. 5(a)). (2) Compression is caused by clustering of internal representations, with clusters comprising mostly samples from the same class, as seen in the scatter plots on the right sides of Figs. 5(a) and (c), and the distributions of pairwise distances in Figs. 6 and 8. (3) Regularization that limits the network's ability to saturate hidden units can suppress the formation of tight clusters in the internal representation spaces and, consequently, eliminate compression (Fig. 5(b)). Observing that the regularized network from Fig. 5(b) (where no compression occurs) generalizes better than the unregularized version in Fig. 5(a), provides further evidence that $I(X;T_\ell)$ *compression and generalization may not be causally related*. This relation warrants further investigation. (4) Fig. 5 demonstrated that $I(X;T_\ell)$ and $H\big(\mathsf{Bin}(T_\ell)\big)$ are highly correlated, establishing the latter as another measure for clustering (applicable both in noisy and deterministic DNNs). (5) Clustering of internal representations can also be observed in a somewhat larger, convolutional network trained on MNIST. While Fig. 7 shows that due to the high dimensionality, $H\big(\mathsf{Bin}(T_\ell)\big)$ fails to track compression in the larger CNN, strong evidence for clustering is found via estimates done at the individual units level and the analysis of pairwise distances between samples shown in Fig. 8.

6. Conclusions and Future Work

This work studied the mutual information $I(X;T_\ell)$ in a DNN. Through on our rigorous approach, we reexamined the 'compression' aspect of the Information Bottleneck theory (Shwartz-Ziv & Tishby, 2017), noting that fluctuations of $I(X;T_\ell)$ in deterministic networks with strictly monotone nonlinearities are theoretically impossible. Aiming to test for $I(X;T_\ell)$ compression in a sound manner and discover its source, we: (1) created a rigorous framework for studying and accurately estimating information-theoretic quantities in DNNs whose weights are fixed; (2) identified clustering of the learned representations as the geometric phenomenon underlying compression; and (3) demonstrated that the compression-related experiments on deterministic networks in prior works were in fact measuring this clustering through the lens of the binned mutual information.

We thus identify clustering as the common phenomenon of interest, which happens in both deterministic and noisy networks. In contrast, the mutual information $I(X;T_\ell)$ is meaningful only if a mechanism for shedding information (e.g., noise) exists in the network, and even then, it merely tracks the same clustering effect. Although binning-based measures do not accurately estimate mutual information, they are simple to compute (as opposed to the exponential in dimension burden of mutual information estimation) and are useful for tracking changes in clustering. We believe that further study of this geometric phenomenon is warranted to gain more insight into the learned representations and potentially establish connections with generalization. To this end we are currently developing efficient methods to track clustering in high-dimensional spaces. Such methods also open the door to algorithmic advances in deep learning. In fact, inspired by the clustering phenomenon, we are working on a new regularization procedure for DNN training that encourages intermediate layers of the network to increase a clustering-based analog of $I(Y;T_\ell)$. This makes the regularized layer learn well-separated representations of the data (with possibly nonlinear decision boundaries) and enhances the training process, according to our initial experiments. We aim to further develop this into a practical algorithm that accelerates DNN training in the near future.

References

Achille, A. and Soatto, S. On the emergence of invariance and disentangling in deep representations. *Journal of Machine Learning Research*, 19:1–34, 2018.

Amjad, R. A. and Geiger, B. C. Learning representations for neural network-based classification using the information bottleneck principle. arXiv:1802.09766 [cs.LG], 2018.

Belghazi, M. I., Baratin, A., Rajeswar, S., Ozair, S., Bengio, Y., Courville, A., and Hjelm, R. D. Mutual information neural estimation. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2018.

Cisse, M., Bojanowski, P., Grave, E., Dauphin, Y., and Usunier, N. Parseval networks: Improving robustness to adversarial examples. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2017.

Cover, T. M. and Thomas, J. A. *Elements of Information Theory*. John Wiley & Sons, 2nd edition, 2006.

Goldfeld, Z., Greenewald, K., Weed, J., and Polyanskiy, Y. Optimality of the plug-in estimator for differential entropy estimation under Gaussian convolutions. Paris, France, July 2019.

Han, Y., Jiao, J., Weissman, T., and Wu, Y. Optimal rates of entropy estimation over Lipschitz balls. arXiv:1711.02141 [math.ST], 2017.

Hjelm, R. D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., and Bengio, Y. Learning deep representations by mutual information estimation and maximization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019. To appear.

Kandasamy, K., Krishnamurthy, A., Poczos, B., Wasserman, L., and Robins, J. M. Nonparametric von Mises estimators for entropies, divergences and mutual informations. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 397–405, 2015.

Kolchinsky, A. and Tracey, B. D. Estimating mixture entropy with pairwise distances. *Entropy*, 19(7):361, July 2017.

Kolchinsky, A., Tracey, B. D., and Van Kuyk, S. Caveats for information bottleneck in deterministic scenarios. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019. To appear.

Kozachenko, L. F. and Leonenko, N. N. Sample estimate of the entropy of a random vector. *Problemy Peredachi Informatsii*, 23(2):9–16, 1987.

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

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1999.

Linsker, R. Self-organization in a perceptual network. *Computer*, 21(3):105–117, March 1988.

Paninski, L. Estimation of entropy and mutual information. *Neural Computation*, 15:1191–1253, June 2003.

Polyanskiy, Y. and Wu, Y. Lecture notes on information theory. 2012–2017. URL `http://people.lids.mit.edu/yp/homepage/data/itlectures_v5.pdf`.

Robert, C. P. *Monte Carlo Methods*. Wiley Online Library, 2004.

Saxe, A. M., Bansal, Y., Dapello, J., Advani, M., Kolchinsky, A., Tracey, B. D., and Cox, D. D. On the information bottleneck theory of deep learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

Shwartz-Ziv, R. and Tishby, N. Opening the black box of deep neural networks via information. arXiv:1703.00810 [cs.LG], 2017.

van den Oord, A., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. arXiv:1807.03748 [cs.LG], 2018.

Supplement to Estimating Information Flow in Deep Neural Networks

Ziv Goldfeld [1 2] **Ewout van den Berg** [3 2] **Kristjan Greenewald** [3 2] **Igor Melnyk** [3 2] **Nam Nguyen** [3 2]
Brian Kingsbury [3 2] **Yury Polyanskiy** [1 2]

NOTE: All references from the main text to this supplementary document can be replaced at publication time by references to a preprint on arXiv, per ICML guidelines.

7. Two-Neuron Leaky-ReLU Network Example

To expand upon Section **??**, we provide here a second example to illustrate the relation between clustering and compression of mutual information. In particular, this example also shows that as opposed to the claim from (Saxe et al., 2018), non-saturating nonlinearities can achieve compression. Consider the non-saturating Leaky-ReLU nonlinearity $R(x) \triangleq \max(x, x/10)$. Let $\mathcal{X} = \mathcal{X}_0 \cup \mathcal{X}_{1/4}$, with $\mathcal{X}_0 = \{1, 2, 3, 4\}$ and $\mathcal{X}_{1/4} = \{5, 6, 7, 8\}$, and labels 0 and $1/4$, respectively. We train the network via GD with learning rate 0.001 and mean squared loss. Initialization (shown in Fig. 9(a)) was chosen to best illustrate the connection between the Gaussians' motion and mutual information. The network converges to a solution where $w_1 < 0$ and b_1 is such that the elements in $\mathcal{X}_{1/4}$ cluster. The output of the first layer is then negated using $w_2 < 0$ and the bias ensures that the elements in $\mathcal{X}_0$ are clustered without spreading out the elements in $\mathcal{X}_{1/4}$. Figs. 9(b) show the Gaussian motion at the output of the first layer and the resulting clustering. For the second layer (Fig. 9(c)), the clustered bundle $\mathcal{X}_{1/4}$ is gradually raised by growing b_2, such that its elements successively split as they cross the origin; further tightening of the bundle is due to shrinking $|w_2|$. Fig. 9(d) shows the mutual information of the first (blue) and second (red) layers. The merging of the elements in $\mathcal{X}_{1/4}$ after their initial divergence is clearly reflected in the mutual information. Likewise, the spreading of the bundle, and successive splitting and coalescing of the elements in $\mathcal{X}_{1/4}$ are visible in the spikes in the red mutual information curve. The figure also shows how the bounds on $I\big(X; T(k)\big)$ precisely track its evolution.

8. Experimental Details

8.1. SZT Model

In this section we provide additional experimental details and results for the SZT model discussed in Section **??** of the main paper.

To regularize the network weights, we followed (Cisse et al., 2017) and adopted their approach for enforcing an orthonormality constraint. Specifically, we first update the weights $\{\mathrm{W}_\ell\}_{\ell \in [L]}$ using the standard gradient descent step, and then perform a secondary update to set

$$\mathrm{W}_\ell \leftarrow \mathrm{W}_\ell - \alpha \left(\mathrm{W}_\ell \mathrm{W}_\ell^T - \mathrm{I}_{d_\ell}\right) \mathrm{W}_\ell,$$

where the regularization parameter α controls the strength of the orthonormality constraint. The value of α was was selected from the set $\{1.0 \times 10^{-5}, 2.0 \times 10^{-5}, 3.0 \times 10^{-5}, 4.0 \times 10^{-5}, 5.0 \times 10^{-5}, 6.0 \times 10^{-5}, 7.0 \times 10^{-5}\}$ and the optimal value was found to be equal to 5.0×10^{-5} for both the tanh and ReLU.

In Fig. 10 we present additional experimental results that provide further insight into the clustering and compression phenomena for both tanh and ReLU nonlinearities. Fig. 10(a) shows what happens when the additive noise has a high variance. In this case, although saturation still occurs (see the histograms on top of Fig. 10(a)) and the Gaussians still cluster together (see the scatter plots on the right for the epoch 54 and epoch 8990), compression overall is very mild. The effect of increasing the noise parameter was explained in Section **??** of the main text (see, in particular, Fig. **??**(d) therein). Comparing Fig. 10(a) to Fig. **??**(a) of the main text, for which $\beta = 0.005$ was used and compression was observed, further highlights the effect of large β. Recall that smaller β values correspond to narrow Gaussians, while larger β values correspond to wider Gaussians. When β is small, even Gaussians that belong to the same cluster are distinguishable so long as they are not too close. When clusters tighten, the in-class movement brings these Gaussians closer together, effectively merging them, and causing a reduction

[*] Equal contribution [1] Massachusetts Institute of Technology [2] MIT-IBM Watson AI Lab [3] IBM Research. Correspondence to: Ziv Goldfeld <zivg@mit.edu>.

Proceedings of the 36^th International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

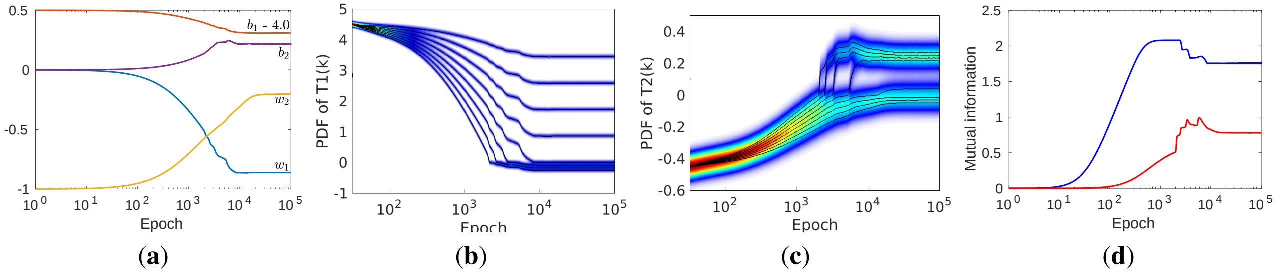

Figure 9. Two-layer leaky ReLU network: (a) network parameters as a function of epoch, (b,c) the corresponding PDFs $p_{T_1(k)}$ and $p_{T_2(k)}$, and (d) the mutual information for both layers.

in mutual information (compression). One the other hand, for large β, the in-class movement is blurred at the outset (before clusters tighten). Thus, the only effect on mutual information is the separation between the clusters: as these blobs move away from each other, mutual information rises.

Based on the above observation, we can conclude that while the two notions of "clustering Gaussians" and "compression/decrease in mutual information" are strongly related in the low-beta regime, once the noise becomes large, these phenomena decouple, i.e., the network may cluster inputs and neurons may saturate, but this will not be reflected in a decrease of mutual information.

Finally, we present results for ReLU activation without weight normalization (Fig. 10(b)) and with orthonormal weight regularization (Fig. 10(c)). We see that both these networks exhibit almost no compression. For Fig. 10(c), the lack of compression is attributed to regularization of the weight matrices, as explained in Section **??** of the main text. For Fig. 10(b), the reduction in compression can be explained by the fact that although ReLU forces saturation of the neurons at the origin (which promotes clustering), since the positive axes remain unconstrained, the Gaussians can move off towards infinity without bound. This is visible from the histograms in the top row of Fig. 10(b), where, for example, in layer 5 the neurons can take arbitrarily large positive values (note that the bin corresponding to the value 5 accumulates all the values from 5 to infinity). Therefore, the clustering at the origin and the potential drop in mutual information is counterbalanced by the spread of Gaussians along the positive axes and the potential increase of mutual information it causes. Eventually, this leads to the approximately constant profile of the mutual information plot in Fig. 10(b).

The behavior of the weight-normalized ReLU in Fig. 10(c) is similar to Fig. 10(b), although now the growth of the network weights is bounded and the saturation around origin is reduced. For example, for layers 4 and 5 we can see an upward trend in the mutual information, which is then flattened at the end of training. This occurs since more Gaussians are moving away from the origin, although their motion remains bounded (see the histograms on the top and the scatter plots on the right), thus decreasing the clustering density, leading to the rise in the mutual information profile. Once the Gaussians are prevented from moving any further along the positive axes, a slight compression occurs and the mutual information flattens.

8.2. Spiral Model

In this section we present results for another synthetic example. We generated data in the form of spiral as in Fig. 11. The network architecture was similar to SZT model, except that the size of each layer was set to 3.

Fig. 12 shows MI estimates $I(X;T_\ell)$ computed using SP estimator and the discrete entropy estimates $H\big(\mathsf{Bin}(T_\ell)\big)$ for weight un-normalized Fig. 12 (a) and normalized models Fig. 12 (b) and using additive noise $\beta = 0.005$. Similar as in the main paper, the results in the figure illustrate a connection between clustering and compression.

Finally, in Fig. 13 we also show an estimate of $H\big(\mathsf{Bin}(T_\ell)\big)$ for the case of deterministic DNN trained on spiral data. For the particular choice of the bin size, the result of the estimated entropy reveal a certain level of clustering granularity.

8.3. MNIST CNN

In this section, we describe in detail the architecture of the MNIST CNN models used in Sections **??** and **??** in the main paper.

The MNIST CNNs were trained using PyTorch (Paszke et al., 2017) version `0.3.0.post4`. The CNNs use the following fairly standard architecture with two convolutional layers, two fully connected layers, and batch normalization.

1. 2-d convolutional layer with 1 input channel, 16 output channels, 5x5 kernels, and input padding of 2 pixels

2. Batch normalization

3. Tanh() activation function

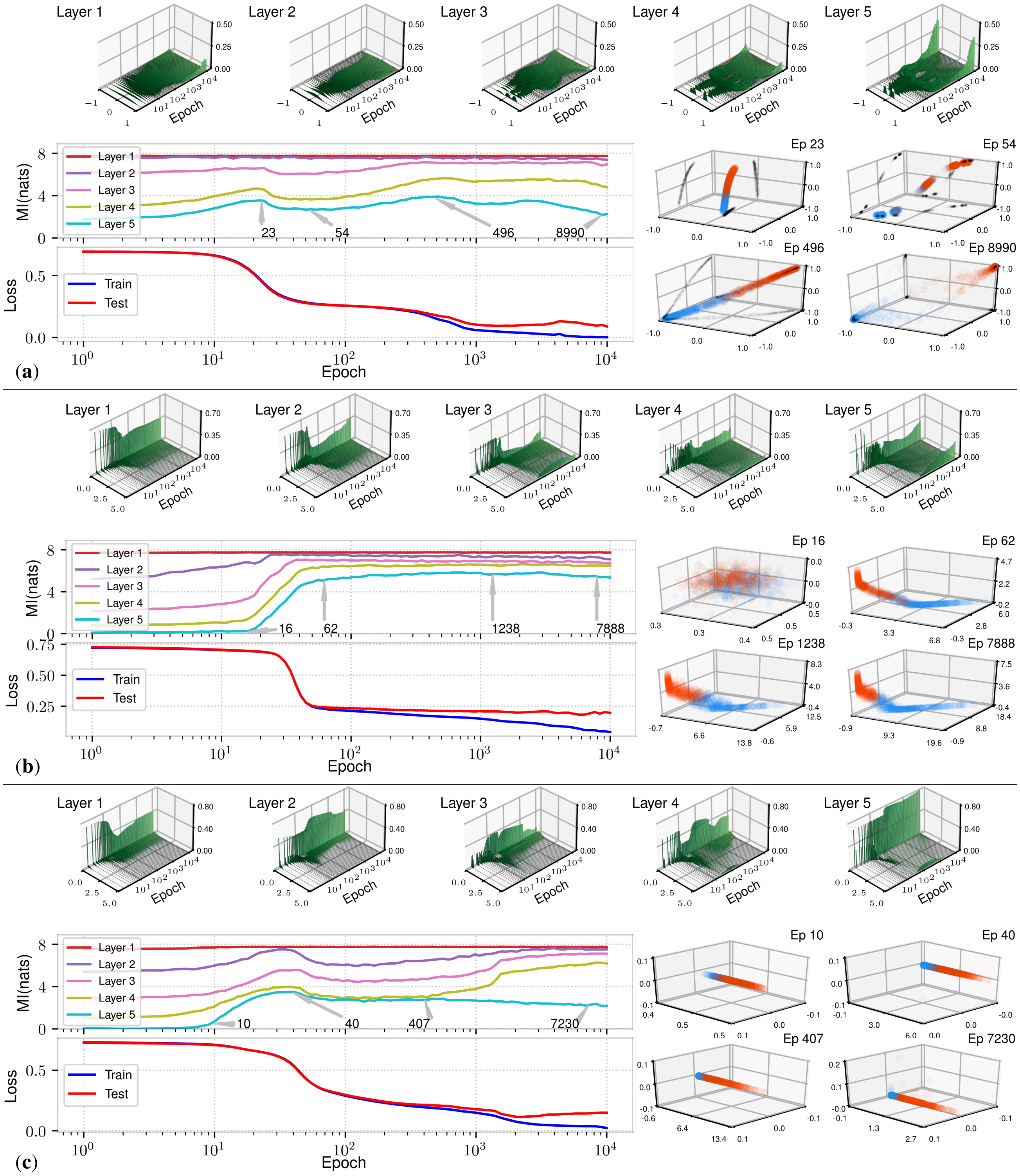

Figure 10. SZT model with (a) tanh nonlinearity and additive noise $\beta = 0.01$ without weight normalization, (b) ReLU nonlinearity and $\beta = 0.01$ without weight normalization, (c) ReLU nonlinearity and $\beta = 0.01$ with weight normalization. Test classification accuracy is 97%, 96%, and 97%, respectively.

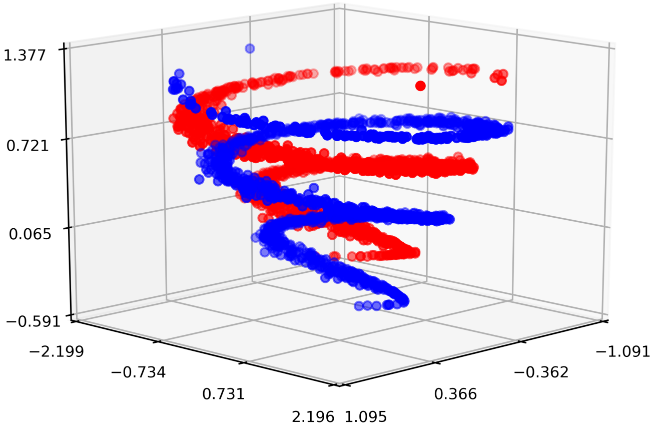

Figure 11. Generated spiral data for binary classification problem.

4. Zero-mean additive Gaussian noise with variance β^2 or dropout with a dropout probability of 0.2

5. 2x2 max-pooling

6. 2-d convolutional layer with 16 input channels, 32 output channels, 5x5 kernels, and input padding of 2 pixels

7. Batch normalization

8. Tanh() activation function

9. Zero-mean additive Gaussian noise with variance β^2 or dropout with a dropout probability of 0.2

10. 2x2 max-pooling

11. Fully connected layer with 1586 (32x7x7) inputs and 128 outputs

12. Batch normalization

13. Tanh() activation function

14. Zero-mean additive Gaussian noise with variance β^2 or dropout with a dropout probability of 0.2

15. Fully connected layer with 128 inputs and 10 outputs

All convolutional and fully connected layers have weights and biases, and the weights are initialized using the default initialization, which draws weights from $\mathsf{Unif}[-1/\sqrt{m}, 1/\sqrt{m}]$, with m the fan-in to a neuron in the layer. Training uses cross-entropy loss, and is performed using stochastic gradient descent with no momentum, 128 training epochs, and 32-sample minibatches. The initial learning rate is 5×10^{-3}, and it is reduced following a geometric schedule such that the learning rate in the final epoch is 5×10^{-4}. To improve the test set performance of our models, we applied data augmentation to the training set by translating, rotating, and shear-transforming each training example each time it was selected. Translations in the x- and y-directions were drawn uniformly from $\{-2, -1, 0, 1, 2\}$, rotations were drawn from $\mathsf{Unif}(-10^\circ, 10^\circ)$, and shear transforms were drawn from $\mathsf{Unif}(-10^\circ, 10^\circ)$.

To obtain more reliable performance results, we train eight different models and report the mean number of errors and standard deviation of the number of errors on the MNIST validation set. To ensure that the internal representations of different models are comparable, which is necessary for the use of the cosine similarity measure between internal representations, for each noise condition (deterministic, noisy with $\beta = 0.05$, noisy with $\beta = 0.1$, noisy with $\beta = 0.2$, noisy with $\beta = 0.5$, and dropout with $p = 0.2$), we use a common random seed (different for the eight replications, of course) so the models have the same initial weights and access the training data in the same order (use the same minibatches).

At test time, all models are fully deterministic: the additive noise blocks and dropout layers are replaced by identities. Thus, in the figures and text in the main paper, “Layer 1” is the output of step 5 (2x2 max-pooling), “Layer 2” is the output of step 10 (2x2 max-pooling), “Layer 3” is the output of step 13 (Tanh() activation function), and “Layer 4” is the output of step 15 (fully connected layer with 10 outputs).

9. Sample Propagation Estimator - Theoretic Guarantees

In this section we state performance guarantees for the SP estimator. We cite several foundational theorems from our work (Goldfeld et al., 2019), where this estimation problem is thoroughly studied. An anonymized copy of that paper is found at the end of the supplement and cited when needed. Proofs of all other results are relegated to Supplement 10.

9.1. Preliminary Definitions

Consider the estimation of the differential entropy $h(S+Z) = h(P * \varphi_\beta)$ based on n i.i.d. samples of $S \sim P$, where P is unknown and belongs to some nonparametric class, and φ_β (a PDF of an isotropic Gaussian with parameter β) is known. The minimax absolute-error risk over a given nonparametric class of distributions $\mathcal{F}$ is

$$\mathcal{R}^\star(n, \beta, \mathcal{F}) \triangleq \inf_{\hat{h}} \sup_{P \in \mathcal{F}} \mathbb{E}\left| h(P * \varphi_\beta) - \hat{h}(S^n, \beta) \right|, \quad (5)$$

where $\hat{h}$ is the estimator and $S^n \triangleq (S_i)_{i \in [n]}$ are the samples from P. In (5), by $P * \varphi_\beta$ we mean either: (i) $(P * \varphi_\beta)(x) = \int p(u)\varphi_\beta(x-u)du = (p * \varphi_\beta)(x)$, when P is continuous with density p; or (ii) $(P * \varphi_\beta)(x) =$

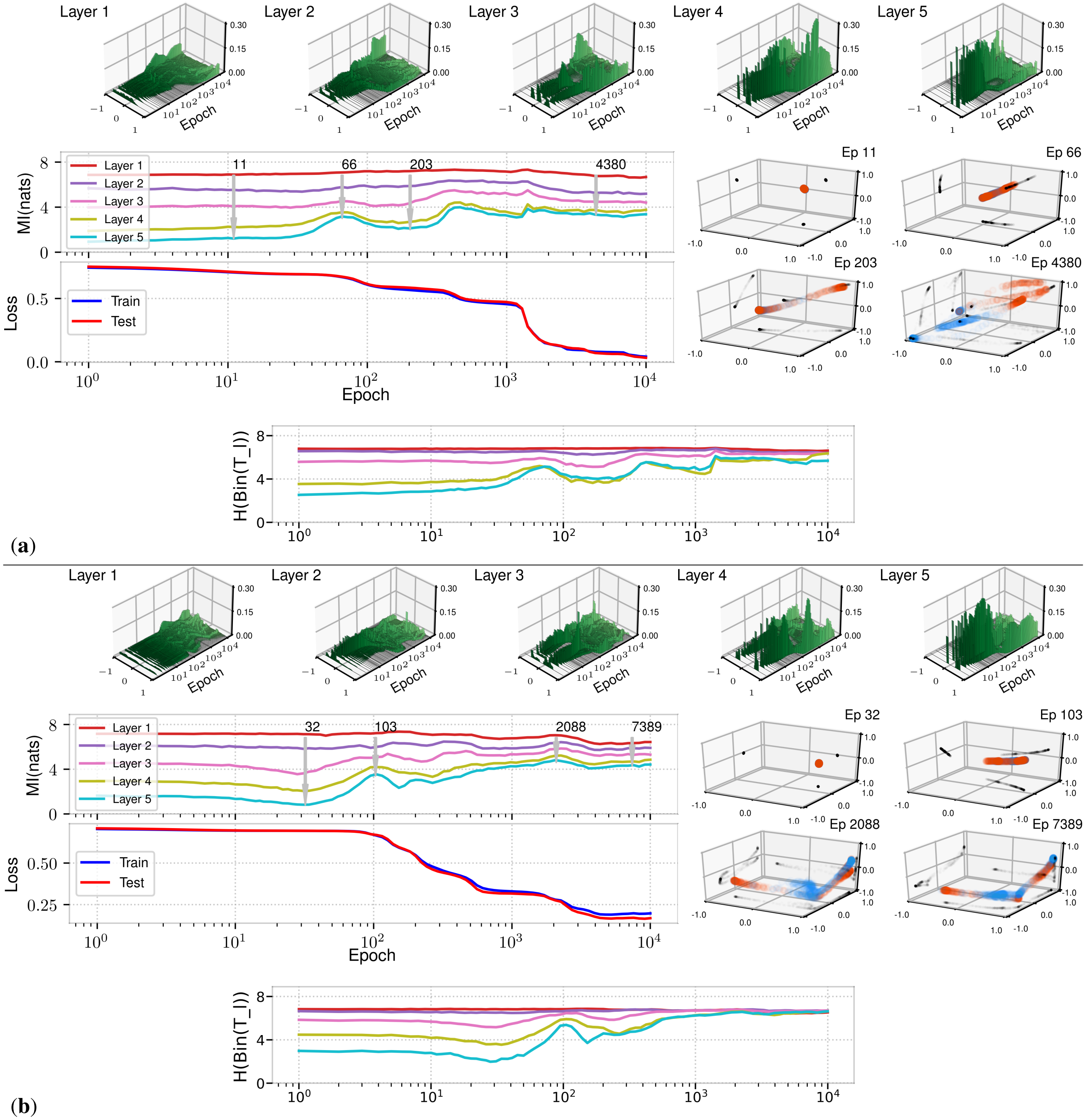

Figure 12. (a) Evolution of $I(X;T_\ell)$ and training/test losses across training epochs for Spiral dataset with $\beta = 0.005$ and tanh nonlinearities. The scatter plots on the right are the values of Layer 5 ($d_5 = 3$) at the arrow-marked epochs on the mutual information plot. The bottom plot shows the entropy estimate $H\big(\mathsf{Bin}(T_\ell)\big)$ across epochs for bin size $B = 10\beta$. (b) Same setup as in (a) but with a regularization that encourages orthonormal weight matrices.

$\sum_{u:\ p(u)>0} p(u)\varphi_\beta(x-u)$, if P is discrete with PMF p. This convolved distribution can be defined generally in a way that the two instances above as special cases using measure-theoretic concepts (see (Goldfeld et al., 2019)). Regardless of the nature of P, however, we stress that $P*\varphi_\beta$ is always a continuous distribution since it corresponds to the random variable $S+Z$, where Z is an isotropic Gaussian vector. The sample complexity $n^\star(\eta,\beta,\mathcal{F})$ is defined as the smallest number of samples n required to achieve a risk value less than or equal to a specified constant η in (5).

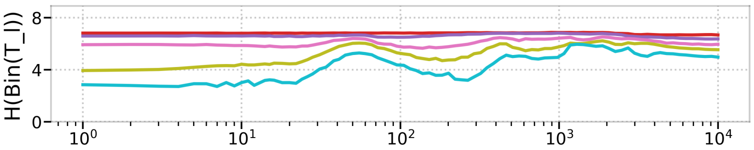

Figure 13. $H\big(\mathsf{Bin}(T_\ell)\big)$ estimate for deterministic net using spiral data. Bin size was set to $B = 0.001$.

Let $\mathcal{F}_d$ be the set of distributions P with $\mathrm{supp}(P) \subseteq [-1,1]^d$.[1] Furthermore, let $\mathcal{F}^{(\mathsf{SG})}_{d,\mu,K}$ be the class of K-subgaussian distributions, where we adopt the subgaussianity definition from (Hsu et al., 2012). Namely, $P \in \mathcal{F}^{(\mathsf{SG})}_{d,\mu,K}$, for $\mu \geq 0$ and $K > 0$, if $X \sim P$ satisfies $\|\mathbb{E}X\| \leq \mu$ and

$$\mathbb{E}\Big[\exp\big(\alpha^T(X - \mathbb{E}X)\big)\Big] \leq \exp\big(0.5K^2\|\alpha\|^2\big), \forall \alpha \in \mathbb{R}^d, \tag{6}$$

i.e., every one-dimensional projection of X is subgaussian. Clearly, there exists a $K' > 0$ such that $\mathcal{F}_d \subseteq \mathcal{F}^{(\mathsf{SG})}_{d,0,K'}$. We therefore state our lower bound results (Theorem 2) for $\mathcal{F}_d$, while the upper bound (Theorem 3) is given for $\mathcal{F}^{(\mathsf{SG})}_{d,\mu,K}$. The class $\mathcal{F}_d$ corresponds to hidden layers with bounded nonlinearities (such as tanh or sigmoid), while $\mathcal{F}^{(\mathsf{SG})}_{d,\mu,K}$ accounts for ReLU nonlinearities (when, for example, the input X is itself subgaussian).

9.2. Sample Complexity is Exponential in Dimension

We start with Theorem 1 from (Goldfeld et al., 2019), which states that the sample complexity of any good estimator of $h(P * \varphi_\beta)$ (to within an additive gap η) is exponential in d.

Theorem 2 (Theorem 1 from (Goldfeld et al., 2019))**.** *The following holds:*

1. *Fix $\beta > 0$. There exist $d_0(\beta) \in \mathbb{N}$, $\eta_0(\beta) > 0$ and $\gamma(\beta) > 0$ (monotonically decreasing in β), such that for all $d \geq d_0(\beta)$ and $\eta < \eta_0(\beta)$ we have sample complexity $n^\star(\eta, \beta, \mathcal{F}_d) \geq \Omega\left(\frac{2^{\gamma(\beta)d}}{d\eta}\right)$.*
2. *Fix $d \in \mathbb{N}$. There exist $\beta_0(d), \eta_0(d) > 0$, such that for all $\beta < \beta_0(d)$ and $\eta < \eta_0(d)$ we have sample complexity $n^\star(\eta, \beta, \mathcal{F}_d) \geq \Omega\left(\frac{2^d}{\eta d}\right)$.*

The exponent $\gamma(\beta)$ being monotonically decreasing in β suggests that larger values of β are favorable for estimation. Part 1 of the theorem states that an exponential sample complexity is inevitable when d is large. As a complementary result, the second part gives a sample complexity lower bound valid in any dimension for a small noise parameter. Nonetheless, the result accounts for orders of β considered in this work.

Remark 1 (Critical $\boldsymbol{\beta}$ Values)**.** *Theorem 2 is stated in asymptotic form for simplicity. We note that, for any d, the critical $\beta_0(d)$ value from the second part can be extracted by following the constants through the proof (which relies on Proposition 3 from (Wu & Yang, 2016)). These critical values are not unreasonably small. For example for $d = 1$, a careful analysis gives that Theorem 2 holds for all $\beta < 0.08$, which is satisfied by most of the experiments in this paper. This threshold on β changes very slowly when increasing d due to the rapid decay of the PDF of the normal distribution.*

9.3. Estimation Risk Bounds

We next focus on analyzing the performance of the SP mutual information estimator. We start by citing Theorem 2 of (Goldfeld et al., 2019), where the risk of the entropy estimation problem is bounded. Recall that the estimator of $h(P * \varphi_\beta)$ is $h(\hat{P}_{S^n} * \varphi_\beta)$, where $S^n = (S_i)_{i=1}^n$ is an i.i.d. sample set from P and $\hat{P}_{S^n}$ is their empirical distribution. The following theorem shows that the expected absolute error of this estimator decays at a rate of estimation $O\left(\frac{c^d}{\sqrt{n}}\right)$, for a numerical constant c and all dimensions d. A better rate of convergence with n cannot be attained due to the parametric estimation lower bound (see, e.g., Proposition 1 of (Chen, 1997)). The exponential dependence in d is also necessary as established by Theorem 2.

Theorem 3 (Theorem 2 from (Goldfeld et al., 2019))**.** *Fix $\beta > 0, d \geq 1$. Then*

$$\begin{aligned}
&\sup_{P \in \mathcal{F}^{(\mathsf{SG})}_{d,\mu,K}} \mathbb{E}\left|h(P * \varphi_\beta) - h(\hat{P}_{S_n} * \varphi_\beta)\right| \\
&\leq \left(\frac{1}{\sqrt{2}} + \frac{K}{\beta}\right)^{\frac{d}{2}} \\
&\qquad \times \left(\frac{8\big(2\mu^4 + 32d^2K^4 + d(d+2)(K + \beta/\sqrt{2})^4\big)}{\beta^4}\right)^{\frac{1}{2}} \\
&\qquad \times \exp\left(\frac{3d}{16} + \frac{\mu^2}{4(K + \beta/\sqrt{2})^2}\right)\frac{1}{\sqrt{n}}.
\end{aligned} \tag{7}$$

Remark 2 (Improved Constant for Bounded Support)**.** *Theorem 3 also applies to the narrower nonparametric class $\mathcal{F}_d$*

[1] Any support included in a compact subset of $\mathbb{R}^d$ would do. We focus on the case of $\mathrm{supp}(P) \subseteq [-1,1]^d$ due to its correspondence to a noisy DNN with tanh nonlinearities.

in place of $\mathcal{F}^{(\mathsf{SG})}_{d,\mu,K}$. By directly analyzing this bounded support scenario[2] *($P \in \mathcal{F}_d$) one may improve the constant factor in Theorem 3 to give a bound of* $\max\{1, \beta^{-d}\} 2^{d+2}\sqrt{\frac{d}{n}}$.

Remark 3 (Comparison to Generic Estimators)**.** *Note that one could always sample φ_β and add the obtained noise samples to S^n to obtain a sample set from $P * \varphi_\beta$. These samples can be used to get a proxy of $h(P * \varphi_\beta)$ via a kNN- or a KDE-based differential entropy estimator. However, $P * \varphi_\beta$ violates the boundedness away from zero assumption that most of the convergence rate results in the literature rely on (Levit, 1978; Hall, 1984; Joe, 1989; Hall & Morton, 1993; Tsybakov & Van der Meulen, 1996; Haje & Golubev, 2009; Sricharan et al., 2012; Singh & Póczos, 2016; Kandasamy et al., 2015). Two recent works that weakened/dropped the boundedness from below assumption, providing general-purpose estimators whose risk bounds are valid in our setup, are (Han et al., 2017) and (Berrett et al., 2019). However, the analysis of the KDE-based estimator proposed in (Han et al., 2017) holds only for Lipschitz smoothness parameters up to $s \leq 2$ and attains the slow rate (overlooking multiplicative polylogarithmic factors) of $O\left(n^{-\frac{s}{s+d}}\right)$. The second work (Berrett et al., 2019) studies a weighted-kNN estimator in the high smoothness regime and proved its asymptotic efficiency. However, no explicit risk bounds were derived in that work and empirically the estimator is significantly outperformed by $h(\hat{P}_{S^n} * \varphi_\beta)$ (see Section V of (Goldfeld et al., 2019)).*

We now show how the theoretical guarantee on the accuracy of the differential entropy estimator (Theorem 3) translates to mutual information estimation via the SP estimator from (**??**). To formulate the claim, recall that $T_\ell = S_\ell + Z_\ell$, where $S_\ell \sim P_{S_\ell} = P_{f_\ell(T_{\ell-1})}$ and $Z_\ell \sim \mathcal{N}(0, \beta^2 \mathrm{I}_{d_\ell})$ are independent. Thus,

$$h(T_\ell) = h(P_{S_\ell} * \varphi_\beta) \tag{8a}$$

$$h(T_\ell | X = x) = h(P_{S_\ell|X=x_i} * \varphi_\beta). \tag{8b}$$

Provided n i.i.d. samples $\mathcal{X} = \{X_i\}_{i\in[n]}$ from P_X, the DNN's generative model enables sampling from P_{S_ℓ} and $P_{S_\ell|X}$ as follows:

1. **Unconditional Sampling:** To generate the sample set from P_{S_ℓ}, feed each X_i, for $i \in [n]$, into the DNN and collect the outputs it produces at the $(\ell-1)$-th layer. The function f_ℓ is then applied to each collected output to obtain $S_\ell^n \triangleq \{S_{\ell,1}, S_{\ell,2}, \ldots, S_{\ell,n}\}$, which is a set of n i.i.d. samples from P_{S_ℓ}.
2. **Conditional Sampling Given $\boldsymbol{X}$:** To generate i.i.d. samples from $P_{S_\ell|X=x_i}$, for $i \in [n]$, we feed X_i into the DNN n times, collect outputs from $T_{\ell-1}$ corresponding to different noise realizations, and apply f_ℓ on each. Denote the obtained samples by $S_\ell^n(X_i)$.[3]

The knowledge of φ_β and the generated samples S_ℓ^n and $S_\ell^n(X_i)$ can be used to estimate the unconditional and the conditional entropies, from (8a) and (8b), respectively.

For notational simplicity, the layer index ℓ is dropped for the remainder of this subsection. With the above sampling procedure we construct an estimator $\hat{I}_{\mathsf{SP}}\big(X^n, \hat{h}\big)$ of $I(X;T)$ based on a given estimator $\hat{h}(A^n, \beta)$ of $h(P * \varphi_\beta)$ for $P \in \mathcal{F}_d$ that uses i.i.d. samples $A^n = (A_1, \ldots, A_n)$ from P and knowledge of φ_β. Assume that $\hat{h}$ attains

$$\sup_{P\in\mathcal{F}_d} \mathbb{E}\left| h(P * \varphi_\beta) - \hat{h}(A^n, \beta)\right| \leq \Delta_{\beta,d}(n). \tag{9}$$

An example of such an $\hat{h}$ is the estimator $h(\hat{P}_{S^n} * \varphi_\beta)$ from Theorem 3; the corresponding $\Delta_{\beta,d}(n)$ term is the RHS of (7). Our SP mutual information estimator is (see (**??**))

$$\hat{I}_{\mathsf{SP}}\left(X^n, \hat{h}, \beta\right) \triangleq \hat{h}(S^n, \beta) - \frac{1}{n}\sum_{i=1}^n \hat{h}\big(S^n(X_i), \beta\big). \tag{10}$$

The following theorem bounds the expected absolute error of $\hat{I}_{\mathsf{SP}}\left(X^n, \hat{h}, \beta\right)$. The proof is given in Supplement 10.1.

Theorem 4. *For the above described setup, we have*

$$\sup_{P_X} \mathbb{E}\left| I(X;T) - \hat{I}_{\mathsf{SP}}\left(X^n, \hat{h}, \beta\right)\right| \leq 2\Delta_{\beta,d}(n) + \frac{d\log\left(1 + \frac{1}{\beta^2}\right)}{4\sqrt{n}}. \tag{11}$$

Theorem **??** of the main text is an immediate consequence of Theorems 3 and 4. Interestingly, the quantity $\frac{1}{\beta^2}$ is the signal-to-noise ratio (SNR) between S and Z. The larger β is the easier estimation becomes, since the noise smooths out the complicated P_X distribution. Also note that the dimension of the ambient space in which X lies does not appear in the absolute-risk bound for estimating $I(X;T)$. The bound depends only on the dimension of T (through $\Delta_{\beta,d}$). This is because the additive noise resides in the T domain, limiting the possibility of encoding the rich structure of X into T in full. On a technical level, the blurring effect caused by the noise enables uniformly lower bounding $\inf_x h(T|X=x)$ and thereby controlling the variance

[2] e.g., by employing Proposition 5 from (Polyanskiy & Wu, 2016) to control the entropy difference via a Wasserstein 1 distance and them using Theorem 6.15 from (Villani, 2006) to bound the latter by an expression that lands itself for an elementary analysis.

[3] The described sampling procedure is valid for any layer $\ell \geq 2$. For $\ell = 1$, S_1 coincides with $f_1(X)$ but the conditional samples are undefined. Nonetheless, noting that for the first layer $h(T_1|X) = h(Z) = \frac{d}{2}\log(2\pi e\beta^2)$, we see that no estimation of the conditional entropy is needed. The mutual information estimator given in (10) is modified by replacing the subtracted term with $h(Z)$.

of the estimator for each conditional entropy. In turn, this reduces the impact of X on the estimation of $I(X;T)$ to that of an empirical average converging to its expected value with rate $\frac{1}{\sqrt{n}}$.

9.4. Sample Propagation Estimator Bias

The results of the previous subsection are of a minimax flavor. That is, they state worst-case convergence rates of $h(P * \varphi_\beta)$ estimation over a nonparametric class of distributions. In practice, the true distribution may not be one that attains these worst-case rates, and convergence may be faster. However, while variance of $h(\hat{P}_{S^n} * \varphi_\beta)$ can be empirically evaluated using bootstrapping, there is no empirical test for the bias. Specifically, even if multiple estimations of $h(P * \varphi_\beta)$ via $h(\hat{P}_{S^n} * \varphi_\beta)$ consistently produce similar values, this does not necessarily suggest that these values are close to the true $h(P * \varphi_\beta)$. To have a guideline to the least number of samples needed to avoid biased estimation, we present the following lower bound on the estimation bias.

Theorem 5. *Fix $\beta > 0$, $d \geq 1$, and let $\epsilon \in \left(1 - \left(1 - 2Q\left(\frac{1}{2\beta}\right)\right)^d, 1\right]$, where Q is the Q-function.*[4] *Set $k_\star \triangleq \left\lfloor \frac{1}{\beta Q^{-1}\left(\frac{1}{2}\left(1-(1-\epsilon)^{\frac{1}{d}}\right)\right)} \right\rfloor$, where Q^{-1} is the inverse of the Q-function. By the choice of ϵ, clearly $k_\star \geq 2$, and the bias of the SP estimator over the class $\mathcal{F}_d$ is bounded as*

$$\sup_{P \in \mathcal{F}_d} \left| h(P * \varphi_\beta) - \mathbb{E} h(\hat{P}_{S^n} * \varphi_\beta) \right| \geq \log\left(\frac{k_\star^{d(1-\epsilon)}}{n}\right) - H_b(\epsilon). \tag{12}$$

Consequently, the bias cannot be less than a given $\delta > 0$ so long as $n \leq k_\star^{d(1-\epsilon)} \cdot e^{-(\delta + H_b(\epsilon))}$.

Theorem 5 is proved in Supplement 10.2. Since $H_b(\epsilon)$ shrinks with ϵ, for sufficiently small ϵ values the lower bound from (12) shows that the SP estimator will not have negligible bias unless $n > k_\star^{d(1-\epsilon)}$ is satisfied. The condition $\epsilon > 1 - \left(1 - 2Q\left(\frac{1}{2\beta}\right)\right)^d$ is non-restrictive in any relevant regime of β and d. For instance, for typical β values we work with - around 0.1 - this lower bound is at most 0.0057 for all dimensions up to at least $d = 10^4$. Setting, e.g., $\epsilon = 0.01$ (for which $H_b(0.01) \approx 0.056$), the corresponding $k_\star$ equals 3 for $d \leq 11$ and 2 for $12 \leq d \leq 10^4$. Thus, with these parameters, in order to have negligible bias the number of estimation samples n should be at least $2^{0.99d}$, for any conceivably relevant dimension d.

9.5. Computing the Sample Propagation Estimator

Evaluating the SP mutual information estimator requires computing the differential entropy of a Gaussian mixture. Although it cannot be computed in closed form, this section presents a method for approximate computation via MCI (Robert, 2004). To simplify the presentation, we present the method for an arbitrary Gaussian mixture without referring to the notation of the estimation setup.

Let $g(t) \triangleq \frac{1}{n}\sum_{i\in[n]} \varphi_\beta(t-\mu_i)$ be a d-dimensional n-mode Gaussian mixture, with $\{\mu_i\}_{i\in[n]} \subset \mathbb{R}^d$ and φ_β as the PDF of $\mathcal{N}(0, \beta^2 \mathrm{I}_d)$. Let $C \sim \mathsf{Unif}\{\mu_i\}_{i\in[n]}$ be independent of $Z \sim \mathcal{N}(0, \beta^2 \mathrm{I}_d)$ and note that $V \triangleq C + Z \sim g$.

We use MCI (Robert, 2004) to compute $h(g)$. First note that

$$\begin{aligned} h(g) &= -\mathbb{E}\log g(V) \\ &= -\frac{1}{n}\sum_{i\in[n]} \mathbb{E}\Big[\log g(\mu_i + Z)\Big| C = \mu_i\Big] \\ &= -\frac{1}{n}\sum_{i\in[n]} \mathbb{E}\log g(\mu_i + Z), \end{aligned} \tag{13}$$

where the last step follows by the independence of Z and C. Let $\left\{Z_j^{(i)}\right\}_{\substack{i\in[n] \\ j\in[n_{\mathsf{MC}}]}}$ be $n \times n_{\mathsf{MC}}$ i.i.d. samples from φ_β. For each $i \in [n]$, we estimate the i-th summand on the RHS of (13) by

$$\hat{L}_{\mathsf{MC}}^{(i)} \triangleq \frac{1}{n_{\mathsf{MC}}} \sum_{j\in[n_{\mathsf{MC}}]} \log g\left(\mu_i + Z_j^{(i)}\right), \tag{14a}$$

which produces

$$\hat{h}_{\mathsf{MC}} \triangleq \frac{1}{n}\sum_{i\in[n]} \hat{L}_{\mathsf{MC}}^{(i)} \tag{14b}$$

as our estimate of $h(g)$. Define the mean squared error (MSE) of $\hat{h}_{\mathsf{MC}}$ as

$$\mathsf{MSE}\left(\hat{h}_{\mathsf{MC}}\right) \triangleq \mathbb{E}\left[\left(\hat{h}_{\mathsf{MC}} - h(g)\right)^2\right]. \tag{15}$$

We have the following bounds on the MSE for tanh and ReLU networks.

Theorem 6 (MSE Bounds for MC Estimator)**.** *The following holds:*

1. *Assume $C \in [-1,1]^d$ almost surely (i.e., tanh network), then*

$$\mathsf{MSE}\left(\hat{h}_{\mathsf{MC}}\right) \leq \frac{2d(2+\beta^2)}{\beta^2}\frac{1}{n \cdot n_{\mathsf{MC}}}. \tag{16}$$

[4] The Q-function is defined as $Q(x) \triangleq \frac{1}{\sqrt{2\pi}}\int_x^\infty e^{-\frac{t^2}{2}}\,dt$.

2. *Assume $M_C \triangleq \mathbb{E}\|C\|_2^2 < \infty$ (e.g., ReLU network with bounded second moments), then*

$$\begin{aligned}&\mathsf{MSE}\left(\hat{h}_{\mathsf{MC}}\right)\\ &\le \frac{9d\beta^2 + 8(2+\beta\sqrt{d})M_C + 3(11\beta\sqrt{d}+1)\sqrt{M_C}}{\beta^2}\\ &\qquad\qquad\times \frac{1}{n\cdot n_{\mathsf{MC}}}.\end{aligned} \tag{17}$$

The proof of Theorem 6 is found in Supplement 10.3. The MSE bounds scale only linearly with the dimension d, making β^2 in the denominator often the dominating factor experimentally.

10. Proofs

10.1. Proof of Theorem 4

Fix P_X, define $g(x) \triangleq h(T|X=x) = h(P_{S|X=x} * \varphi_\beta)$ and write

$$I(X;T) = h(T) - h(T|X) = h(P_S * \varphi_\beta) - \mathbb{E}g(X). \tag{18}$$

Applying the triangle inequality to (10) we obtain

$$\begin{aligned}&\mathbb{E}\left|\hat{I}_{\mathsf{SP}}\left(X^n, \hat{h}, \beta\right) - I(X;T)\right|\\ &\quad\le \mathbb{E}\left|\hat{h}(S^n,\beta) - h(P_S * \varphi_\beta)\right|\\ &\qquad\qquad + \mathbb{E}\left|\frac{1}{n}\sum_{i=1}^n \hat{h}\big(S^n(X_i),\beta\big) - \mathbb{E}g(X)\right|\\ &\quad\le \underbrace{\mathbb{E}\left|\hat{h}(S^n,\beta) - h(P_S * \varphi_\beta)\right|}_{\text{(I)}}\\ &\qquad\qquad + \underbrace{\frac{1}{n}\sum_{i=1}^n \mathbb{E}\left|\hat{h}\big(S^n(X_i),\beta\big) - g(X_i)\right|}_{\text{(II)}}\\ &\qquad\qquad\quad + \underbrace{\mathbb{E}\left|\frac{1}{n}\sum_{i=1}^n g(X_i) - \mathbb{E}g(X)\right|}_{\text{(III)}}\end{aligned} \tag{19}$$

By assumption (9) and because $\operatorname{supp}(P_S) \subseteq [-1,1]^d$, we have

$$\mathbb{E}\left|\hat{h}(S^n,\beta) - h(P_S * \varphi_\beta)\right| \le \Delta_{\beta,d}(n). \tag{20}$$

Similarly, for any fixed $X^n = x^n$, $\operatorname{supp}(P_{S|X=x_i}) \subseteq [-1,1]^d$ for all x_i, where $i \in [n]$, and hence

$$\begin{aligned}&\mathbb{E}\left[\left|\hat{h}(S^n(X_i),\beta) - g(X_i)\right|\,\middle|\,X^n = x^n\right]\\ &\qquad\overset{(a)}{=} \mathbb{E}\left|\hat{h}\big(S^n(x_i),\beta\big) - h(P_{S|X=x_i} * \varphi_\beta)\right|\\ &\qquad\le \Delta_{\beta,d}(n),\end{aligned} \tag{21}$$

where (a) is because for a fixed x_i, sampling from $P_{S|X=x_i}$ corresponds to drawing multiple noise realization for the previous layers of the DNN. Since these noises are independent of X, we may remove the conditioning from the expectation. Taking an expectation on both sides of (21) and the law of total expectation we have

$$\text{(II)} = \frac{1}{n}\sum_{i=1}^n \mathbb{E}\left|\hat{h}(S^n(X_i)) - g(X_i)\right| \le \Delta_{\beta,d}(n). \tag{22}$$

Turning to term (III), observe that $\big\{g(X_i)\big\}_{i\in[n]}$ are i.i.d random variables. Hence

$$\frac{1}{n}\sum_{i=1}^n g(X_i) - \mathbb{E}g(X) \tag{23}$$

is the difference between an empirical average and the expectation. By monotonicity of moments we have

$$\begin{aligned}\text{(III)}^2 &= \left(\mathbb{E}\left|\frac{1}{n}\sum_{i=1}^n g(X_i) - \mathbb{E}g(X)\right|\right)^2\\ &\le \mathbb{E}\left[\left(\frac{1}{n}\sum_{i=1}^n g(X_i) - \mathbb{E}g(X)\right)^2\right]\\ &= \frac{1}{n}\mathsf{var}\big(g(X)\big)\\ &\le \frac{1}{4n}\left(\sup_x h(p_{T|X=x}) - \inf_x h(p_{T|X=x})\right)^2.\end{aligned} \tag{24}$$

The last inequality follows since $\mathsf{var}(A) \le \frac{1}{4}(\sup A - \inf A)^2$ for any random variable A.

It remains to bound the supremum and infimum of $h(p_{T|X=x})$ uniformly in $x \in \mathbb{R}^{d_0}$. By definition $T = S+Z$, where S and Z are independent and $Z \sim \mathcal{N}(0,\beta^2\mathrm{I}_d)$. Therefore, for all $x \in \mathbb{R}^{d_0}$

$$h(p_{T|X=x}) \ge h(S+Z|S,X=x) = \frac{d}{2}\log(2\pi e\beta^2), \tag{25}$$

where we have used the independence of Z and (S,X) and the fact that conditioning cannot increase entropy. On the other hand, denoting the entries of T by $T \triangleq \big(T(k)\big)_{k\in[d]}$, we can obtain an upper bound as

$$h(p_{T|X=x}) = h(T|X=x) \le \sum_{k=1}^d h\big(T(k)\big|X=x\big), \tag{26}$$

since independent random variables maximize differential entropy. Now for any $k \in [d]$, we have

$$\mathsf{var}\big(T(k)\big|X=x\big) \le \mathbb{E}\big[T^2(k)\big|X=x\big] \le 1+\beta^2, \tag{27}$$

since $S(k) \in [-1,1]$ almost surely. For a fixed variance the Gaussian distribution maximizes differential entropy, and therefore

$$h(p_{T|X=x}) \leq \frac{d}{2}\log\big(2\pi e(1+\beta^2)\big). \qquad (28)$$

for all $x \in \mathbb{R}^{d_0}$. Substituting the lower bound (25) and upper bound (28) into (24) gives

$$(\mathrm{III})^2 \leq \left(\frac{d\log\left(1+\frac{1}{\beta^2}\right)}{4\sqrt{n}}\right)^2. \qquad (29)$$

Inserting this along with (20) and (22) into the bound (19) bounds the expected estimation error as

$$\mathbb{E}\Big|\hat{I}_{\mathsf{SP}}\Big(X^n,\hat{h},\beta\Big)-I(X;T)\Big| \leq 2\Delta_{\beta,d}(n)+\frac{d\log\left(1+\frac{1}{\beta^2}\right)}{4\sqrt{n}}. \qquad (30)$$

Taking the supremum over P_X concludes the proof.

10.2. Proof of Theorem 5

First note that since $h(q)$ is concave in q and because $\mathbb{E}\hat{P}_{S^n}=P$, by Jensen's inequality we have

$$\mathbb{E}h(\hat{P}_{S^n}*\varphi_\beta) \leq h(P*\varphi_\beta). \qquad (31)$$

Now, let $W \sim \mathsf{Unif}([n])$ be independent of (S^n,Z) and define $Y=S_W+Z$. We have the following lemma.

Lemma 1. *The following equality holds:*

$$h(P*\varphi_\beta)-\mathbb{E}h(\hat{P}_{S^n}*\varphi_\beta)=I(S^n;Y). \qquad (32)$$

Proof. We expand $I(S^n;Y)=h(Y)-h(Y|S^n)$ and denote by F_A the cumulative distribution function (CDF) of a random variable A. Let $T=S+Z\sim P*\varphi_\beta$ and first note that

$$F_Y(y)=\mathbb{P}\big(S_W+Z\leq y\big)=\frac{1}{n}\sum_{i=1}^{n}\mathbb{P}(S_i+Z\leq y)=F_T(y). \qquad (33)$$

Thus, $h(Y)=h(P*\varphi_\beta)$.

It remains to show that $h(Y|S^n)=\mathbb{E}h(\hat{P}_{S^n}*\varphi_\beta)$. Fix $S^n=s^n$ and consider

$$F_{Y|S^n}(y|s^n)=\mathbb{P}\big(S_W+Z\leq y\big|S^n=s^n\big)=\frac{1}{n}\mathbb{P}\big(s_i+Z\leq y\big), \qquad (34)$$

which implies that the density $p_{Y|S^n=s^n}=\hat{P}_{s^n}*\varphi_\beta$. Consequently, $h(Y|S^n=s^n)=h(\hat{P}_{s^n}*\varphi_\beta)$, and by definition of conditional entropy $h(Y|S^n)=\mathbb{E}h(\hat{P}_{S^n}*\gamma)$.

□

Using the lemma, we have

$$\left|\sup_{P\in\mathcal{F}_d}\mathbb{E}h(P*\varphi_\beta)-h(\hat{P}_{S^n}*\varphi_\beta)\right|=\sup_{P\in\mathcal{F}_d}I(S^n;Y), \qquad (35)$$

where the right hand side is the mutual information between n i.i.d. random samples S_i from P and the random vector $Y=S_W+Z$, formed by choosing one of the S_i's at random and adding Gaussian noise.

To obtain a lower bound on the supremum, we consider the following P. Partition the hypercube $[-1,1]^d$ into k^d equal-sized smaller hypercubes, each of side length k. Denote these smaller hypercubes as $\mathsf{C}_1,\mathsf{C}_2,\ldots,\mathsf{C}_{k^d}$ (the exact order does not matter). For each $i\in[k^d]$ let $c_i\in\mathsf{C}_i$ be the centroid of the hypercube C_i. Let $\mathcal{C}\triangleq\{c_i\}_{i=1}^{k^d}$ and choose P as the uniform distribution over $\mathcal{C}$.

By the mutual information chain rule and the non-negativity of discrete entropy, we have

$$\begin{aligned}I(S^n;Y)&=I(S^n;Y,S_W)-I(S^n;S_W|Y)\\&\overset{(a)}{\geq} I(S^n;S_W)-H(S_W|Y)\\&=H(S_W)-H(S_W|S^n)-H(S_W|Y),\end{aligned} \qquad (36)$$

where step (a) uses the independence of (S^n,W) and Z. Clearly $H(S_W)=\log|\mathcal{C}|$, while $H(S_W|S^n)\leq H(S_W,W|S^n)\leq H(W)=\log n$, via the independence of W and S^n. For the last (subtracted) term in (36) we use Fano's inequality to obtain

$$\begin{aligned}H(S_W|Y)&\leq H\big(S_W\big|\psi_{\mathcal{C}}(Y)\big)\\&\leq H_b\big(\mathsf{P}_{\mathsf{e}}(\mathcal{C})\big)+\mathsf{P}_{\mathsf{e}}(\mathcal{C})\cdot\log|\mathcal{C}|,\end{aligned} \qquad (37)$$

where $\psi_{\mathcal{C}}:\mathbb{R}^d\to\mathcal{C}$ is a function for decoding S_W from Y and $\mathsf{P}_{\mathsf{e}}(\mathcal{C})\triangleq\mathbb{P}\big(S_W\neq\psi_{\mathcal{C}}(Y)\big)$ is the probability that $\psi_{\mathcal{C}}$ commits an error.

Fano's inequality holds for any decoding function $\psi_{\mathcal{C}}$. We choose $\psi_{\mathcal{C}}$ as the maximum likelihood decoder, i.e., upon observing a $y\in\mathbb{R}^d$ it returns the closest point to y in $\mathcal{C}$. Denote by $\mathcal{D}_i\triangleq\psi_{\mathcal{C}}^{-1}(c_i)$ the decoding region on c_i, i.e., the region $\big\{y\in\mathbb{R}^d\big|\psi_{\mathcal{C}}(y)=c_i\big\}$ that $\psi_{\mathcal{C}}$ maps to c_i. Note that $\mathcal{D}_i=\mathsf{C}_i$ for all $i\in[k^d]$ for which C_i doesn't intersect with the boundary of $[-1,1]^d$. When $Y=S_W+Z$, $S_W\sim\mathsf{Unif}(\mathcal{C})$ and the probability of error for the decoder $\psi_{\mathcal{C}}$ is bounded as:

$$\begin{aligned}\mathsf{P}_{\mathsf{e}}(\mathcal{C})&=\frac{1}{k^d}\sum_{i=1}^{k^d}\mathbb{P}\Big(\psi_{\mathcal{C}}(c_i+Z)\neq c_i\Big|S_W=c_i\Big)\\&=\frac{1}{k^d}\sum_{i=1}^{k^d}\mathbb{P}\big(c_i+Z\notin\mathcal{D}_i\big)\\&\overset{(a)}{\leq}\mathbb{P}\left(\|Z\|_\infty>\frac{2/k}{2}\right)\end{aligned}$$

$$\overset{(b)}{=} 1 - \left(1 - 2Q\left(\frac{1}{k\beta}\right)\right)^{d}, \tag{38}$$

where (a) holds since the C_i have sides of length $2/k$ and the error probability is largest for $i \in [k^d]$ such that C_i is in the interior of $[-1,1]^d$. Step (b) follows from independence and the definition of the Q-function.

Taking $k = k_\star$ in (38) as given in the statement of the theorem gives the desired bound $\mathsf{P}_\mathsf{e}(\mathcal{C}) \leq \epsilon$. Collecting the pieces and inserting back to (36), we obtain

$$I(S^n; Y) \geq \log\left(\frac{k_\star^{d(1-\epsilon)}}{n}\right) - H_b(\epsilon). \tag{39}$$

Together with (35) this concludes the proof.

10.3. Proof of Theorem 6

Denote the joint distribution of (C, Z, V) by $P_{C,Z,V}$. Marginal or conditional distributions are denoted as usual by keeping only the relevant subscripts. Lowercase p is used to denote a PMF or a PDF depending on whether the random variable in the subscript is discrete or continuous. In particular, p_C is the PMF of C, $p_{C|V}$ is the conditional PMF of C given V, while $p_Z = \varphi_\beta$ and $p_V = g$ are the PDFs of Z and V, respectively.

First observe that the estimator is unbiased:

$$\mathbb{E}\hat{h}_{\mathsf{MC}} = -\frac{1}{n \cdot n_{\mathsf{MC}}} \sum_{i=1}^{n} \sum_{j=1}^{n_{\mathsf{MC}}} \mathbb{E} \log g\left(\mu_i + Z_j^{(i)}\right) = h(g). \tag{40}$$

Therefore, the MSE expands as

$$\mathsf{MSE}\left(\hat{h}_{\mathsf{MC}}\right) = \frac{1}{n^2 \cdot n_{\mathsf{MC}}} \sum_{i=1}^{n} \mathsf{var}\Big(\log g(\mu_i + Z)\Big). \tag{41}$$

We next bound the variance of $\log g(\mu_i + Z)$ via Poincaré inequality for the Gaussian measure $\mathcal{N}(0, \beta^2 \mathrm{I}_d)$ (with Poincaré constant β^2). For each $i \in [n]$, we have

$$\mathsf{var}\Big(\log g(\mu_i + Z)\Big) \leq \beta^2 \mathbb{E}\Big[\big\|\nabla \log g(\mu_i + Z)\big\|_2^2\Big]. \tag{42}$$

We proceed with separate derivations of (16) and (17).

10.3.1. MSE Bound for Bounded Support

Since $\|C\|_2 \leq \sqrt{d}$ almost surely, Proposition 3 from (Polyanskiy & Wu, 2016) implies

$$\big\|\nabla \log g(v)\big\|_2 \leq \frac{\|v\|_2 + \sqrt{d}}{\beta^2}. \tag{43}$$

Inserting this into the Poincaré inequality and using $(a+b)^2 \leq 2a^2 + 2b^2$ we have,

$$\mathsf{var}\Big(\log g(\mu_i + Z)\Big) \leq \frac{2d(4 + \beta^2)}{\beta^2}, \tag{44}$$

for each $i \in [n]$. Together with (41), this concludes the proof of (16).

10.3.2. MSE Bound for Bounded Second Moment

To prove (17), we use Proposition 2 from (Polyanskiy & Wu, 2016) to obtain

$$\big\|\nabla \log g(v)\big\|_2 \leq \frac{1}{\beta^2}\big(3\|v\|_2 + 4\mathbb{E}\|C\|_2\big). \tag{45}$$

Via the Poincaré inequality from (42), the variance is bounded as

$$\begin{aligned}
&\mathsf{var}\Big(\log g(\mu_i + Z)\Big) \\
&\leq \frac{1}{\beta^2}\mathbb{E}\Big[(3\|\mu_i + Z\|_2 + 4\mathbb{E}\|C\|)^2\Big] \\
&\leq \frac{1}{\beta^2}\Big(9d\beta^2 + 16M_C + 24\beta\sqrt{dM_C} \\
&\qquad + 3\|\mu_i\|_2\left(3 + 9\beta\sqrt{d} + 8\beta\sqrt{dM_C}\right)\Big),
\end{aligned} \tag{46}$$

where the last step uses Hölder's inequality (namely, $\mathbb{E}\|C\|_2 \leq \sqrt{\mathbb{E}\|C\|_2^2}$). The proof of (17) is concluded by plugging (46) into the MSE expression from (41) and noting that $\frac{1}{n}\sum_{i=1}^{n} \|\mu_i\|_2 \leq \sqrt{M_C}$.

References

Berrett, T. B., Samworth, R. J., and Yuan, M. Efficient multivariate entropy estimation via k-nearest neighbour distances. *Annals Stats.*, 47(1):288–318, 2019.

Chen, J. A general lower bound of minimax risk for absolute-error loss. *Canadian Journal of Statistics*, 25(4):545–558, Dec. 1997.

Cisse, M., Bojanowski, P., Grave, E., Dauphin, Y., and Usunier, N. Parseval networks: Improving robustness to adversarial examples. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2017.

Goldfeld, Z., Greenewald, K., Weed, J., and Polyanskiy, Y. Optimality of the plug-in estimator for differential entropy estimation under Gaussian convolutions. Paris, France, July 2019.

Haje, H. F. E. and Golubev, Y. On entropy estimation by m-spacing method. *Journal of Mathematical Sciences*, 163(3):290–309, Dec. 2009.

Hall, P. Limit theorems for sums of general functions of m-spacings. *Mathematical Proceedings of the Cambridge Philosophical Society*, 96(3):517–532, Nov. 1984.

Hall, P. and Morton, S. C. On the estimation of entropy. *Annals of the Institute of Statistical Mathematics*, 45(1): 69–88, Mar. 1993.

Han, Y., Jiao, J., Weissman, T., and Wu, Y. Optimal rates of entropy estimation over Lipschitz balls. arXiv:1711.02141 [math.ST], 2017.

Hsu, D., Kakade, S., and Zhang, T. A tail inequality for quadratic forms of subgaussian random vectors. *Electronic Communications in Probability*, 17, 2012.

Joe, H. Estimation of entropy and other functionals of a multivariate density. *Annals of the Institute of Statistical Mathematics*, 41(4):683–697, Dec. 1989.

Kandasamy, K., Krishnamurthy, A., Poczos, B., Wasserman, L., and Robins, J. M. Nonparametric von Mises estimators for entropies, divergences and mutual informations. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 397–405, 2015.

Levit, B. Y. Asymptotically efficient estimation of nonlinear functionals. *Problemy Peredachi Informatsii*, 14(3):65–72, 1978.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*, 2017.

Polyanskiy, Y. and Wu, Y. Wasserstein continuity of entropy and outer bounds for interference channels. *IEEE Transactions on Information Theory*, 62(7):3992–4002, Jul. 2016.

Robert, C. P. *Monte Carlo Methods*. Wiley Online Library, 2004.

Saxe, A. M., Bansal, Y., Dapello, J., Advani, M., Kolchinsky, A., Tracey, B. D., and Cox, D. D. On the information bottleneck theory of deep learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

Singh, S. and Póczos, B. Finite-sample analysis of fixed-k nearest neighbor density functional estimators. In *Advances in Neural Information Processing Systems*, pp. 1217–1225, 2016.

Sricharan, K., Raich, R., and Hero, A. O. Estimation of nonlinear functionals of densities with confidence. *IEEE Trans. Inf. Theory*, 58(7):4135–4159, Jul. 2012.

Tsybakov, A. B. and Van der Meulen, E. C. Root-n consistent estimators of entropy for densities with unbounded support. *Scandinavian Journal of Statistics*, pp. 75–83, Mar. 1996.

Villani, C. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2006.

Wu, Y. and Yang, P. Minimax rates of entropy estimation on large alphabets via best polynomial approximation. *IEEE Transactions on Information Theory*, 62(6):3702–3720, June 2016.