entropy

MDPI

Article

The Conditional Entropy Bottleneck

Ian Fischer

Google Research, Mountain View, CA 94043, USA; iansf@google.com

Received: 30 July 2020; Accepted: 28 August 2020; Published: 8 September 2020

check for updates

Abstract: Much of the field of Machine Learning exhibits a prominent set of failure modes, including vulnerability to adversarial examples, poor out-of-distribution (OoD) detection, miscalibration, and willingness to memorize random labelings of datasets. We characterize these as failures of *robust generalization*, which extends the traditional measure of generalization as accuracy or related metrics on a held-out set. We hypothesize that these failures to robustly generalize are due to the learning systems retaining *too much* information about the training data. To test this hypothesis, we propose the *Minimum Necessary Information* (MNI) criterion for evaluating the quality of a model. In order to train models that perform well with respect to the MNI criterion, we present a new objective function, the *Conditional Entropy Bottleneck* (CEB), which is closely related to the *Information Bottleneck* (IB). We experimentally test our hypothesis by comparing the performance of CEB models with deterministic models and Variational Information Bottleneck (VIB) models on a variety of different datasets and robustness challenges. We find strong empirical evidence supporting our hypothesis that MNI models improve on these problems of robust generalization.

Keywords: information theory; information bottleneck; machine learning

1. Introduction

Despite excellent progress in classical generalization (e.g., accuracy on a held-out set), the field of Machine Learning continues to struggle with the following issues:

- **Vulnerability to adversarial examples.** Most machine-learned systems are vulnerable to adversarial examples. Many defenses have been proposed, but few have demonstrated robustness against a powerful, general-purpose adversary. Many proposed defenses are ad-hoc and fail in the presence of a concerted attacker [1,2].
- **Poor out-of-distribution detection.** Most models do a poor job of signaling that they have received data that is substantially different from the data they were trained on. Even generative models can report that an entirely different dataset has higher likelihood than the dataset they were trained on [3]. Ideally, a trained model would give less confident predictions for data that was far from the training distribution (as well as for adversarial examples). Barring that, there would be a clear, principled statistic that could be extracted from the model to tell whether the model *should* have made a low-confidence prediction. Many different approaches to providing such a statistic have been proposed [4–9], but most seem to do poorly on what humans intuitively view as obviously different data.
- **Miscalibrated predictions.** Related to the issues above, classifiers tend to be overconfident in their predictions [4]. Miscalibration reduces confidence that a model's output is fair and trustworthy.
- **Overfitting to the training data.** Zhang et al. [10] demonstrated that classifiers can memorize fixed random labelings of training data, which means that it is possible to learn a classifier with perfect *inability* to generalize. This critical observation makes it clear that a fundamental test of generalization is that the model should *fail* to learn when given what we call *information-free* datasets.

We consider these to be problems of *robust generalization*, which we define and discuss in Section 2.1. In this work, we hypothesize that these problems of robust generalization all have a common cause: models retain *too much* information about the training data. We formalize this by introducing the *Minimum Necessary Information* (MNI) criterion for evaluating a learned representation (Section 2.2). We then introduce an objective function that directly optimizes the MNI, the *Conditional Entropy Bottleneck* (CEB) (Section 2.3) and compare it with the closely-related *Information Bottleneck* (IB) objective [11] in Section 2.5. In Section 2.6, we describe practical ways to optimize CEB in a variety of settings.

Finally, we give empirical evidence for the following claims:

- **Better classification accuracy.** MNI models can achieve superior accuracy on classification tasks than models that capture either more or less information than the minimum necessary information (Sections 3.1.1 and 3.1.6).
- **Improved robustness to adversarial examples.** Retaining excessive information about the training data results in vulnerability to a variety of whitebox and transfer adversarial examples. MNI models are substantially more robust to these attacks (Sections 3.1.2 and 3.1.6).
- **Strong out-of-distribution detection.** The CEB objective provides a useful metric for out-of-distribution (OoD) detection, and CEB models can detect OoD examples as well or better than non-MNI models (Section 3.1.3).
- **Better calibration.** MNI models are better calibrated than non-MNI models (Section 3.1.4).
- **No memorization of information-free datasets.** MNI models fail to learn in information-free settings, which we view as a minimum bar for demonstrating robust generalization (Section 3.1.5).

2. Materials and Methods

2.1. Robust Generalization

In classical generalization, we are interested in a model's performance on held-out data on some task of interest, such as classification accuracy. In *robust generalization*, we want: **(RG1)** *to maintain the model's performance in the classical generalization setting*; **(RG2)** *to ensure the model's performance in the presence of an adversary (unknown at training time)*; and **(RG3)** *to detect adversarial and non-adversarial data that strongly differ from the training distribution*.

Adversarial training approaches considered in the literature so far [12–14] violate **(RG1)**, as they typically result in substantial decreases in accuracy. Similarly, provable robustness approaches (e.g., Cohen et al. [15], Wong et al. [16]) provide guarantees for a particular adversary known at training time, also at a cost to test accuracy. To our knowledge, neither approaches provide any mechanism to satisfy **(RG3)**. On the other hand, approaches for detecting adversarial and non-adversarial out-of-distribution (OoD) examples [4–9] are either known to be vulnerable to adversarial attack [1,2], or do not demonstrate that the approach provides robustness against unknown adveraries, both of which violate **(RG2)**.

Training on information-free datasets [10] provides an additional way to check if a learning system is compatible with **(RG1)**, as memorization of such datasets necessarily results in maximally poor performance on any test set. Model calibration is not obviously a necessary condition for robust generalization, but if a model is well-calibrated on a held-out set, its confidence may provide some signal for distinguishing OoD examples, so we mention it as a relevant metric for **(RG3)**.

To our knowledge, the only works to date that have demonstrated progress on robust generalization for modern machine learning datasets are the *Variational Information Bottleneck* [17,18] (VIB), and *Information Dropout* [19]. Alemi et al. [17] presented preliminary results that VIB improves adversarial robustness on image classification tasks while maintaining high classification accuracy (**(RG1)** and **(RG2)**). Alemi et al. [18] showed that VIB models provide a useful signal, the *Rate*, R, for detecting OoD examples (**(RG3)**). Achille and Soatto [19] also showed preliminary results on adversarial robustness and demonstrated failure to train on information-free datasets.

In this work, we do not claim to "solve" robust generalization, but we do show notable improvement on all three conditions simply by changing the training objective. This evidence supports our core hypothesis that problems of robust generalization are caused in part by retaining too much information about the training data.

2.2. The Minimum Necessary Information

We define the *Minimum Necessary Information* (MNI) criterion for a learned representation in three parts:

- **Information.** We would like a representation Z that captures useful information about a dataset (X, Y). Entropy is the unique measure of information [20], so the criterion prefers information-theoretic approaches. (We assume familiarity with the mutual information and its relationships to entropy and conditional entropy: $I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$ [21] (p. 20).)
- **Necessity.** The semantic value of information is given by a *task*, which is specified by the set of variables in the dataset. Here we will assume that the task of interest is to predict Y given X, as in any supervised learning dataset. The information we capture in our representation Z must be necessary to solve this task. As a variable X may have *redundant* information that is useful for predicting Y, a representation Z that captures the necessary information may not be minimal or unique (the MNI criterion does not require uniqueness of Z).
- **Minimality.** Given all representations $\mathcal{Z}$ that can solve the task, we require one that retains the smallest amount of information about the task: $\inf_{Z \in \mathcal{Z}} I(Z; X, Y)$.

Necessity can be defined as $I(X;Y) \leq I(Y;Z)$. Any less information than that would prevent Z from solving the task of predicting Y from X. *Minimality* can be defined as $I(X;Y) \geq I(X;Z)$. Any more information than that would result in Z capturing information from X that is either redundant or irrelevant for predicting Y. Since the information captured by Z is constrained from above and below, we have the following necessary and sufficient conditions for perfectly achieving the Minimum Necessary Information, which we call the *MNI Point*:

$$I(X;Y) = I(X;Z) = I(Y;Z) \tag{1}$$

The MNI point defines a unique point in the information plane. The geometry of the information plane can be seen in Figure 1. The MNI criterion does not make any Markov assumptions on the models or algorithms that learn the representations. However, the algorithms we discuss here all do rely on the standard Markov chain $Z \leftarrow X \leftrightarrow Y$. See Fischer [22] for an example of an objective that doesn't rely on a Markov chain during training.

A closely related concept to Necessity is called *sufficiency* by Achille and Soatto [19] and other authors. We avoid the term due to potential confusion with minimum sufficient statistics, which maintain the mutual information between a model and the data it generates [21] (p. 35). The primary difference between necessity and sufficiency is the reliance on the Markov constraint to define sufficiency. Ref. [19] also does not identify the MNI point as an idealized target, instead defining the optimization problem: minimize $I(X;Z)$ s.t. $H(Y|Z) = H(Y|X)$.

In general it may not be possible to satisfy Equation (1). As discussed in Anantharam et al. [23–25], for any given dataset (X, Y), there is some maximum value for any possible representation Z:

$$1 \geq \eta_{\mathrm{KL}} = \sup_{Z \leftarrow X \rightarrow Y} \frac{I(Y;Z)}{I(X;Z)} \tag{2}$$

with equality only when $X \rightarrow Y$ is a *deterministic* map. Training datasets are often deterministic in one direction or the other. e.g., common image datasets map each distinct image to a single label.

Thus, in practice, we can often get very close to the MNI on the training set given a sufficiently powerful model.

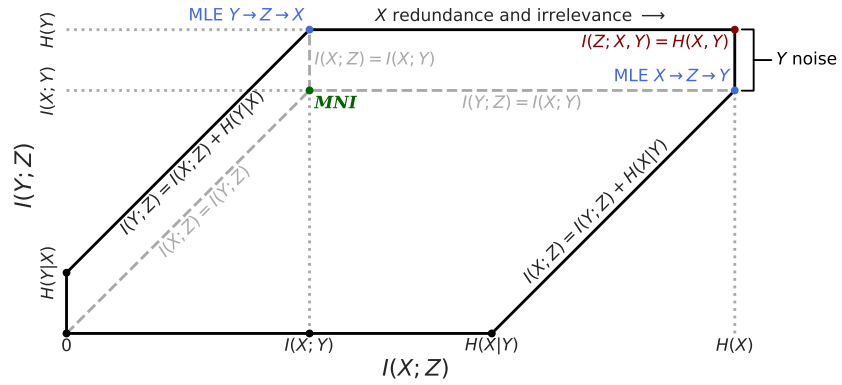

Figure 1. Geometry of the feasible regions in the $(I(X;Z), I(Y;Z))$ information plane for for any algorithm, with key points and edges labeled. The **black** edges bound the feasible region for an (X,Y) pair where $H(X|Y) > I(X;Y) > H(Y|X)$, which would generally be the case in an image classification task, for example. The **gray** dashed lines bound the feasible regions when the underlying model depends on a Markov chain. The $I(X;Z) = I(Y;Z)$ and $I(Y;Z) = I(X;Y)$ lines are the upper bound for $Z \leftarrow X \leftrightarrow Y$. The $I(X;Z) = I(Y;Z)$ and $I(X;Z) = I(X;Y)$ lines are the right bound for $Z \leftarrow Y \leftrightarrow X$. The **blue** points correspond to the best possible Maximum Likelihood Estimates (MLE) for the corresponding Markov chain models. The **red** point corresponds to the maximum information Z could ever capture about (X,Y). The Minimum Necessary Information (MNI) point is **green**. As $I(X;Z)$ increases, Z captures more information that is either redundant or irrelevant with respect to predicting Y. Similarly, any variation in Y that remains once we know X is just noise as far as the task is concerned. The MNI point is the unique point that has no redundant or irrelevant information from X, and everything but the noise from Y.

2.2.1. MNI and Robust Generalization

To satisfy **(RG1)** (classical generalization), a model must have $I(X;Z) \geq I(X;Y) = I(Y;Z)$ on the *test* dataset. Shamir et al. [26] show that $|I(X;Z) - \hat{I}(X;Z)| \approx O\left(\frac{2^{\hat{I}(X;Z)}}{\sqrt{N}}\right)$, where $\hat{I}(\cdot)$ indicates the training set information and N is the size of the training set. More recently, Bassily et al. [27] gave a similar result in a PAC setting. Both results indicate that models that are *compressed on the training data* should do *better at generalizing* to similar test data.

Less clear is how an MNI model might improve on **(RG2)** (adversarial robustness). In this work, we treat it as a hypothesis that we investigate empirically rather than theoretically. The intuition behind the hypothesis can be described in terms of the idea of *robust* and *non-robust features* from Ilyas et al. [28]: non-robust features in X should be compressed as much as possible when we learn Z, whereas robust features should be retained as much as is necessary. If Equation (1) is satisfied, Z must have "scaled" the importance of the the features in X according to their importance for predicting Y. Consequently, an attacker that tries to take advantage of a non-robust feature will have to change it much more in order to confuse the model, possibly exceeding the constraints of the attack before it succeeds.

For **(RG3)** (detection), the MNI criterion does not directly apply, as that will be a property of specific modeling choices. However, if the model provides an accurate way to measure $I(X = x; Z = z)$ for a particular pair (x,z), Alemi et al. [18] suggests that can be a valuable signal for OoD detection.

2.3. The Conditional Entropy Bottleneck

We would like to learn a representation Z of X that will be useful for predicting Y. We can represent this problem setting with the Markov chain $Z \leftarrow X \leftrightarrow Y$. We would like Z to satisfy Equation (1). Given the conditional independence $Z \perp\!\!\!\perp Y|X$ in our Markov chain, $I(Y;Z) \leq I(X;Y)$, by the data processing inequality. Thus, maximizing $I(Y;Z)$ is consistent with the MNI criterion.

However, $I(X;Z)$ does not clearly have a constraint that targets $I(X;Y)$, as $0 \leq I(X;Z) \leq H(X)$. Instead, we can notice the following identities at the MNI point:

$$I(X;Y|Z) = I(X;Z|Y) = I(Y;Z|X) = 0 \tag{3}$$

The conditional mutual information is always non-negative, so learning a compressed representation Z of X is equivalent to minimizing $I(X;Z|Y)$. Using our Markov chain and the chain rule of mutual information [21]:

$$I(X;Z|Y) = I(X,Y;Z) - I(Y;Z) = I(X;Z) - I(Y;Z) \tag{4}$$

This leads us to the general *Conditional Entropy Bottleneck*:

$$\begin{aligned} \text{CEB} &\equiv \min_Z I(X;Z|Y) - \gamma I(Y;Z) && (5)\\ &= \min_Z H(Z) - H(Z|X) - H(Z) + H(Z|Y) \\ &\qquad - \gamma(H(Y) + H(Y|Z)) && (6)\\ &\Leftrightarrow \min_Z -H(Z|X) + H(Z|Y) + \gamma H(Y|Z) && (7) \end{aligned}$$

In line 7, we can optionally drop $H(Y)$ because it is constant with respect to Z. Here, any $\gamma > 0$ is valid, but for deterministic datasets (Section 2.2), $\gamma = 1$ will achieve the MNI for a sufficiently powerful model. Further, we should expect $\gamma = 1$ to yield *consistent* models and other values of γ not to: since $I(Y;Z)$ shows up in two forms in the objective, weighing them differently forces the optimization procedure to count bits of $I(Y;Z)$ in two different ways, potentially leading to a situation where $H(Z) - H(Z|Y) \neq H(Y) - H(Y|Z)$ at convergence. Given knowledge of those four entropies, we can define a consistency metric for Z:

$$C_{I(Y;Z)}(Z) \equiv |H(Z) - H(Z|Y) - H(Y) + H(Y|Z)| \tag{8}$$

2.4. Variational Bound on CEB

We will variationally upper bound the first term of Equation (5) and lower bound the second term using three distributions: $e(z|x)$, the *encoder* which defines the joint distribution we will use for sampling, $p(x,y,z) \equiv p(x,y)e(z|x)$; $b(z|y)$, the *backward encoder*, an approximation of $p(z|y)$; and $c(y|z)$, the *classifier*, an approximation of $p(y|z)$ (the name is arbitrary, as Y may not be labels). All of $e(\cdot)$, $b(\cdot)$, and $c(\cdot)$ may have learned parameters, just like the encoder and decoder of a VAE [29], or the encoder, classifier, and marginal in VIB.

In the following, we write expectations $\langle \log e(z|x) \rangle$. They are always with respect to the joint distribution; here, that is $p(x,y,z) \equiv p(x,y)e(z|x)$. The first term of Equation (5):

$$\begin{aligned} I(X;Z|Y) &= -H(Z|X) + H(Z|Y) = \langle \log e(z|x) \rangle - \langle \log p(z|y) \rangle && (9)\\ &= \langle \log e(z|x) \rangle - \langle \log b(z|y) \rangle - \langle \text{KL}[p(z|y)||b(z|y)] \rangle && (10)\\ &\leq \langle \log e(z|x) \rangle - \langle \log b(z|y) \rangle && (11) \end{aligned}$$

The second term of Equation (5):

$$\begin{aligned} I(Y;Z) &= H(Y) - H(Y|Z) \propto -H(Y|Z) = \langle \log p(y|z) \rangle && (12)\\ &= \langle \log c(y|z) \rangle + \langle \text{KL}[p(y|z)||c(y|z)] \rangle && (13)\\ &\geq \langle \log c(y|z) \rangle && (14) \end{aligned}$$

These variational bounds give us a tractable objective function for amortized inference, the *Variational Conditional Entropy Bottleneck* (VCEB):

$$\text{VCEB} \equiv \min_{e,b,c} \langle \log e(z|x) \rangle - \langle \log b(z|y) \rangle - \gamma \langle \log c(y|z) \rangle \tag{15}$$

There are a number of other ways to optimize Equation (5). We describe a few of them in Section 2.6 and Appendices B and C.

2.5. Comparison to the Information Bottleneck

The Information Bottleneck (IB) [11] learns a representation Z from X subject to a soft constraint:

$$IB \equiv \min_{Z} I(X;Z) - \beta I(Y;Z) \tag{16}$$

where β^{-1} controls the strength of the constraint. As $\beta \to \infty$, IB recovers the standard cross-entropy loss.

In Figure 2 we show information diagrams comparing which regions IB and CEB maximize and minimize. See Yeung [30] for a theoretical explanation of information diagrams.CEB avoids trying to both minimize and maximize the central region at the same time. In Figure 3 we show the feasible regions for CEB and IB, labeling the MNI point on both. CEB's rectification of the information plane means that we can always measure in absolute terms how much more we could compress our representation *at the same predictive performance*: $I(X;Z|Y) \geq 0$. For IB, it is not possible to tell *a priori* how far we are from optimal compression.

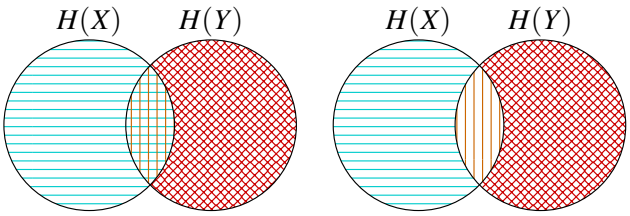

Figure 2. Information diagrams showing how IB and CEB maximize and minimize different regions. ⊗ regions inaccessible to the objective due to the Markov chain $Z \leftarrow X \leftrightarrow Y$. ◍ regions being maximized by the objective ($I(Y;Z)$ in both cases). ⊖ regions being minimized by the objective. **IB** minimizes the intersection between Z and both $H(X|Y)$ and $I(X;Y)$. **CEB** only minimizes the intersection between Z and $H(X|Y)$.

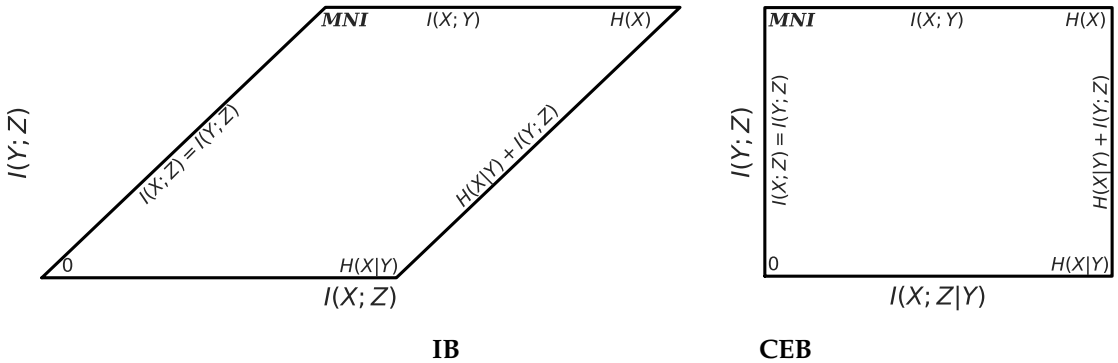

Figure 3. Geometry of the feasible regions for IB and CEB, with all points labeled. CEB rectifies IB's parallelogram by subtracting $I(Y;Z)$ at every point. Everything outside of the black lines is unattainable by any model on any dataset. Compare the IB feasible region to the dashed region in Figure 1.

From Equations (4), (5) and (16), it is clear that CEB and IB are equivalent for $\gamma = \beta - 1$. To simplify comparison of the two objectives, we can parameterize them with:

$$\rho = \log \gamma = \log(\beta - 1) \tag{17}$$

Under this parameterization, for deterministic datasets, sufficiently powerful models will target the MNI point at $\rho = 0$. As ρ increases, more information is captured by the model. $\rho < 0$ *may* capture less than the MNI. $\rho > 0$ *may* capture more than the MNI.

2.5.1. Amortized IB

As described in Tishby et al. [11], IB is a tabular method, so it is not usable for amortized inference. The tabular optimization procedure used for IB trivially applies to CEB, just by setting $\beta = \gamma + 1$. Two recent works have extended IB for amortized inference. Achille and Soatto [19] presents *InfoDropout*, which uses IB to motivate a variation on Dropout [31]. Alemi et al. [17] presents the *Variational Information Bottleneck* (VIB):

$$VIB \equiv \langle \log e(z|x) \rangle - \langle \log m(z) \rangle) - \beta \langle \log c(y|z) \rangle \tag{18}$$

Instead of the backward encoder, VIB has a *marginal posterior*, $m(z)$, which is a variational approximation to $e(z) = \int dx\, p(x) e(z|x)$.

Following Alemi et al. [32], we define the *Rate* (R):

$$R \equiv \langle \log e(z|x) \rangle - \langle \log m(z) \rangle \geq I(X;Z) \tag{19}$$

We similarly define the *Residual Information* (Re_X):

$$Re_X \equiv \langle \log e(z|x) \rangle - \langle \log b(z|y) \rangle \geq I(X;Z|Y) \tag{20}$$

During optimization, observing R does not tell us how tightly we are adhering to the MNI. However, observing Re_X tells us exactly how many bits we are from the MNI point, assuming that our current classifier is optimal.

For convenience, define $CEB_x \equiv CEB_{\rho=x}$, and likewise for VIB. We can compare variational CEB with VIB by taking their difference at $\rho = 0$:

$$CEB_0 - VIB_0 = \langle \log b(z|y) \rangle - \langle \log m(z) \rangle \tag{21}$$

$$- \langle \log c(y|z) \rangle + \langle \log p(y) \rangle \tag{22}$$

Solving for $m(z)$ when that difference is 0:

$$m(z) = \frac{b(z|y)p(y)}{c(y|z)} \tag{23}$$

Since the optimal $m^*(z)$ is the marginalization of $e(z|x)$, at convergence we must have:

$$m^*(z) = \int dx\, p(x) e(z|x) = \frac{p(z|y)p(y)}{p(y|z)} \tag{24}$$

This solution may be difficult to find, as $m(z)$ only gets information about y indirectly through $e(z|x)$. For otherwise equivalent models, we may expect VIB_0 to converge to a looser approximation of $I(X;Z) = I(Y;Z) = I(X;Y)$ than CEB. Since VIB optimizes an upper bound on $I(X;Z)$, VIB_0 will report R converging to $I(X;Y)$, but may capture less than the MNI. In contrast, if Re_X converges to 0, the variational tightness of $b(z|y)$ to the optimal $p(z|y)$ depends only on the tightness of $c(y|z)$ to the optimal $p(y|z)$.

2.6. Model Variants

We introduce some variants on the basic variational CEB classification model that we will use in Section 3.1.6.

2.6.1. Bidirectional CEB

We can learn a shared representation Z that can be used to predict both X and Y with the following bidirectional CEB model: $Z_X \leftarrow X \leftrightarrow Y \rightarrow Z_Y$. This corresponds to the following joint: $p(x, y, z_X, z_Y) \equiv p(x, y)e(z_X|x)b(z_Y|y)$. The main CEB objective can then be applied in both directions:

$$\begin{aligned} \text{CEB}_{\text{bidir}} \equiv \min \; & -H(Z_X|X) + H(Z_X|Y) + \gamma_X H(Y|Z_X) \\ & - H(Z_Y|Y) + H(Z_Y|X) + \gamma_Y H(X|Z_Y) \end{aligned} \tag{25}$$

For the two latent representations to be useful, we want them to be consistent with each other (minimally, they must have the same parametric form). Fortunately, that consistency is trivial to encourage by making the natural variational substitutions: $p(z_Y|x) \rightarrow e(z_Y|x)$ and $p(z_X|y) \rightarrow b(z_X|y)$. This gives variational $\text{CEB}_{\text{bidir}}$:

$$\begin{aligned} \min \; & \langle \log e(z_X|x) \rangle - \langle \log b(z_X|y) \rangle - \gamma_X \langle \log c(y|z_X) \rangle \\ & + \langle \log b(z_Y|y) \rangle - \langle \log e(z_Y|x) \rangle - \gamma_Y \langle \log d(x|z_Y) \rangle \end{aligned} \tag{26}$$

where $d(x|z)$ is a *decoder* distribution. At convergence, we learn a unified Z that is consistent with both Z_X and Z_Y, permitting generation of either output given either input in the trained model, in the same spirit as Vedantam et al. [33], but without needing to train a joint encoder $q(z|x, y)$.

2.6.2. Consistent Classifier

We can reuse the backwards encoder as a classifier: $c(y|z) \propto b(z|y)p(y)$. We refer to this as the *Consistent Classifier*: $c(y|z) \equiv \text{softmax}\, b(z|y)p(y)$. If the labels are uniformly distributed, the $p(y)$ factor can be dropped; otherwise, it suffices to use the empirical $p(y)$. Using the consistent classifier for classification problems results in a model that only needs parameters for the two encoders, $e(z|x)$ and $b(z|y)$. This classifier differs from the more common *maximum a posteriori* (MAP) classifier because $b(z|y)$ is not the sampling distribution of either Z or Y.

2.6.3. CatGen Decoder

We can further generalize the idea of the consistent classifier to arbitrary prediction tasks by relaxing the requirement that we perfectly marginalize Y in the softmax. Instead, we can marginalize Y over any minibatch of size K we see at training time, under an assumption of a uniform distribution over the training examples we sampled:

$$p(y|z) = \frac{p(z|y)p(y)}{\int dy'\, p(z|y')p(y')} \tag{27}$$

$$\approx \frac{p(z|y)\frac{1}{K}}{\sum_{k=1}^{K} p(z|y_k)\frac{1}{K}} = \frac{p(z|y)}{\sum_{k=1}^{K} p(z|y_k)} \tag{28}$$

$$\approx \frac{b(z|y)}{\sum_{k=1}^{K} b(z|y_k)} \equiv c(y|z) \tag{29}$$

We can immediately see that this definition of $c(y|z)$ gives a valid distribution, as it is just a softmax over the minibatch. That means it can be directly used in the original objective without violating the variational bound. We call this decoder *CatGen*, for *Categorical Generative Model* because it

can trivially "generate" Y: the softmax defines a categorical distribution over the batch; sampling from it gives indices of $Y = y_j$ that most closely correspond to $Z = z_i$.

Maximizing $I(Y;Z)$ in this manner is a universal task, in that it can be applied to any paired data X, Y. This includes images and labels – the CatGen model may be used in place of both $c(y|z_X)$ and $d(x|z_Y)$ in the $\text{CEB}_{\text{bidir}}$ model (using $e(z|x)$ for $d(x|z_Y)$). This avoids a common concern when dealing with multivariate predictions: if predicting X is disproportionately harder than predicting Y, it can be difficult to balance the model [33,34]. For CatGen models, predicting X is never any harder than predicting Y, since in both cases we are just trying to choose the correct example out of K possibilities.

It turns out that CatGen is mathematically equivalent to *Contrastive Predictive Coding* (CPC) [35] after an offset of $\log K$. We can see this using the proof from Poole et al. [36], and substituting $\log b(z|y)$ for $f(y,z)$:

$$I(Y;Z) \geq \frac{1}{K}\sum_{k=1}^{K} \mathbb{E}_{\prod_j y_k, z\sim p(y_j)p(x_k|y_k)e(z|x_k)}\left[\log \frac{e^{f(y_k,z)}}{\frac{1}{K}\sum_{i=1}^{K} e^{f(y_i,z)}}\right] \tag{30}$$

$$= \frac{1}{K}\sum_{k=1}^{K} \mathbb{E}_{\prod_j y_k, z\sim p(y_j)p(x_k|y_k)e(z|x_k)}\left[\log \frac{b(z|y_k)}{\frac{1}{K}\sum_{i=1}^{K} b(z|y_i)}\right] \tag{31}$$

The advantage of the CatGen approach over CPC in the CEB setting is that we already have parameterized the forward and backward encoders to compute $I(X;Z|Y)$, so we don't need to introduce any new parameters when using CatGen to maximize the $I(Y;Z)$ term.

As with CPC, the CatGen bound is constrained by $\log K$, but when targeting the MNI, it is more likely that we can train with $\log K \geq I(X;Y)$. This is trivially the case for the datasets we explore here, where $I(X;Y) \leq \log 10$. It is also practical for larger datasets like ImageNet, where models are routinely trained with batch sizes in the thousands (e.g., Goyal et al. [37]), and $I(X;Y) \leq \log 1000$.

3. Results

We evaluate deterministic, VIB, and CEB models on Fashion MNIST [38] and CIFAR10 [39]. Our experiments focus on comparing the performance of otherwise *identical* models when we change only the objective function and vary ρ. Thus, we are interested in relative differences in performance that can be directly attributed to the difference in objective and ρ. These experiments cover the three aspects of *Robust Generalization* (Section 2.1): **(RG1)** (classical generalization) in Sections 3.1 and 3.1.6; **(RG2)** (adversarial robustness) in Sections 3.1 and 3.1.6; and **(RG3)** (detection) in Section 3.1.

3.1. (RG1), (RG2), and (RG3): Fashion MNIST

Fashion MNIST [38] is an interesting dataset in that it is visually complex and challenging, but small enough to train in a reasonable amount of time. We trained 60 different models on Fashion MNIST, four each for the following 15 types: a deterministic model (*Determ*); seven VIB models (VIB_{-1}, ..., VIB_5); and seven CEB models (CEB_{-1}, ..., CEB_5). Subscripts indicate ρ. All 60 models share the same inference architecture and are trained with otherwise identical hyperparameters. See Appendix A for details.

3.1.1. (RG1): Accuracy and Compression

In Figure 4 we see that both VIB and CEB have improved accuracy over the deterministic baseline, consistent with compressed representations generalizing better. Also, CEB outperforms VIB at every ρ, which we can attribute to the tighter variational bound given by minimizing Re_X rather than R. In the case of a simple classification problem with a uniform distribution over classes in the training set (like Fashion MNIST), we can directly compute $I(X;Y) = \log C$, where C is the number of classes. In order to compare the relative complexity of the learned representations for the VIB and CEB models, in the second panel of Figure 4 we show the maximum *rate lower bound* seen during training:

$R_X \equiv \left\langle \log \frac{e(z|x)}{\frac{1}{K}\sum_k^K e(z|x_k)} \right\rangle \leq I(X;Z)$ using the encoder's minibatch marginal for both VIB and CEB. This lower bound on $I(X;Z)$ is the "InfoNCE with a tractable encoder" bound from Poole et al. [36]. The two sets of models show nearly the same R_X at each value of ρ. Both models converge to exactly $I(X;Y) = \log 10 \approx 2.3$ nats at $\rho = 0$, as predicted by the derivation of CEB.

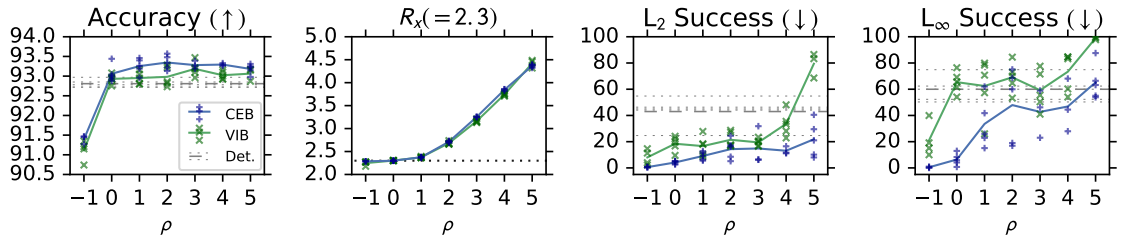

Figure 4. Test accuracy, maximum rate lower bound $R_X \leq I(Z;X)$ seen during training, and robustness to targeted PGD L_2 and L_∞ attacks on CEB, VIB, and Deterministic models trained on Fashion MNIST. At every ρ the CEB models outperform the VIB models on both accuracy and robustness, while having essentially identical maximum rates. *None of these models is adversarially trained.*

3.1.2. (RG2): Adversarial Robustness

The bottom two panels of Figure 4 show robustness to targeted *Projected Gradient Descent* (PGD) L_2 and L_∞ attacks [14]. All of the attacks are targeting the *trouser* class of Fashion MNIST, as that is the most distinctive class. Targeting a less distinctive class, such as one of the shirt classes, would confuse the difficulty of classifying the different shirts and the robustness of the model to adversaries. To measure robustness to the targeted attacks, we count the number of predictions that changed from a correct prediction on the clean image to an incorrect prediction of the target class on the adversarial image, and divide by the original number of correct predictions. Consistent with testing **(RG2)**, these adversaries are completely unknown to the models at training time – none of these models see any adversarial examples during training. CEB again outperforms VIB at every ρ, and the deterministic baseline at all but the least-compressed model ($\rho = 5$). We also see for both models that as ρ decreases, the robustness to both attacks increases, indicating that more compressed models are more robust.

Consistent with the MNI hypothesis, at $\rho = 0$ we end up with CEB models that have hit exactly 2.3 nats for the rate lower bound, have maintained high accuracy, and have strong robustness to both attacks. Moving to $\rho = -1$ gives only a small improvement to robustness, at the cost of a large decrease in accuracy.

3.1.3. (RG3): Out-of-Distribution Detection

We compare the ability of Determ, CEB_0, VIB_0, and VIB_4 to detect four different out-of-distribution (OoD) detection datasets. $U(0,1)$ is uniform noise in the image domain. MNIST uses the MNIST test set. Vertical Flip is the most challenging, using vertically flipped Fashion MNIST test images, as originally proposed in Alemi et al. [18]. CW is the Carlini-Wagner L_2 attack [40] at the default settings found in Papernot et al. [41], and additionally includes the adversarial attack success rate against each model.

We use two different metrics for thresholding, proposed in Alemi et al. [18]. H is the classifier entropy. R is the rate, defined in Section 2.5. These two threshold scores are used with the standard suite of proper scoring rules [42]: *False Positive Rate at 95% True Positive Rate* (FPR 95% TPR), *Area Under the ROC Curve* (AUROC), and *Area Under the Precision-Recall Curve* (AUPR).

Table 1 shows that using R to detect OoD examples can be much more effective than using classifier-based approaches. The deterministic baseline model is far weaker at detection using H than either of the high-performing stochastic models (CEB_0 and VIB_4). Those models both saturate detection performance, providing reliable signals for all four OoD datasets. However, as VIB_0 demonstrates,

simply having R available as a signal does not guarantee good detection. As we saw above, the VIB_0 models had noticeably worse classification performance, indicating that they had not achieved the MNI point: $I(Y;Z) < I(X;Z)$ for those models. These results indicate that for detection, violating the MNI criterion by having $I(X;Z) > I(X;Y)$ may not be harmful, but violating the criterion in the opposite direction *is* harmful.

Table 1. Results for out-of-distribution detection (*OoD*). *Thrsh.* is the threshold score used: H is the entropy of the classifier; R is the rate. Determ cannot compute R, so only H is shown. For VIB and CEB models, H is always inferior to R, similar to findings in Alemi et al. [18], so we omit it. *Adv. Success* is attack success of the CW adversary (bottom four rows). Arrows denote whether higher or lower scores are better. **Bold** indicates the best score in that column for that OoD dataset.

OoD	Model	Thrsh.	FPR @ 95% TPR ↓	AUROC ↑	AUPR In ↑	Adv. Success ↓
U(0,1)	Determ	H	35.8	93.5	97.1	N/A
	VIB_4	R	**0.0**	**100.0**	**100.0**	N/A
	VIB_0	R	80.6	57.1	51.4	N/A
	CEB_0	R	**0.0**	**100.0**	**100.0**	N/A
MNIST	Determ	H	59.0	88.4	90.0	N/A
	VIB_4	R	**0.0**	**100.0**	**100.0**	N/A
	VIB_0	R	12.3	66.7	91.1	N/A
	CEB_0	R	**0.1**	94.4	**99.9**	N/A
Vertical Flip	Determ	H	66.8	88.6	90.2	N/A
	VIB_4	R	**0.0**	**100.0**	**100.0**	N/A
	VIB_0	R	17.3	52.7	91.3	N/A
	CEB_0	R	**0.0**	90.7	**100.0**	N/A
CW	Determ	H	15.4	90.7	86.0	100.0%
	VIB_4	R	**0.0**	**100.0**	**100.0**	55.2%
	VIB_0	R	**0.0**	98.7	**100.0**	**35.8%**
	CEB_0	R	**0.0**	**99.7**	**100.0**	**35.8%**

3.1.4. (RG3): Calibration

A *well-calibrated* model is correct half of the time it gives a confidence of 50% for its prediction. In Figure 5, we show calibration plots at various points during training for four models. Calibration curves help analyze whether models are underconfident or overconfident. Each point in the plots corresponds to a 5% confidence bin. Accuracy is averaged for each bin. All four networks move from under- to overconfidence during training. However, CEB_0 and VIB_0 end up only slightly overconfident, while $\rho = 2$ is already sufficient to make VIB and CEB (not shown) nearly as overconfident as the deterministic model.

3.1.5. (RG1): Overfitting Experiments

We replicate the basic experiment from Zhang et al. [10] by using the images from Fashion MNIST, but replacing the training labels with fixed random labels. This dataset is *information-free* because $I(X;Y) = 0$. We use that dataset to train multiple deterministic models, as well as CEB and VIB models at ρ from 0 through 7. We find that the CEB and VIB models with $\rho < 6$ *never* learn, even after 100 epochs of training, but the deterministic models *always* learn. After about 40 epochs of training they begin to memorize the random labels, indicating severe overfitting and a perfect *failure* to generalize.

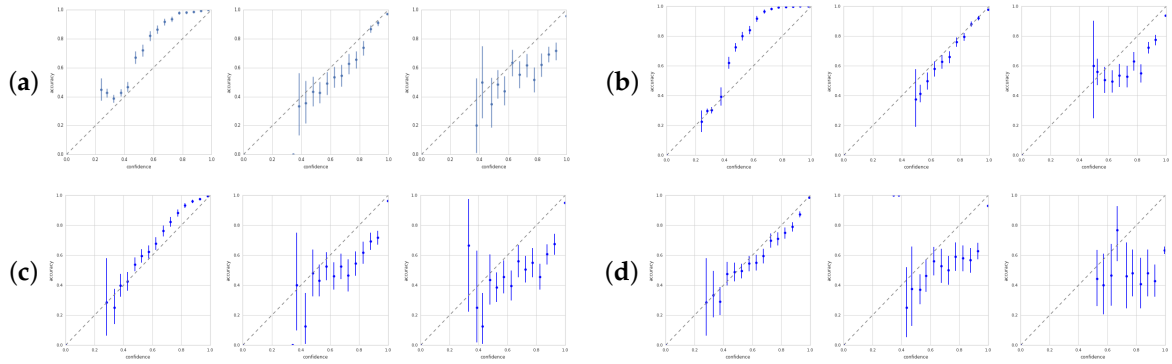

Figure 5. Calibration plots with 90% confidence intervals for four of the models after 2000 steps, 20,000 steps, and 40,000 steps (left, center, and right of each trio): (**a**) is CEB_0; (**b**) is VIB_0; (**c**) is VIB_2; (**d**) is Determ. *Perfect calibration* corresponds to the dashed diagonal lines. *Underconfidence* occurs when the points are above the diagonal. *Overconfidence* is below the diagonal. The $\rho = 0$ models are nearly perfectly calibrated still at 20,000 steps, but even at $\rho = 2$, the VIB model is almost as overconfident as Determ.

3.1.6. (RG1) and (RG2): CIFAR10 Experiments

For CIFAR10 [39] we trained the largest Wide ResNet [43] we could fit on a single GPU with a batch size of 250. This was a 62×7 model trained using AutoAugment [44]. We trained 3 CatGen $\text{CEB}_{\text{bidir}}$ models each of CEB_0 and CEB_5 and then selected the two models with the highest test accuracy for the adversarial robustness experiments. We evaluated the CatGen models using the consistent classifier, since CatGen models only train $e(z|x)$ and $b(z|y)$. CEB_0 reached **97.51%** accuracy. This result is better than the 28×10 Wide ResNet from AutoAugment by 0.19 percentage points, although it is still worse than the Shake-Drop model from that paper. We additionally tested the model on the CIFAR-10.1 test set [45], getting accuracy of 93.6%. This is a gap of only **3.9** percentage points, which is better than all of the results reported in that paper, and substantially better than the Wide ResNet results (but still inferior to the Shake-Drop AutoAugment results). The CEB_5 model reached 97.06% accuracy on the normal test set and 91.9% on the CIFAR-10.1 test set, showing that increased ρ gave substantially worse generalization.

To test robustness of these models, we swept ϵ for both PGD attacks (Figure 6). The CEB_0 model not only has substantially higher accuracy than the adversarially-trained Wide ResNet from Madry et al. [14] (*Madry*), it also beats the Madry model on both the L_2 and the L_∞ attacks at almost all values of ϵ. We also show that this model is even more robust to two transfer attacks, where we used the CEB_5 model and the Madry model to generate PGD attacks, and then test them on the CEB_0 model. This result indicates that these models are not doing "gradient masking", a failure mode of some attempts at adversarial defense [2], since these are black-box attacks that do not rely on taking gradients through the target model.

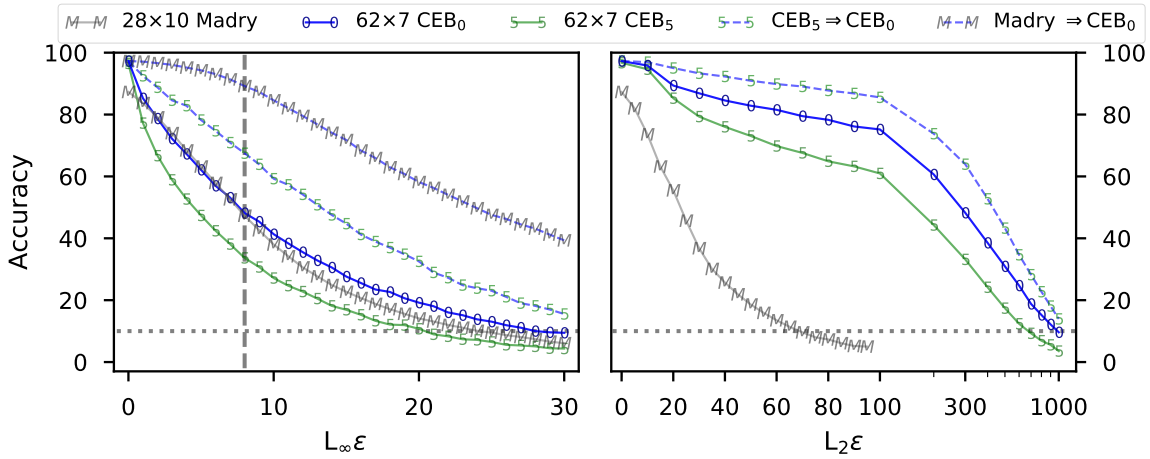

Figure 6. Left: Accuracy on untargeted L_∞ attacks at different values of ε for all 10,000 CIFAR10 test set examples. CEB_0 is the model with the highest accuracy (97.51%) trained at $\rho = 0$. CEB_5 is the model with the highest accuracy (97.06%) trained at $\rho = 5$. Madry is the best adversarially-trained model from Madry et al. [14] with 87.3% accuracy (values provided by Aleksander Madry). $\text{CEB}_5 \Rightarrow \text{CEB}_0$ is transfer attacks from the CEB_5 model to the CEB_0 model. Madry $\Rightarrow \text{CEB}_0$ is transfer attacks from the Madry model to the CEB_0 model. Madry was trained with 7 steps of PGD at $\varepsilon = 8$ (grey dashed line). Chance is 10% (grey dotted line). **Right:** Accuracy on untargeted L_2 attacks at different values of ε. All values are collected at 7 steps of PGD. CEB_0 outperforms Madry everywhere except the region of $L_\infty\varepsilon \in [2, 7]$. Madry appears to have overfit to L_∞, given its poor performance on L_2 attacks relative to either CEB model. *None of the CEB models are adversarially trained.*

4. Conclusions

We have presented the Conditional Entropy Bottleneck (CEB), motivated by the Minimum Necessary Information (MNI) criterion and the hypothesis that failures of *robust generalization* are due in part to learning models that retain *too much* information about the training data. We have shown empirically that simply by switching to CEB, models may substantially improve their robust generalization, including **(RG1)** higher accuracy, **(RG2)** better adversarial robustness, and **(RG3)** stronger OoD detection. We believe that the MNI criterion and CEB offer a promising path forward for many tasks in machine learning by permitting fast amortized inference in an easy-to-implement framework that improves robust generalization.

Funding: This research received no external funding.

Acknowledgments: We would like to thank Alex Alemi and Kevin Murphy for valuable discussions in the preparation of this work.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Model Details

Here we collect a number of results that are not critical to the core of the paper, but may be of interest to particular audiences.

Appendix A.1. Fashion MNIST

All of the models in our Fashion MNIST experiments have the same core architecture: A 7×2 Wide Resnet [43] for the encoder, with a final layer of $D = 4$ dimensions for the latent representation, followed by a two layer MLP classifier using ELU [46] activations with a final categorical distribution over the 10 classes.

The stochastic models parameterize the mean and variance of a $D = 4$ fully covariate multivariate Normal distribution with the output of the encoder. Samples from that distribution are passed into

the classifier MLP. Apart from that difference, the stochastic models don't differ from Determ during evaluation. None of the five models uses any form of regularization (e.g., L_1, L_2, DropOut [31], BatchNorm [47]).

The VIB models have an additional learned marginal, $m(z)$, which is a mixture of 240 $D = 4$ fully covariate multivariate Normal distributions. The CEB model instead has the backward encoder, $b(z|y)$ which is a $D = 4$ fully covariate multivariate Normal distribution parameterized by a 1 layer MLP mapping the label, $Y = y$, to the mean and variance. In order to simplify comparisons, for CEB we additionally train a marginal $m(z)$ identical in form to that used by the VIB models. However, for CEB, $m(z)$ is trained using a separate optimizer so that it doesn't impact training of the CEB objective in any way. Having $m(z)$ for both CEB and VIB allows us to compare the rate, R, of each model except Determ.

Appendix A.2. CIFAR-10

For the 62×7 CEB CIFAR-10 models, we used the AutoAugment policies for CIFAR-10. We trained the models for 800 epochs, lowering the learning rate by a factor of 10 at 400 and 600 epochs. We trained all of the models using Adam [48] at a base learning rate of 10^{-3}.

Appendix A.3. Distributional Families

Any distributional family may be used for the encoder. Reparameterizable distributions [29,49] are convenient, but it is also possible to use the score function trick [50] to get a high-variance estimate of the gradient for distributions that have no explicit or implicit reparameterization. In general, a good choice for $b(z|y)$ is the same distributional family as $e(z|x)$, or a mixture thereof. These are modeling choices that need to be made by the practitioner, as they depend on the dataset. In this work, we chose normal distributions because they are easy to work with and will be the common choice for many problems, particularly when parameterized with neural networks, but that choice is incidental rather than fundamental.

Appendix B. Mutual Information Optimization

As an objective function, CEB is independent of the methods used to optimize it. Here we focus on variational objectives because they are simple, tractable, and well-understood, but any approach to optimize mutual information terms can work, so long as they respect the side of the bounds required by the objective. For example, both Oord et al. [35], Hjelm et al. [51] could be used to maximize $I(Y; Z)$.

Appendix B.1. Finiteness of the Mutual Information

The conditions for infinite mutual information given in Amjad and Geiger [52] do not apply to either CEB or VIB, as they both use stochastic encoders $e(z|x)$. In our experiments using continuous representations, we did not encounter mutual information terms that diverged to infinity, although it is possible to make modeling and data choices that make it more likely that there will be numerical instabilities. This is not a flaw specific to CEB or VIB, however, and we found numerical instability to be almost non-existent across a wide variety of modeling and architectural choices for both variational objectives.

Appendix C. Additional CEB Objectives

Here we describe a few more variants of the CEB objective.

Appendix C.1. Hierarchical CEB

Thus far, we have focused on learning a single latent representation (possibly composed of multiple latent variables at the same level). Here, we consider one way to learn a hierarchical model with CEB.

Consider the graphical model $Z_2 \leftarrow Z_1 \leftarrow X \leftrightarrow Y$. This is the simplest hierarchical supervised representation learning model. The general form of its information diagram is given in Figure A1.

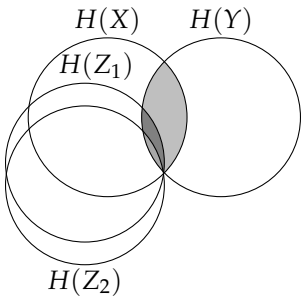

Figure A1. Information diagram for the basic hierarchical CEB model, $Z_2 \leftarrow Z_1 \leftarrow X \leftrightarrow Y$.

The key observation for generalizing CEB to hierarchical models is that the target mutual information doesn't change. By this, we mean that all of the Z_i in the hierarchy should cover $I(X;Y)$ at convergence, which means maximizing $I(Y;Z_i)$. It is reasonable to ask why we would want to train such a model, given that the final set of representations are presumably all effectively identical in terms of information content. Doing so allows us to train deep models in a principled manner such that all layers of the network are consistent with each other and with the data. We need to be more careful when considering the residual information terms, though – it is not the case that we want to minimize $I(X;Z_i|Y)$, which is not consistent with the graphical model. Instead, we want to minimize $I(Z_{i-1};Z_i|Y)$, defining $Z_0 = X$.

This gives the following simple *Hierarchical CEB* objective:

$$CEB_{\text{hier}} \equiv \min \sum_i I(Z_{i-1};Z_i|Y) - I(Y;Z_i) \tag{A1}$$

$$\Leftrightarrow \min \sum_i -H(Z_i|Z_{i-1}) + H(Z_i|Y) + H(Y|Z_i) \tag{A2}$$

Because all of the Z_i are targetting Y, this objective is as stable as regular CEB.

Appendix C.2. Sequence Learning

Many of the richest problems in machine learning vary over time. In Bialek et al. [53], the authors define the *Predictive Information*:

$$I(X_{past}, X_{future}) = \left\langle \log \frac{p(x_{past}, x_{future})}{p(x_{past})p(x_{future})} \right\rangle$$

This is of course just the mutual information between the past and the future. However, under an assumption of temporal invariance (any time of fixed length is expected to have the same entropy), they are able to characterize the predictive information, and show that it is a subextensive quantity: $\lim_{T\to\infty} I(T)/T \to 0$, where $I(T)$ is the predictive information over a time window of length $2T$ (T steps of the past predicting T steps into the future). This concise statement tells us that past observations contain vanishingly small information about the future as the time window increases.

The application of CEB to extracting the predictive information is straightforward. Given the Markov chain $X_{<t} \to X_{\geq t}$, we learn a representation Z_t that optimally covers $I(X_{<t}, X_{\geq t})$ in *Predictive CEB*:

$$CEB_{\text{pred}} \equiv \min I(X_{<t};Z_t|X_{\geq t}) - I(X_{\geq t},Z_t) \tag{A3}$$

$$\Rightarrow \min -H(Z_t|X_{<t}) + H(Z_t|X_{\geq t}) + H(X_{\geq t}|Z_t) \tag{A4}$$

Given a dataset of sequences, CEB_{pred} may be extended to a bidirectional model. In this case, two representations are learned, $Z_{<t}$ and $Z_{\geq t}$. Both representations are for timestep t, the first representing the observations before t, and the second representing the observations from t onwards. As in the normal bidirectional model, using the same encoder and backwards encoder for both parts of the bidirectional CEB objective ties the two representations together.

Appendix C.2.1. Modeling and Architectural Choices

As with all of the variants of CEB, whatever entropy remains in the data after capturing the entropy of the mutual information in the representation must be modeled by the decoder. In this case, a natural modeling choice would be a probalistic RNN with powerful decoders per time-step to be predicted. However, it is worth noting that such a decoder would need to sample at each future step to decode the subsequent step. An alternative, if the prediction horizon is short or the predicted data are small, is to decode the entire sequence from Z_t in a single, feed-forward network (possibly as a single autoregression over all outputs in some natural sequence). Given the subextensivity of the predictive information, that may be a reasonable choice in stochastic environments, as the useful prediction window may be small.

Likely a better alternative, however, is to use the CatGen decoder, as no generation of the long future sequences is required in that case.

Appendix C.2.2. Multi-Scale Sequence Learning

As in WaveNet [54], it is natural to consider sequence learning at multiple different temporal scales. Combining an architecture like time-dilated WaveNet with CEB is as simple as combining CEB_{pred} with CEB_{hier} (Appendix C.1). In this case, each of the Z_i would represent a wider time dilation conditioned on the aggregate Z_{i-1}.

Appendix C.3. Unsupervised CEB

For unsupervised learning, it seems challenging to put the decision about what information should be kept into objective function hyperparameters, as in the β VAE and penalty VAE [32] objectives. That work showed that it is possible to constrain the amount of information in the learned representation, but it is unclear how those objective functions keep only the "correct" bits of information for the downstream tasks you might care about. This is in contrast to supervised learning while targeting the MNI point, where the task clearly defines the both the correct amount of information and which bits are likely to be important.

Our perspective on the importance of defining a task in order to constrain the information in the representation suggests that we can turn the problem into a data modeling problem in which the practitioner who selects the dataset also "models" the likely form of the useful bits in the dataset for the downstream task of interest.

In particular, given a dataset X, we propose selecting a function $f(X) \to X'$ that transforms X into a new random variable X'. This defines a paired dataset, $P(X, X')$, on which we can use CEB as normal. Note that choosing the identity function for f results in maximal mutual information between X and X' ($H(X)$ nats), which will result in a representation that is far from the MNI for normal downstream tasks.

It may seem that we have not proposed anything useful, as the selection of $f(.)$ is unconstrained, and seems much more daunting than selecting β in a β VAE or σ in a penalty VAE. However, there is a very powerful class of functions that makes this problem much simpler, and that also make it clear using CEB will *only* select bits from X that are useful. That class of functions is the noise functions.

Appendix C.3.1. Denoising CEB Autoencoder

Given a dataset X without labels or other targets, and some set of tasks in mind to be solved by a learned representation, we may select a random noise variable U, and function $X' = f(X, U)$ that we

believe will destroy the irrelevant information in X. We may then add representation variables $Z_X, Z_{X'}$ to the model, giving the joint distribution $p(x, x', u, z_X, z_{X'}) \equiv p(x)p(u)p(x'|f(x,u))e(z_X|x)b(z_{X'}|x')$. This joint distribution is represented in Figure A2.

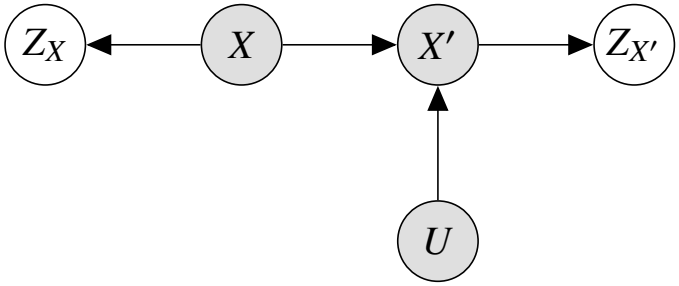

Figure A2. Graphical model for the Denoising CEB Autoencoder.

Denoising Autoencoders were originally proposed in Vincent et al. [55]. In that work, the authors argue informally that reconstruction of corrupted inputs is a desirable property of learned representations. In this paper's notation, we could describe their proposed objective as $\min H(X|Z_{X'})$, or equivalently $\min \langle \log d(x|z_{X'} = f(x,\eta)) \rangle_{x,\eta \sim p(x)p(\theta)}$.

We also note that, practically speaking, we would like to learn a representation that is consistent with uncorrupted inputs as well. Consequently, we are going to use a bidirectional model.

$$\begin{aligned} CEB_{\text{denoise}} &\equiv \min I(X; Z_X|X') - I(X'; Z_X) && \text{(A5)} \\ &\quad + I(X'; Z_{X'}|X) - I(X; Z_{X'}) && \text{(A6)} \\ &\Rightarrow \min -H(Z_X|X) + H(Z_X|X') + H(X'|Z_X) && \text{(A7)} \\ &\quad - H(Z_{X'}|X') + H(Z_{X'}|X) + H(X|Z_{X'}) && \text{(A8)} \end{aligned}$$

This requires two encoders and two decoders, which may seem expensive, but it permits a consistent learned representation that can be used cleanly for downstream tasks. Using a single encoder/decoder pair would result in either an encoder that does not work well with uncorrupted inputs, or a decoder that only generates noisy outputs.

If you are only interested in the learned representation and not in generating good reconstructions, the objective simplifies to the first three terms. In that case, the objective is properly called a *Noising CEB Autoencoder*, as the model predicts the noisy X' from X:

$$\begin{aligned} CEB_{\text{noise}} &\equiv \min I(X; Z_X|X') - I(X'; Z_X) && \text{(A9)} \\ &\Rightarrow \min -H(Z_X|X) + H(Z_X|X') + H(X'|Z_X) && \text{(A10)} \end{aligned}$$

In these models, the noise function, $X' = f(X, U)$ must encode the practitioner's assumptions about the structure of information in the data. This obviously will vary per type of data, and even per desired downstream task.

However, we don't need to work too hard to find the perfect noise function initially. A reasonable choice for f is:

$$\begin{aligned} f(x, \eta) &= \text{clip}(x + \eta, \mathcal{D}) && \text{(A11)} \\ \eta &\sim \lambda U(-1, 1) * \mathcal{D} && \text{(A12)} \\ \mathcal{D} &= \text{domain}(X) && \text{(A13)} \end{aligned}$$

In other words, add uniform noise scaled to the domain of X and by a hyperparameter λ, and clip the result to the domain of X. When $\lambda = 1$, X' is indistinguishable from uniform noise. As $\lambda \to 0$, this maintains more and more of the original information from X in X'. For some value of $\lambda > 0$,

most of the irrelevant information is destroyed and most of the relevant information is maintained, if we assume that higher frequency content in the domain of X is less likely to contain the desired information. That information is what will be retained in the learned representation.

Theoretical Optimality of Noise Functions

Above we claimed that this learning procedure will only select bits that are useful for the downstream task, given that we select the proper noise function. Here we prove that claim constructively. Imagine an oracle that knows which bits of information should be destroyed, and which retained in order to solve the future task of interest. Further imagine, for simplicity, that the task of interest is classification. What noise function must that oracle implement in order to ensure that $CEB_{denoise}$ can only learn exactly the bits needed for classification? The answer is simple: for every $X = x_i$, select $X' = x_i'$ uniformly at random from among all of the $X = x_j$ that should have the same class label as $X = x_i$. Now, the only way for CEB to maximize $I(X; Z_{X'})$ and minimize $I(X'; Z_{X'})$ is by learning a representation that is isomorphic to classification, and that encodes exactly $I(X; Y)$ nats of information, even though it was only trained "unsupervisedly" on X, X' pairs. Thus, if we can choose the correct noise function that destroys only the bits we don't care about, $CEB_{denoise}$ will learn the desired representation and nothing else (caveated by model, architecture, and optimizer selection, as usual).

References

1. Carlini, N.; Wagner, D. Adversarial examples are not easily detected: Bypassing ten detection methods. In Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, Dallas, TX, USA, 10–17 August 2017; pp. 3–14.
2. Athalye, A.; Carlini, N.; Wagner, D. Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples. In Proceedings of the 35th International Conference on Machine Learning (ICML), Stockholm, Sweden, 10–15 July 2018.
3. Nalisnick, E.; Matsukawa, A.; Teh, Y.W.; Gorur, D.; Lakshminarayanan, B. Do deep generative models know what they don't know? In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
4. Guo, C.; Pleiss, G.; Sun, Y.; Weinberger, K.Q. On Calibration of Modern Neural Networks. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017.
5. Lakshminarayanan, B.; Pritzel, A.; Blundell, C. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. In *Advances in Neural Information Processing Systems 30*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S.; Garnett, R., Eds.; MIT Press: Cambridge, MA, USA, 2017; pp. 6402–6413; Available online: https://papers.nips.cc/paper/7219-simple-and-scalable-predictive-uncertainty-estimation-using-deep-ensembles (accessed on 7 September 2020).
6. Hendrycks, D.; Gimpel, K. A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. In Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
7. Liang, S.; Li, Y.; Srikant, R. Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
8. Lee, K.; Lee, H.; Lee, K.; Shin, J. Training Confidence-calibrated Classifiers for Detecting Out-of-Distribution Samples. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
9. Devries, T.; Taylor, G.W. Learning Confidence for Out-of-Distribution Detection in Neural Networks. *arXiv* **2018**, arXiv:1802.04865.
10. Zhang, C.; Bengio, S.; Hardt, M.; Recht, B.; Vinyals, O. Understanding deep learning requires rethinking generalization. In Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017.

11. Tishby, N.; Pereira, F.C.; Bialek, W. The information bottleneck method. In Proceedings of the 37th annual Allerton Conference on Communication, Control, and Computing, Allerton, IL, USA, 22–24 September 1999; pp. 368–377.
12. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and Harnessing Adversarial Examples. *arXiv* **2014**, arXiv:1412.6572. Available online: https://arxiv.org/abs/1412.6572 (accessed on 7 September 2020).
13. Kurakin, A.; Goodfellow, I.; Bengio, S. Adversarial machine learning at scale. In Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
14. Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; Vladu, A. Towards Deep Learning Models Resistant to Adversarial Attacks. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
15. Cohen, J.M.; Rosenfeld, E.; Kolter, J.Z. Certified adversarial robustness via randomized smoothing. *arXiv* **2019**, arXiv:1902.02918.
16. Wong, E.; Schmidt, F.; Metzen, J.H.; Kolter, J.Z. Scaling provable adversarial defenses. In *Advances in Neural Information Processing Systems*; NIPS: La Jolla, CA, USA, 2018; Available online: https://papers.nips.cc/paper/8060-scaling-provable-adversarial-defenses (accessed on 7 September 2020).
17. Alemi, A.A.; Fischer, I.; Dillon, J.V.; Murphy, K. Deep Variational Information Bottleneck. In Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
18. Alemi, A.A.; Fischer, I.; Dillon, J.V. Uncertainty in the Variational Information Bottleneck. *arXiv* **2018**, arXiv:1807.00906. Available online: https://arxiv.org/abs/1807.00906 (accessed on 7 September 2020).
19. Achille, A.; Soatto, S. Information dropout: Learning optimal representations through noisy computation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 2897–2905. [CrossRef] [PubMed]
20. Shannon, C.E. A Mathematical Theory of Communication. *Bell Syst. Tech. J.* **1948**, *27*, 379–423. [CrossRef]
21. Cover, T.M.; Thomas, J.A. *Elements of Information Theory*, 2nd ed.; John Wiley & Sons: Hoboken, NJ, USA, 2006.
22. Fischer, I. Bounding the Multivariate Mutual Information. Information Theory and Machine Learning Workshop. 2019. Available online: https://drive.google.com/file/d/17lJiJ4v_6h0p-ist_jCrr-o1ODi7yELx/view (accessed on 7 September 2020).
23. Anantharam, V.; Gohari, A.; Kamath, S.; Nair, C. On hypercontractivity and a data processing inequality. In Proceedings of the 2014 IEEE International Symposium on Information Theory, Honolulu, HI, USA, 29 June–4 July 2014; pp. 3022–3026.
24. Polyanskiy, Y.; Wu, Y. Strong data-processing inequalities for channels and Bayesian networks. In *Convexity and Concentration*; Springer: New York, NY, USA, 2017; pp. 211–249.
25. Wu, T.; Fischer, I.; Chuang, I.L.; Tegmark, M. Learnability for the Information Bottleneck. *Entropy* **2019**, *21*, 924.10.3390/e21100924. [CrossRef]
26. Shamir, O.; Sabato, S.; Tishby, N. Learning and generalization with the information bottleneck. *Theor. Comput. Sci.* **2010**, *411*, 2696–2711. [CrossRef]
27. Bassily, R.; Moran, S.; Nachum, I.; Shafer, J.; Yehudayoff, A. Learners that Use Little Information. In Proceedings of the Machine Learning Research, New York, NY, USA, 23–24 February 2018; Janoos, F., Mohri, M., Sridharan, K., Eds.; 2018; Volume 83, pp. 25–55; Available online: http://proceedings.mlr.press/v83/bassily18a.html (accessed on 7 September 2020)
28. Ilyas, A.; Santurkar, S.; Tsipras, D.; Engstrom, L.; Tran, B.; Madry, A. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems*; NIPS: La Jolla, CA, USA, 2019; pp. 125–136; Available online: https://papers.nips.cc/paper/8307-adversarial-examples-are-not-bugs-they-are-features (accessed on 7 September 2020).
29. Kingma, D.P.; Welling, M. Auto-encoding variational Bayes. In Proceedings of the International Conference on Learning Representations, Banff, AB, Canada, 14–16 April 2014.
30. Yeung, R.W. A new outlook on Shannon's information measures. *IEEE Trans. Inf. Theory* **1991**, *37*, 466–474. [CrossRef]
31. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
32. Alemi, A.A.; Poole, B.; Fischer, I.; Dillon, J.V.; Saurous, R.A.; Murphy, K. Fixing a Broken ELBO. In *ICML2018*. 2018. Available online: https://icml.cc/Conferences/2018/ScheduleMultitrack?event=2442 (accessed on 7 September 2020).

33. Vedantam, R.; Fischer, I.; Huang, J.; Murphy, K. Generative Models of Visually Grounded Imagination. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
34. Higgins, I.; Sonnerat, N.; Matthey, L.; Pal, A.; Burgess, C.P.; Bošnjak, M.; Shanahan, M.; Botvinick, M.; Hassabis, D.; Lerchner, A. SCAN: Learning Hierarchical Compositional Visual Concepts. *arXiv* **2018**, arXiv:1707.03389.
35. Oord, A.V.D.; Li, Y.; Vinyals, O. Representation learning with contrastive predictive coding. *arXiv* **2018**, arXiv:1807.03748.
36. Poole, B.; Ozair, S.; van den Oord, A.; Alemi, A.A.; Tucker, G. On Variational Bounds of Mutual Information. In Proceedings of the ICML2019, Long Beach, CA, USA, 9–15 June 2019.
37. Goyal, P.; Dollár, P.; Girshick, R.; Noordhuis, P.; Wesolowski, L.; Kyrola, A.; Tulloch, A.; Jia, Y.; He, K. Accurate, large minibatch SGD: Training imagenet in 1 hour. *arXiv* **2017**, arXiv:1706.02677 .
38. Xiao, H.; Rasul, K.; Vollgraf, R. Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms. *arXiv* **2017**, arXiv:1708.07747.
39. Krizhevsky, A.; Hinton, G. *Learning Multiple Layers of Features From Tiny Images*; Technical Report; University of Toronto: Toronto, ON, USA, 2009; Available online: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf (accessed on 7 September 2020).
40. Carlini, N.; Wagner, D. Towards evaluating the robustness of neural networks. In Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2017; pp. 39–57.
41. Papernot, N.; Faghri, F.; Carlini, N.; Goodfellow, I.; Feinman, R.; Kurakin, A.; Xie, C.; Sharma, Y.; Brown, T.; Roy, A.; et al. Technical Report on the CleverHans v2.1.0 Adversarial Examples Library. *arXiv* **2018**, arXiv:1610.00768.
42. Lee, K.; Lee, K.; Lee, H.; Shin, J. A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks. In *Advances in Neural Information Processing Systems 31*; Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R., Eds.; NIPS: La Jolla, CA, USA, 2018; pp. 7167–7177; Available online: https://papers.nips.cc/paper/7947-a-simple-unified-framework-for-detecting-out-of-distribution-samples-and-adversarial-attacks (accessed on 7 September 2020).
43. Zagoruyko, S.; Komodakis, N. Wide Residual Networks. In *Proceedings of the British Machine Vision Conference (BMVC), York, UK, 9–12 September 2019*; Richard, C., Wilson, E.R.H., Smith, W.A.P., Eds.; BMVA Press: London, UK, 2016; pp. 87.1–87.12; Available online: http://www.bmva.org/bmvc/2016/papers/paper087/ (accessed on 7 September 2020).
44. Cubuk, E.D.; Zoph, B.; Mane, D.; Vasudevan, V.; Le, Q.V. AutoAugment: Learning Augmentation Strategies From Data. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019.
45. Recht, B.; Roelofs, R.; Schmidt, L.; Shankar, V. Do CIFAR-10 classifiers generalize to CIFAR-10? *arXiv* **2018**, arXiv:1806.00451.
46. Clevert, D.A.; Unterthiner, T.; Hochreiter, S. Fast and accurate deep network learning by exponential linear units (elus). In Proceedings of the International Conference on Learning Representations Workshop, San Juan, Puerto Rico, 2–4 May 2016.
47. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the Machine Learning Research, Lille, France, 7–9 July 2015*; Bach, F., Blei, D., Eds.; PMLR: Lille, France, 2015; Volume 37, pp. 448–456; Available online: http://proceedings.mlr.press/v37/ioffe15 (accessed on 7 September 2020).
48. Kingma, D.; Ba, J. Adam: A method for stochastic optimization. In Proceedings of the International Conference on Learning Representations, San Diego, CA, USA, 4–8 May 2015.
49. Figurnov, M.; Mohamed, S.; Mnih, A. Implicit Reparameterization Gradients. In *Advances in Neural Information Processing Systems 31*; Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R., Eds.; NIPS: La Jolla, CA, USA, 2018; pp. 441–452; Available online: https://papers.nips.cc/paper/7326-implicit-reparameterization-gradients (accessed on 7 September 2020).
50. Williams, R.J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.* **1992**, *8*, 229–256. [CrossRef]

51. Hjelm, R.D.; Fedorov, A.; Lavoie-Marchildon, S.; Grewal, K.; Bachman, P.; Trischler, A.; Bengio, Y. Learning deep representations by mutual information estimation and maximization. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
52. Amjad, R.A.; Geiger, B.C. Learning Representations for Neural Network-Based Classification Using the Information Bottleneck Principle. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 2225–2239. [CrossRef] [PubMed]
53. Bialek, W.; Nemenman, I.; Tishby, N. Predictability, complexity, and learning. *Neural Comput.* **2001**, *13*. [CrossRef] [PubMed]
54. Van Den Oord, A.; Dieleman, S.; Zen, H.; Simonyan, K.; Vinyals, O.; Graves, A.; Kalchbrenner, N.; Senior, A.W.; Kavukcuoglu, K. WaveNet: A Generative Model for Raw Audio. *arXiv* **2016**, arXiv:1609.03499. Available online: https://arxiv.org/abs/1609.03499 (accessed on 7 September 2020).
55. Vincent, P.; Larochelle, H.; Bengio, Y.; Manzagol, P.A. Extracting and composing robust features with denoising autoencoders. In Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, 5–9 July 2008; pp. 1096–1103.