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.

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

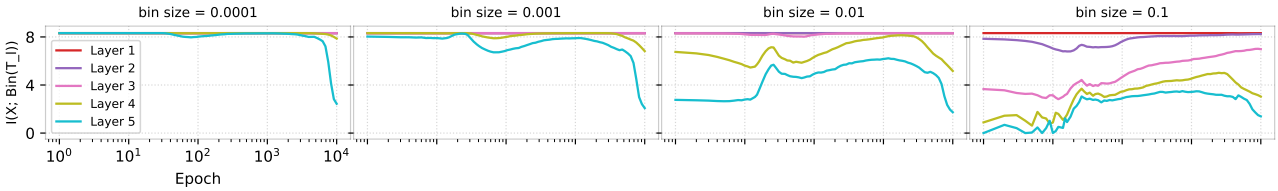


Figure 1. $I\big(X;\mathsf{Bin}(T_\ell)\big)$ vs. epochs for different bin sizes and the model in (Shwartz-Ziv & Tishby, 2017), where X is uniformly distributed over a 2^{12}-sized empirical dataset. The curves converge to $H(X) = \ln(2^{12}) \approx 8.3$ for small bins.

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-

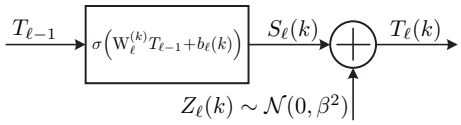


Figure 2. kth noisy neuron in layer ℓ: $\mathrm{W}_\ell^{(k)}$ and $b_\ell(k)$ are the kth row/entry of the weight matrix and the bias, respectively.

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 $[k:\ell] \triangleq \left\{i \in \mathbb{Z} \middle| k \leq i \leq \ell\right\}$ 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 $Z_\ell \sim \mathcal{N}\left(0, \beta^2 \mathrm{I}_{d_\ell}\right)$; 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).

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 $P_{X,T_1,\ldots,T_L} \triangleq P_X P_{T_1,\ldots,T_L|X}$, 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.

Table 1. MNIST validation errors for different models: mean $\pm$ 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

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 $x \in \mathcal{X}$.

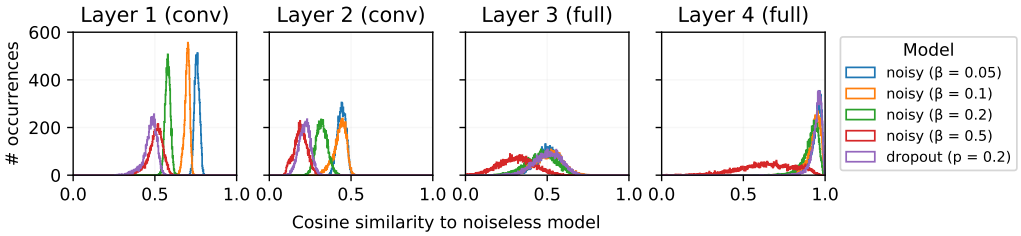


Figure 3. Cosine similarity histograms between internal representations of deterministic, noisy, and dropout MNIST CNNs.

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

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

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*

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

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.

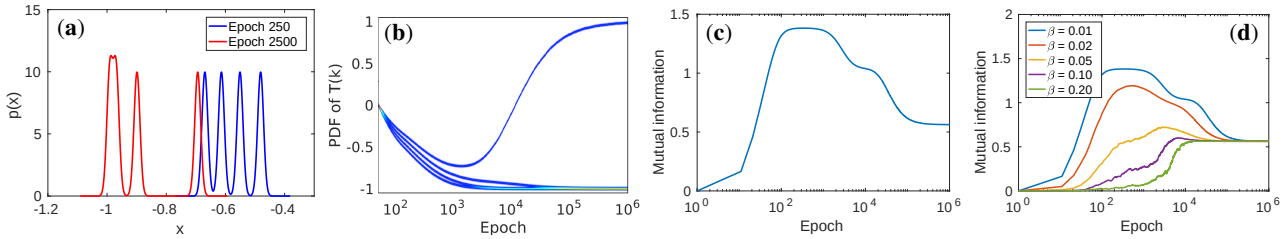


Figure 4. Single-layer tanh network: (a) the density $p_{T(k)}$ at epochs $k = 250, 2500$; (b) $p_{T(k)}$ and (c) $I\big(X;T(k)\big)$ as a function of k; and (d) mutual information as a function of k, for different β values..

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 (≈ 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

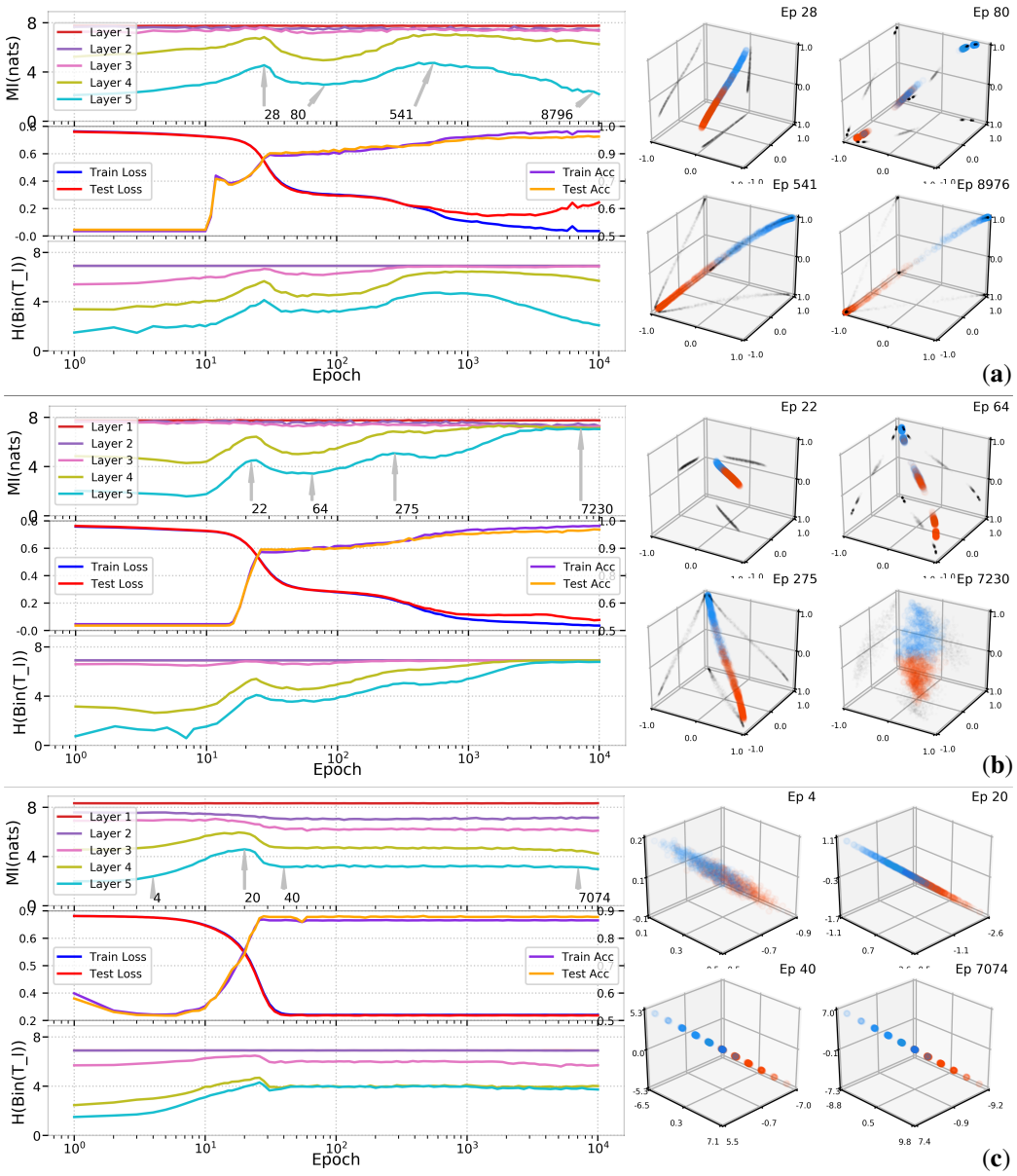
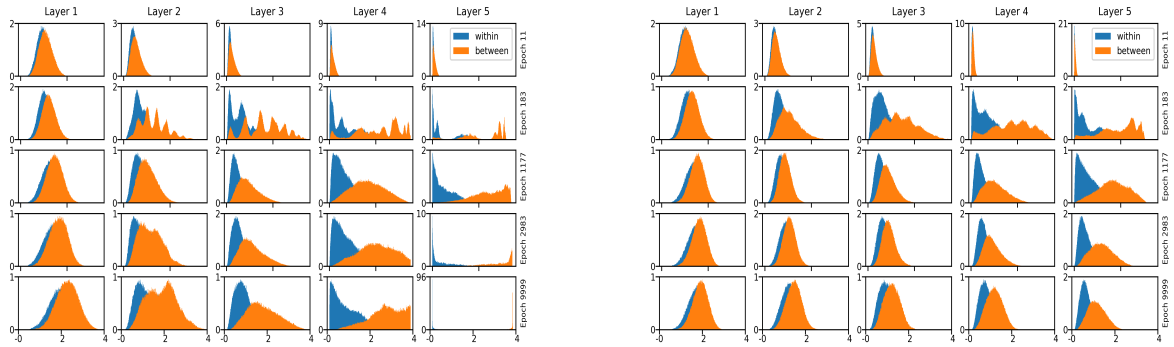


Figure 5. (a) Evolution of $I(X;T_\ell)$ and training/test losses across training epochs for the SZT model with $\beta = 0.005$ and tanh nonlinearities. The scatter plots show the values of Layer 5 ($d_5 = 3$) at the arrow-marked epochs on the mutual information plot. The bottom plot shows $H\big(\mathsf{Bin}(T_\ell)\big)$ 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-



(a) **(b)**

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

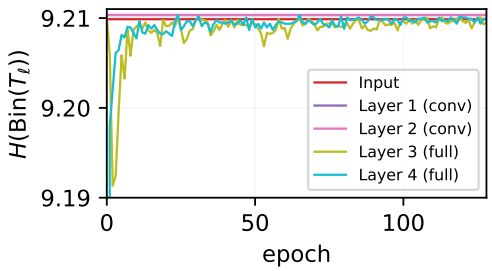


Figure 7. $H\big(\text{Bin}(T_\ell)\big)$ for the MNIST CNN, computed using two bins: $[-1, 0]$ and $(0, 1]$. The tiny range of the y axis shows the near injectivity of the model.

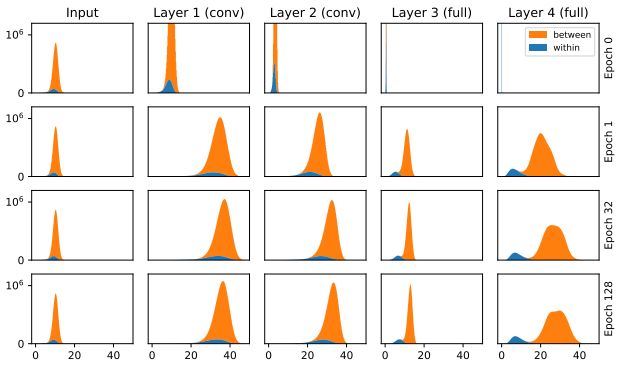


Figure 8. Histograms of within-class and between-class pairwise distances from the MNIST CNN.

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 $H\big(\text{Bin}(T_\ell)\big)$ 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, $H\big(\text{Bin}(T_\ell)\big)$ 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.