University of Amsterdam

MSc Artificial Intelligence
Master Thesis

Validating the Information Bottleneck principle

by
Nikita Tokovenko
12185892

September 28, 2020

36 EC
February 2020 - September 2020

Assessor:
Dr Efstratios Gavves
QUVA LAB
University of Amsterdam

Supervisor:
MSc Marco Federici
AMLab
University of Amsterdam

Informatics Institute
AMLab

Nikita Tokovenko
Validating the Information Bottleneck principle

September 28, 2020
36 EC
February 2020 - September 2020
Examiner: Dr Efstratios Gavves
Supervisor: MSc Marco Federici

University of Amsterdam
AMLab
Informatics Institute
Science Park 904
1098 XH and Amsterdam

Abstract

Information-based models for deep learning has become a sought-after subject of ongoing research. Many recent methods for self-supervised and unsupervised representation learning train feature extractors having their motivation from the information theory. Such an approach proved its ability to reach state-of-the-art performance in many applications. We believe that investigating the Information Bottleneck principle will give useful insights to find a key to understanding how to build more efficient and accurate algorithms to solve the tasks of deep learning.

The target goal of this thesis is to define a procedure to make one able to reason about the generalization capability of different encoding procedures. To do this we proposed a framework to make a comparison of various supervised encoding techniques. We show that incorporation of the Information Bottleneck principles leads to a substantial improvement of generalization and compression ability of deep neural models. Our investigation leads to the development of the novel method for learning representations from the image data that shows promising results and is competitive to the previous state-of-the-art.

Acknowledgement

> *The real challenge of growth mentally, emotionally, spiritually comes when you get knocked down. It takes courage to act.*
>
> — **Author unknown**

I want to thank everyone who believed in me and helped to keep the fire of curiosity about life burning.

Contents

1 Introduction

1.1 Motivation and Problem Statement

Recent successes in the field of deep learning have been one of the biggest breakthroughs towards building Artificial General Intelligence so far. Deep neural networks showed to surpass most competitors for solving supervised learning tasks on real data challenges (Hinton et al., 2012, Bengio et al., 2012, Krizhevsky et al., 2012). Despite having strong empirical success, a theoretical understanding of its reasons remains unsatisfactory. By incorporating concepts from the information theory, the research community managed to get to a certain point of understanding of the underlying theory behind the black-box nature of deep neural networks (Saxe et al., 2018, Shwartz-Ziv and Tishby, 2017). Specifically, using the information-theoretic approach to the investigation of signal processing systems, the goal of deep learning could be formulated as a trade-off between compression and prediction.

The one could reasonably ask: why do we care about data compression? The overall success of the application of any machine learning algorithm in the real world somehow or other could be attributed to its generalization performance - by an ability to work well on previously unseen data. The latter is very important to provide sustainable solutions to predictive tasks. So the main motivation for compression can be seen as that passing raw input features to the deep neural network cannot guarantee good generalization. The initial representation of the data might be too complex and have a lot of attributes that can only prevent accurate decision making and prediction. Thus, by stacking hidden layers we should aim to compress the input by discarding irrelevant features to ensure the worst-case generalization error is tolerable (Tishby and Zaslavsky, 2015).

Why discarding irrelevant features is useful in practice? Imagine an example of the task for image classification of the shape for two types of colored objects: squares and circles. Note that the color information is completely irrelevant, as it does not have any causal effect on the end shape of the entity. Of course, we can fit the vanilla neural network on training examples from the finite sample. However, there are no guarantees for the end model to generalize well if we bump into an instance of previously unseen color. Thus, if we can get rid of the color information from the representation before actually making the decision, it might significantly improve the chances of correctly classifying the shape of the incoming object. On the other hand, having fewer attributes conveyed in representations narrows the cardinality of the space for classifier approximators. It means that it potentially becomes much easier to choose the optimal linear classifier.

In turn, the Information Bottleneck (IB) principle (Tishby et al., 1999), having its background and motivation from the Rate-Distortion theory, states that it is possible to find an optimal bottleneck point, where the amount of information needed to solve the task is maximal saturated while requiring least amount of bits to store it. The positioning of this point on the information plane[1] corresponds to the amount of task-relevant information being maximal while keeping the number of additional, useless features conveyed in a representation to be minimal.

In this work, we aim to validate the IB principle via investigation of its variational approximation, as direct application to deep learning problems is hard due to limitations outlined later in the text. We provide an extensive comparison to commonly-used regularization techniques of weight decay and dropout and show that the incorporation of the Information Bottleneck method's variations can outperform existing tools for generalization improvement in the context of representation learning.

Thus, the main research contributions of this thesis are:

1. Definition of the custom build metric to access the generalization capability of deep neural networks;
2. Development of the unified framework to reason about the quality of different encoding procedures concerning compression and generalization performance;
3. Proposition of a novel method for learning representations based on discarding only *task-irrelevant* information.

1.2 Related work

Previously, the idea for generalization improvement for neural networks was widely addressed through the study of regularization. One of the standard regularization techniques that proved to cause better generalization in a feed-forward networks is a *weight decay* or, alternatively, L_2-regularization (Krogh and Hertz, 1992). The core

[1] By this we refer to planes constructed by having the estimate for values of mutual information between the representation and the input positioned on the x-axis while having the estimates for values of mutual information between the representation and prediction task labels on the y-axis

idea is to constrain a network complexity by penalizing large weights for growing too large during training unless it is really necessary. Thus, the generalization improvement is achieved by suppressing any irrelevant components of the weight vector by choosing the smallest that solves the learning problem. In practice, it is equivalent to using the maximum a posteriori estimation (Chapter 1 from (Bishop, 2006) on the parameters of the model, that in Bayesian neural networks (Neal, 1995) case is equivalent to having a Gaussian prior on the weights.

Second commonly used technique is *dropout* (Srivastava et al., 2014). Using this technique, the compression is achieved by randomly dropping out units in a neural network at every training time-step. Thus, for each iteration the backpropagation is only done for units that "survived" the dropout and managed to contribute to the calculation of the empirical loss, preventing the network from overfitting the training data.

Recently these standard techniques gained popularity among machine and deep learning practitioners, resulting in further development for extensions of these regularization methods (Molchanov et al., 2017, Achille and Soatto, 2018, Zolna et al., 2018, Singh et al., 2016).

On the other hand, the Information Bottleneck (IB) principle in the original formulation was first proposed in (Tishby et al., 1999) with further extension concerning the deep learning problem formulation in (Tishby and Zaslavsky, 2015). Unfortunately, in the initial formulation, the optimization procedure relied on the Blahut-Arimoto algorithm (Yeung, 2008), which is unfeasible to apply to modern deep neural networks, due to its iterative nature. Thus, to solve real-world machine learning tasks the IB method had to be adopted. The solution came with the development of the variational approximation to the IB principle. The resulting method got the name of the Deep Variational Information Bottleneck, or, shortly, VIB (Alemi et al., 2016). Being closely related to Variational Autoencoders (VAEs) (Kingma and Welling, 2014) in the unsupervised learning setting, the VIB method efficiently incorporates variational inference through the reparametrization trick to training stochastic neural networks. In (Alemi et al., 2016) authors provided experimental results showing their method of VIB to be state-of-the-art and outperforming others, apart from those studied in our work, techniques for regularization - confidence penalty (Pereyra et al., 2017) and label smoothing (Müller et al., 2019). In our research, we go further by extending the VIB formulation to the case of compression being achieved by discarding only task-irrelevant features via the superfluous information minimization. Such opening results in a novel method we give the name of the *Conditional Variational Information Bottleneck*. The term "conditional" appears due to the fact of using the variational approximation of the conditional prior on the compressed space to force the regularizing effect on the representations.[2]

One of the main drawbacks of information-theoretically inspired methods for deep learning is that often the usage of such methods is constrained by the nature of concepts involved. For instance, the center for the IB concept of mutual information is notoriously hard to be operated on directly while working on real data challenges. Because of this, the previous validation of the IB principle was hard to be achieved. Some works managed to try to prove the underlying theory on the synthetic tasks (Shwartz-Ziv and Tishby, 2017) or using trivial methods to approximate the mutual information, such as compressed space quantization to calculate the mutual information via binning (Saxe et al., 2018, Cheng et al., 2018).

Luckily, despite having certain general limitations (McAllester and Stratos, 2020), recent progress in the field of mutual information estimation led to the successful development of ways to measure the complexity of representations, as well as, to measure the ability of deep learning algorithms to compress the data (Poole et al., 2019, Belghazi et al., 2018 etc.). Worth mentioning that such approaches found its wide application in the field of deep representation learning through direct maximization of the estimate (Hjelm et al., 2018, Oord et al., 2018, Löwe et al., 2019 etc.). However, recent work of (Tschannen et al., 2020) explains why successes of representation learning methods that make the use of mutual information estimation should not be attributed to the properties of the estimation alone. Namely, their experimental results show that the quality of representations might be impacted more by carefully choosing the architecture of the encoding network, than the particular choice of estimator. However, in our work, we only study the topic of mutual information estimation in the context of evaluating the ability to compress for different encoding procedures and do this disregard for the training methods for obtaining high-quality representations from image data.

Usually, the quality or representations is assessed via the linear evaluation protocol (e.g. Kolesnikov et al., 2019). In our work, we adopt this commonly used technique to measure the generalization capability of representations concerning the drop of performance for the end classifier in the case of additional training data cut-off.

Given everything above, our validation method is formalized as a unified information-theoretic framework being able to successfully incorporate both measures for generalization and compression.

[2] In the original VIB method this is done using the variational marginal distribution, same as for the case of VAEs.

1.3 Hypothesis and research questions

Our hypothesis: the assessment of the generalization capabilities of deep neural models, trained with different objectives, can be done by investigating the trade-off between compression and retaining predictive accuracy and, hence, by doing so, we can improve upon existing techniques for learning high-level representations.

In order to validate this hypothesis we aim to answer following research questions:

1. How can we quantitatively evaluate the quality of representations with respect to the amount of mutual information preserved?

2. How does the choice of the model architecture affect this metric?

3. How true is that usage of information bottleneck method helps to compress the data without the loss of its expressiveness?

4. How does using information bottleneck objective relate to applying regularization to the model during training?

5. Can we design a more effective method to discard only irrelevant information with respect to VIB?

1.4 Outline

Chapter 2 provides the general formulation of the information-theoretic approach to training deep neural networks. We start our investigation with the relevance extraction problem formulation. Then we position the Information Bottleneck principle in the context of supervised learning using deep neural networks. In Section 2.4, we thoroughly discuss the underlying structure of the Deep Variational Information Bottleneck method, with further discussion on our novel modification in Section 2.5.

In Chapter 3, we do an explanation of the validation procedure chosen for the scope of this work. We provide general information on the problem of mutual information estimation, as well as supplementary discussion on implementation details. Later, in Section 3.2, we present a custom build metric for linear evaluation to assess the generalization capabilities of representation learning algorithms.

Chapter 4 starts with an overview of datasets and models used for doing our research. In Section 4.3, we provide an extensive explanation and discussion of the results for the empirical study of the problem of the Information Bottleneck principle validation.

Conclusions and future work discussion are present in Chapter 5.

Information-theoretic perspective of Deep learning 2

In this chapter, we investigate the application of Information Bottleneck (IB) to deep neural networks used to learn representations of the data. We formulate the general overview of the information-theoretic approach in the context of supervised representation learning. Also, we discuss how can we improve upon the state-of-the-art methods (Alemi et al., 2016).

2.1 Extracting relevance using Information theory

In the information theory formulation the goal of representation learning is to define an encoding procedure to get the relevant quantization of the input signal X[1]. To build a space of compressed representations (or quantized codebook) $\widetilde{\mathcal{X}}$, we aim to find a possibly stochastic mapping characterized by p.d.f. $p(X|\widetilde{X})$. As in this section we assume both $\mathcal{X}$ and $\widetilde{\mathcal{X}}$ to be finite, by marginalizing all possible values of X we can define the prior probability distribution for our codewords, so we can access the probability measure $p(\widetilde{X})$ directly.

Traditionally, the average number of elements from $\mathcal{X}$ that correspond to the same element in the codebook is $2^{H(X|\widetilde{X})}$, where $H(X|\widetilde{X})$ is the conditional entropy of X given $\widetilde{X}$. Recall that mutual information $I(X;\widetilde{X})$ shows the reduction in the uncertainty about the value of the input X after observing its representation $\widetilde{X}$. Namely, it can be calculated as the difference:

$$I(X;\widetilde{X}) = H(X) - H(X|\widetilde{X}). \tag{2.1}$$

The question that arises is how to determine the quality of quantization built.

[1] Here $X \in \mathcal{X}$ denotes random variable (message) coming from the space of all possible signals $\mathcal{X}$ with provided fixed probability measure $p(X)$

2.1.1 Approaching relevant quantization via Rate Distortion theory

As we intend to obtain the compressed version of the original data, the quality of quantization can be assessed by calculating the rate[2]. For each possible message from $\mathcal{X}$ this value is bounded from below by $I(X;\widetilde{X})$. However, decreasing the information rate causes discarding attributes of the original signal, potentially worsening the predictive capability of representations.

In rate distortion theory this issue is addressed by introducing the distortion function, $d : \mathcal{X} \times \widetilde{\mathcal{X}} \to \mathbb{R}^+$, that is presumed to be small for good solutions. Naturally, having more attributes preserved by the encoding procedure leads to smaller expected distortion $\mathbb{E}_{p(X,\widetilde{X})} d(X,\widetilde{X})$, since representations are becoming more informative about the input. According to the rate distortion theorem of Shannon and Kolmogorov (Cover and Thomas, 1991) the trade-off can be characterized by the rate distortion function $R(D)$, where argument D corresponds to the maximum tolerable level of distortion. Now the problem can be formalized as a constraint optimization problem, since we aim to minimize the achievable rate while not exceeding some level of expected distortion:

$$R(D) = \min_{\{p(\widetilde{X}|X):\mathbb{E}_{p(X,\widetilde{X})} d(X,\widetilde{X}) \leq D\}} I(X;\widetilde{X}) \tag{2.2}$$

This is a variational problem of finding proper conditional p.d.f. $p(\widetilde{X}|X)$. Solutions to it can be found as the one to minimize the functional with introduced Lagrange multiplier β for the expected distortion we want to constrain:

$$\mathcal{F}[p(\widetilde{X}|X),\beta] = I(X;\widetilde{X}) + \beta \mathbb{E}_{p(X,\widetilde{X})} d(X,\widetilde{X}) \tag{2.3}$$

In such formulation this problem has the following closed-form solution:

$$p(\widetilde{X}|X) = \frac{p(\widetilde{X})}{Z(X,\beta)} e^{-\beta d(X,\widetilde{X})}, \tag{2.4}$$

where $Z(X,\beta)$ is a normalization function. For the given level of distortion D, the corresponding value of the Lagrange multiplier is positive and satisfies $\frac{\partial R}{\partial D} = -\beta$.

[2] Rate, in this case, defines the average number of bits per any message $x \in \mathcal{X}$ that is sufficient to specify the corresponding codeword with no probability of confusion

However, there is no easy way to define the right form of the distortion function in the general case making this solution inapplicable to real-world problems.

2.1.2 Relevance through Information Bottleneck

In the original work on the Information Bottleneck (IB) method (Tishby et al., 1999) authors address this issue by proposing an alternative way to define the relevance of quantization - by measuring the amount of relevant information preserved about another variable. Namely, if we are solving the task of predicting Y from X, Y must not be independent of X, causing the mutual information $I(X;Y)$ to be positive. Since $I(X;Y)$ shows exactly the amount of information available to solve the task, the reasonable intention would be to build the relevant quantization $\widetilde{\mathcal{X}}$ in a way to share the as much amount of information about Y $(I(\widetilde{X},Y))$ as possible. According to the Data Processing Inequality, we don't create any new task-related information by processing X. This means that no representation $\widetilde{X}$ can convey more information about the task than the original data, meaning $I(\widetilde{X},Y) \leq I(X;Y)$ for any choice of $\widetilde{X}$. However, the same as before the goal is to maximize the level of compression. We again face a trade-off between preserving meaningful information and compressing the representations.

Now optimization objective 2.3 can be reformulated as:

$$\mathcal{L}[p(\widetilde{X}|X),\beta] = I(X;\widetilde{X}) - \beta I(\widetilde{X},Y) \tag{2.5}$$

We can control the trade-off by adjusting only parameter β. As $\beta \to \infty$, we push the encoder to build an arbitrarily detailed quantization, while setting $\beta = 0$ will result in assigning a single codeword as a representative for all possible messages.

According to authors of (Tishby et al., 1999), the optimal assignment for $p(\widetilde{X}|X)$ that minimizes 2.5 should satisfy the equation:

$$p(\widetilde{X}|X) = \frac{p(\widetilde{X})}{Z(X,\beta)} e^{-\beta \sum_{y\in\mathcal{Y}} p(y|X)\log\frac{p(y|X)}{p(y|\widetilde{X})}}, \tag{2.6}$$

where all random variables follow a Markov chain under the condition $Y \leftrightarrow X \leftrightarrow \widetilde{X}$ and the distribution $p(Y|\widetilde{X})$ is obtained by the Bayes rule. Notice that the exponent argument corresponds to $D_{KL}(p(Y|X)||p(Y|\widetilde{X}))$ scaled by the factor β. Thus, solution becomes similar to 2.4 with the Kullback-Leibler divergence as the distortion measure.

In (Tishby et al., 1999) authors provide detailed derivation for 2.4 and 2.6, as well as, an iterative method for finding these unknown distributions based on the Blahut-Arimoto (BA) algorithm (Yeung, 2008).

2.2 Supervised learning prospect

In real-life supervised machine learning tasks, probabilistic spaces are rarely accessible. Instead we are provided with a dataset $\mathcal{D}_n = \{(x_i, y_i)\}_{i=1}^{n}$ of pairs of instances of random variables X and Y drawn from the joint distribution $p(X,Y) = P(Y|X)P(X)$. In such cases, X and Y are often referred to as input features and task outputs, respectively. Supervised learning aims to learn the approximation for the true conditional density $p(Y|X)$. This task has proved to be successfully solvable by deep neural networks (Hinton et al., 2012). Namely, we seek for a reconstruction $\hat{Y}$ characterized by the conditional density $q(\hat{Y}|X)$ parametrized by the set of trainable parameters $\boldsymbol{\theta}$, or simply $q(\hat{Y}|X;\boldsymbol{\theta})$, to be as close to the original task outputs as possible.

In deep representation learning we want to learn intermediate representations Z of the input features (e.g. hidden layers activations), such that all random variables follow the Markov condition $Y \leftrightarrow X \leftrightarrow Z$. Given that, we can achieve the end-goal of learning $q(Y|X;\boldsymbol{\theta})$ by learning encoding and decoding distributions $p(Z|X)$ and $q(Y|Z)$ approximated by the neural network of our choice parametrized by $\boldsymbol{\theta}$.[3]

2.2.1 Relevance through cross-entropy minimization

In the tasks of classification, when the variable Y is discrete and represents belonging to one of the target classes, the common choice for the cost function falls on the cross-entropy, which in our case can be formulated as the representation cross-entropy cost function:

$$J_{CE}(p(X,Y);\boldsymbol{\theta}) = \mathbb{E}_{(z,x)\sim p(Z|X;\boldsymbol{\theta})p(X)}\big[-\mathbb{E}_{y\sim p(Y|Z=z)}\log(q(y|Z=z;\boldsymbol{\theta}))\big], \quad (2.7)$$

where $p(Y|Z) = \mathbb{E}_{x\sim p(X)}\left[\frac{p(Y|X=x)p(Z|X=x;\boldsymbol{\theta})}{\mathbb{E}_{x\sim p(X)}\left[p(Z|X=x;\boldsymbol{\theta})\right]}\right]$.

[3] In order to avoid confusion with $p(\cdot)$ we refer to the true distribution and with $q(\cdot)$ to the variational approximation. For the improved readability from now on we will assume all distributions can be parametrized by the set of trainable parameters $\boldsymbol{\theta}$ present in the model, making notation $q(\cdot;\boldsymbol{\theta})$, $p(\cdot;\boldsymbol{\theta})$ and $q(\cdot)$, $p(\cdot)$ being equal respectively. We still use the expanded notation to show the dependence on the parameters explicitly where it is needed.

Thus, by training neural networks concerning minimizing 2.7 we aim to learn the structure of the data that mimics its true structure characterized by $p(Y|X)$.

It is also known that minimizing cross-entropy implies maximizing the amount of relevant information $I(Z;Y)$ preserved by the intermediate variable (Proposition 1 of Rodríguez Gálvez, 2020). We will show this fact explicitly in the following section (see 2.9).

In turn, direct minimization of $I(X;Z)$ while either minimizing 2.7 (Theorem 1 of Vera et al., 2018) or maximizing $I(Z;Y)$ (Theorem 4 of Sharim et al., 2010) leads to the tightening of the generalization gap.[4]

Thus, we built a bridge between the information-theoretic and supervised learning points of view on building representations of the data.

2.3 IB in the context of training deep neural networks

View of the Information Bottleneck in the context of training deep neural networks was first pointed in (Tishby and Zaslavsky, 2015). In this work, the authors address questions of theoretical limits on the efficiency of such training, as well as complexity bounds on learning empirical estimate of the mutual information based on the finite sample distributions. The worth-noting conclusion drawn by authors is that we should constrain the complexity of representations to sustain a tolerable level of generalization drop when given unseen data.

However, no experimental results were provided, since the learning method proposed relied on the BA algorithm. The latter makes training neural networks interminable due to the high-dimension nature of the data, such as images. Furthermore, the goal of the BA algorithm is to find an optimal partitioning of the input space $\mathcal{X}$, given the chosen space of representatives $\mathcal{Z}$ (or $\widetilde{\mathcal{X}}$, as formulated in Section 2.1), but not to learn the mapping directly from data.

Alternatively to minimizing the objective 2.5 one can aim to maximize the inverse objective. The motivation under it is that we narrow our focus on the problem of

[4] By this we refer to the error incurred by computing the empirical loss function as an average of point-wise losses for the dataset rather than the expectation over the entire joint distribution $P(X,Y)$, i.e. the real cost.

maintaining the performance level, while ensuring that a certain level of compression is achieved. This implies maximizing the following Lagrangian:

$$\mathcal{L}[q(Z|X), \beta] = I(Z;Y) - \beta I(X;Z) \tag{2.8}$$

Note that such reformulation implies corresponding reparametrization of the trade-off parameter β.

But still, the major issue that arises when applying IB directly is that computing true mutual information is notoriously hard, as this requires knowledge of the marginal distributions of variables involved. Of course, in cases when the cardinality of sample space is finite, meaning that we can access the discrete distribution, IB can be applied directly, but often it is not the case.

2.4 Bounding the IB objective with variational inference

Instead of fighting hard trying to access the values of MI directly, we can derive a tractable lower bound on 2.8 using variational inference. Such variational approximation to the Information Bottleneck was first proposed in (Alemi et al., 2016) and the corresponding training method was given the name of Variational Information Bottleneck (VIB).

The core idea is to derive lower and upper bounds on $I(Z;Y)$ and $I(X;Z)$, respectively.

$$\begin{aligned}
I(Z;Y) = \mathbb{E}_{(y,z)\sim p(Y|Z)p(Z)}\big[\log \frac{p(y|Z=z)}{p(y)}\big] \\
= H(Y) + \mathbb{E}_{(y,z)\sim p(Y|Z)p(Z)}\big[\log p(y|Z=z)\big] \\
= H(Y) + \mathbb{E}_{z\sim p(Z)}\big[D_{KL}(p(Y|Z=z)||q(Y|Z=z;\boldsymbol{\theta})\big] \\
+ \mathbb{E}_{(y,z)\sim p(Y|Z)p(Z)}\big[\log q(y|Z=z;\boldsymbol{\theta})\big] \\
\geq H(Y) + \mathbb{E}_{(y,z)\sim p(Y|Z)p(Z)}\big[\log q(y|Z=z;\boldsymbol{\theta})\big] \\
= H(Y) - J_{CE}(p(X,Y);\boldsymbol{\theta}).
\end{aligned} \tag{2.9}$$

Since the entropy $H(Y)$ is constant, we can achieve maximization of $I(Z;Y)$ directly by minimizing $J_{CE}(p(X,Y);\theta)$. This approach is good since it only requires samples both from our stochastic encoder characterized by $p(Z|X)$ and samples from our joint data distribution $p(X, Y)$, as well as, a tractable variational approximation $q(Y|Z)$, which in such case we have.

In turn, the variational upper bound on $I(X; Z)$ can be derived as follows:

$$\begin{aligned} I(X;Z) &= \mathbb{E}_{(x,z)\sim p(Z|X)p(X)}\big[\log \frac{p(z|X=x;\boldsymbol{\theta})}{p(z)}\big] \\ &= \mathbb{E}_{(x,z)\sim p(Z|X)p(X)}\big[\log p(z|X=x;\boldsymbol{\theta})\big] + \mathbb{E}_{z\sim p(Z)}\big[\log p(z)\big]. \end{aligned} \tag{2.10}$$

As computing the marginal $p(Z)$ might be hard, its variational approximation $q(Z)$ can be introduced. Thus, following the same idea based on the non-negativity of the Kullback-Leibler divergence as in 2.9:

$$\begin{aligned} I(X;Z) &= D_{KL}(p(Z|X)||p(Z)) \\ &= \mathbb{E}_{(x,z)\sim p(Z|X)p(X)}\big[\log p(z|X=x;\boldsymbol{\theta})\big] - \mathbb{E}_{z\sim p(Z)}\big[D_{KL}(p(z)||q(z))\big] \\ &+ \mathbb{E}_{z\sim p(Z)}\big[\log q(z)\big] \\ &\leq \mathbb{E}_{(x,z)\sim p(Z|X)p(X)}\big[\log p(z|X=x;\boldsymbol{\theta})\big] + \mathbb{E}_{z\sim p(Z)}\big[\log q(z)\big] \\ &= \mathbb{E}_{(x,z)\sim p(Z|X)p(X)}\big[D_{KL}(p(z|X=x;\boldsymbol{\theta})||q(z))\big] \end{aligned} \tag{2.11}$$

Traditionally, the choice of $q(z)$ falls to a standard Gaussian $\mathcal{N}(Z|0, I)$. By putting 2.9 and 2.11 together we obtain the following training objective for VIB:

$$L(p(X,Y),\boldsymbol{\theta},\beta) = J_{CE}(p(X,Y);\boldsymbol{\theta}) - \beta\mathbb{E}_{(x,z)\sim q(Z|X)p(X)}\big[D_{KL}(q(z|X=x;\boldsymbol{\theta})||q(z)\big] \tag{2.12}$$

We can model the encoder $q(Z|X)$ to correspond to output the minimal sufficient statistics for the Gaussian distribution $\mathcal{N}(Z|f_e^{\mu}(X), f_e^{\sigma}(X))$. To avoid backpropagation through the sampling operation, we can use the reparametrization trick (Kingma and Welling, 2014) to introduce the noise term that is independent of the parameters of the model $\boldsymbol{\theta}$. Using the experimentation results provided in (Alemi et al., 2016) authors conclude that VIB works as a decent regularizer in and of itself.

2.5 Learning tighter bounds via Superfluous information minimization

But can we do better? What if we learn the posterior distribution $p(Z|X;\boldsymbol{\theta})$ modeled by the encoder while being regularized to fit more complex variational prior $q(Z)$ rather than just Unit Gaussian?

The idea of modeling complex multi-modal distributions has found a wide range of applications in deep learning using Normalizing Flows (Rezende and Mohamed, 2015, Dinh et al., 2016). We won't go far in this topic, as the focus of this thesis is mainly in supervised representation learning. Anyways, it's worth saying that such approach gained some success in finding complex structure in the real-world data, although it constrains the variety of architecture choices, due to the requirement of all transformations to be invertible, meaning that we cannot directly represent data in a lower-dimensional, compressed space.

As the task labels are available during the training time we could infer the information about the partition of the compressed space under certain regions being occupied by only associating points of the correspondent target class. We can see this as modeling the embedding space to be characterized by a mixture of distributions. Namely, if we model these distributions to be Gaussian we obtain the mixture of Gaussians (Chapter 9 from Bishop, 2006).

How can we achieve this in the VIB set-up? By introducing additional variational distribution modeled by the "inverse" decoder $q(Z|Y)$ we can infer the representation of labels directly. Thus, we model another distribution $\mathcal{N}(Z|f_d^{\mu}(Y), f_d^{\sigma}(Y))$, where $f_d^{\mu}(Y)$ and $f_d^{\sigma}(Y)$ are sufficient statistics of the approximation to the Gaussian conditional prior. We believe that learning more flexible regularizing distributions might lead to improved generalization capability of the representations produced.

This switches our focus from minimizing the KL-divergence in 2.12 to a modified lower bound $D_{KL}(p(Z|X)||q(Z|Y)$. We again said lower bound not to no purpose. In turn, following the same idea of introducing variational approximation as in 2.11, it can be reformulated as follows:

$$D_{KL}(p(Z|X)||q(Z|Y)) = I(X;Z|Y) + D_{KL}(p(Z|Y)||q(Z|Y)), \tag{2.13}$$

where conditional mutual information $I(X;Z|Y)$ is often referred to as the superfluous information. It measures the amount of information conveyed in representations that cannot be used to learn predicting the task. Hence, preserving this information only increases the complexity of representations making the intention to get rid of it reasonable.

More precise inspection of this can be done by subdividing $I(X;Z)$ using the chain rule of mutual information:

$$I(X;Z) = I(X;Z|Y) + I(X;Y) - I(X;Y|Z). \tag{2.14}$$

Note that the predictive information, $I(X;Y)$, is constant and defined by the dataset, while the amount of predictive information not in Z, $I(X;Y|Z)$, has to be 0 for sufficient representations. Thus, the minimization of $I(X;Z)$ is equal to minimizing $I(X;Z|Y)$ for sufficient representations. According to Proposition 2.1 in (Federici et al., 2020), sufficient representation is minimal whenever $I(X;Z|Y)$ is minimal. This means that by directly optimizing superfluous information we can obtain representations presumed to be more optimal in terms of data compression and information content.

Furthermore, following the chain rule of mutual information, superfluous information in the general case can be defined as:

$$I(X;Z|Y) = I(X;Z) + H(Y|X) + H(Y|Z) - H(Y|X,Z) - H(Y). \tag{2.15}$$

Note that given the Markov condition $Y \leftrightarrow X \leftrightarrow Z$, Y and Z are conditional independent when X is observed. This means that once we know values of X, having additional information on the values of Z will not decrease the uncertainty about the values of Y. Hence, $H(Y|X,Z) = H(Y|X)$. Substituting this into 2.15 gives us:

$$\begin{aligned} I(X;Z|Y) &= I(X;Z) + H(Y|X) + H(Y|Z) - H(Y|X) - H(Y) \\ &= I(X;Z) + H(Y|Z) - H(Y) \\ &= I(X;Z) - I(Z;Y). \end{aligned} \tag{2.16}$$

In turn, this lets us to assume that for sufficient representations superfluous information should be minimal whenever for the desired level of compression the representation is maximally expressive about the task.

Given all above, the final modified optimization objective can be given as follows:

$$\begin{aligned} L(p(X,Y),\boldsymbol{\theta},\beta) = J_{CE}(p(X,Y);\boldsymbol{\theta}) \\ - \beta\mathbb{E}_{(x,y,z)\sim p(Y|X)p(Z|X)p(X)}\left[D_{KL}(p(z|X=x)||q(z|Y=y)\right] \end{aligned} \tag{2.17}$$

This approach can be stated to be novel as opposed to previously presented in the literature. Worth noting that we came up with such an idea during the intermediate phase of work on this thesis. We name such a method for supervised representation learning as the Conditional Variational Information Bottleneck (CVIB). However, while working on its development we found out the concurrent work on the Conditional Entropy Bottleneck (Fischer, 2019). Discouraging to us, authors also aimed to focus on minimizing superfluous information, although their motivation relies on the defined Minimum Necessary Information (MNI) criterion rather than on the improvement of regularizing priors. On top of that, they only aim to learn the mean modeled by $f_d^{\mu}(Y)$ while keeping the variance approximation $f_d^{\sigma}(Y)$ to be unit, i.e. fixed and not learned at all. We study the comparison of these two training methods and report correspondent results and discussion in Chapter 4.

3 Validation through estimation and evaluation

In this chapter, we discuss more thoroughly the concept of mutual information, issues, and difficulties that occurred while working with it and ways to overcome it. Also, we address the problem of validation of information-theoretic objectives for extracting robust features from the data evaluating the quality of representation learning algorithms in a unified information-theoretic framework.

3.1 Mutual Information Estimation

Recall that the mutual information (MI) can be thought of as a measure of dependency between probability distributions. Namely, it can be seen as a cost we have to pay when treating probability distributions being independent when they are actually not. This measure is reparametrization-invariant and is formalized as follows:

$$I(X;Y) = D_{KL}(p(X,Y)||p(X) \times p(Y)) = \mathbb{E}_{(x,y)\sim p(X,Y)}[\log \frac{p(x,y)}{p(x)p(y)}] \tag{3.1}$$

Usually, in the real-world machine learning tasks we don't have a direct access to the probability distributions but the samples from the joint distribution stored in a dataset. This means that in the end we cannot compute exactly but only estimate the true value of it (Paninski, 2003). In addition, approaching the estimation in high-dimensional spaces directly is itself a notoriously difficult task - computing expectation in Equation 3.1 involves an intractable integral. Nevertheless, given the Donsker-Varadhan dual representation of the Kullback-Leibler divergence we can establish a connection between the real value of mutual information with a tractable lower-bound:

$$\begin{aligned} I(X;Y) &= D_{KL}(p(X,Y)||p(X) \times p(Y)) \\ &= \sup_{T:\mathcal{X}\times\mathcal{Y}\to\mathbb{R}} [\mathbb{E}_{(x,y)\sim p(X,Y)}[T(x,y)] - \log \mathbb{E}_{x\sim p(X),y\sim p(Y)}[e^{T(x,y)}]], \end{aligned} \tag{3.2}$$

where $T(\cdot,\cdot)$ represents the family of approximation functions that can be parametrized by the neural network. Thus, by maximizing the lower-bound with respect to the parameters of $T(\cdot,\cdot)$, we get ourselves closer to the more accurate estimate of MI. Based on this, the idea of optimizing variational bounds on the estimate has been plausible for the research community, resulting in a wide area of research for discovering variational bounds on MI (Poole et al., 2019, Belghazi et al., 2018 etc.).

Many algorithms and methods involve direct optimization of the estimate (e.g InfoMax principle and it's derivatives (Hjelm et al., 2018, Löwe et al., 2019 etc.)). Furthermore, such an approach is generally computationally inefficient and hard. Due to the high-bias and high-variance nature of the majority of estimators, getting accurate estimates might require putting too much effort into fine-tuning weights of the estimator together with parameters of the encoder during the training process of the target encoding procedure. Undoubtedly, using knowingly bad estimators may lead to both strong biases in training and confusion in reasoning regarding the comparative strength of different ways to compress data. In our case, we did not suffer from any issues related to the bias that occurred in estimations for MI during the training of the target neural network to compress the input, because we used MI estimation disregard of the training of the encoder itself. As the goal of our research was to make one being able to reason about the quality of different encoding procedures, we trained the estimator only once - for fixed and fully-trained encoders, since neither of the algorithms studied involved direct estimation for the mutual information, but rather information-theoretic bounds provided in analytical form.

For the scope of this thesis we make a choice in a favour of the I_{JS} estimator proposed in Poole et al., 2019. According to authors it can provide unbiased evaluations on the estimate of mutual information while being independent of the batch size, as well as, having tractable both objective and corresponding gradient. Moreover, interesting fact, that for evaluation of the estimate of MI using I_{JS}, the I_{NWJ} estimator's (Nguyen et al., 2008) objective is used and formalized as follows:

$$I_{NWJ} = \mathbb{E}_{(x,y)\sim p(X,Y)}\left[\log T(x,y)\right] - \frac{1}{e}\mathbb{E}_{x\sim p(X),y\sim p(Y)}\left[T(x,y)\right] \tag{3.3}$$

Nevertheless, for the update of the estimator's network parameters, the values of the gradient of Equation 3.3 are used. The latter has the form of the Jenson-Shannon divergence used as an objective for f-GANs (Nowozin et al., 2016) and results in the following optimization objective:

$$I_{JS} = \mathbb{E}_{(x,y)\sim p(X,Y)}\left[T(x,y)\right] - \mathbb{E}_{x\sim p(X),y\sim p(Y)}\left[e^{T(x,y)}\right] + 1 \tag{3.4}$$

Basically, in such formulation, we end up with updating parameters of the estimator using the second derivative of the I_{NWJ} objective, as both Equations 3.3 and 3.4 reach the same optimum. In turn, this leads to an improvement of stability of the estimator's entire training process, since the target gradients are easier to compute. Unfortunately, I_{JS} still suffers from the problem of high variance. Because of this, we face two issues we had to teach ourselves to solve. First: it is hard to define unified convergence criteria for training process termination. We overcome this by keeping the number of training epochs for estimator fixed, meaning that in Section 4.3 we provide estimations for mutual information as evaluations for the amount of information that could be recognized under same hyperparameter setting of the MI estimator for encoders of our interest for comparison. Second: for some batches, we might get outlying values of estimations resulting in a high level of noise present in the final evaluations per epoch. Applying pruning of the $k\%$ of the highest and lowest values for each training epoch before averaging across batches helped to ensure that the estimates we get are still unbiased and close enough to the truth.

To improve the significance and the quality of the MI estimation, we perform optimizations over multiple seeds (3 in our case) and then report average evaluations. However, this causes a sizeable increase in the computational and time effort needed to take advantage of the framework, as getting estimates for the mutual information is the most time-consuming part of the entire validation procedure. However, time and, hence, computational complexity is not the only limitation associated with the task of mutual information estimation.

The first and the main limitation for estimating the mutual information is the data "hungriness" - any distribution-free high-confidence bound requires an exponential sample size for the size of the bound (McAllester and Stratos, 2020). So having a batches of size N we can only capture $\mathcal{O}(\log N)$ bits of information, no matter what is the actual amount of information conveyed in representations. This makes high-quality estimation hard in tasks when the dataset available is not big enough or inputs and task labels are generally very statistical dependent (when $I(X;Y)$ is high, so representations can preserve a lot of attributes from the data - more than could be recognized by the estimator).

Secondly, most of the existing estimators are very sensitive to the hyperparameter's choice. Hence, building reliable estimators may require doing the hyperparameter's tuning precisely. We report our hyperparameters set-up in the Appendix.

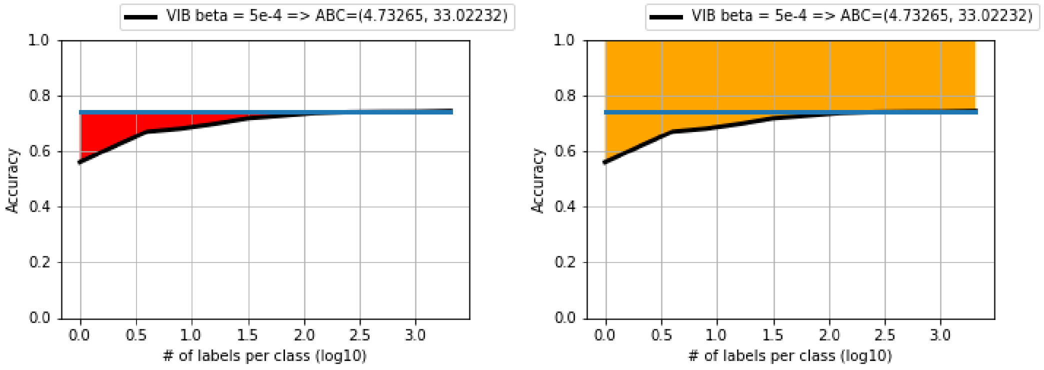
Even though most of the methods investigated in this thesis do not require estimating mutual information directly during training, getting estimates for the value of mutual information is extremely helpful for evaluation of the compressing capability of deep neural networks. In turn, we do take advantage for it to show that discarding irrelevant attributes of the data leads to much higher compression rate and improved generalization as opposed to standard regularization techniques.

3.2 Modifying the linear evaluation protocol

To assess the generalization capability of the models we use a custom modification of the linear evaluation protocol. The core idea is to build a linear classifier (e.g. logistic regression) on top of (frozen) representations and then evaluate the accuracy of solutions for the downstream task (Kolesnikov et al., 2019). In turn, we extend this metric to measure the residual performance given a different amount of additional training examples. We believe it to be extremely useful for the evaluation of solutions to problems when obtaining extra data might be expensive. The goal is to get to know the estimated performance gap between maximum achievable performance (accuracy of classifier fine-tuned using the entire training set) across cases with different amount of labels per class. This simulates the case when the amount of additional data provided is cut-off. We can put all these gaps together by evaluating the area between two curves: first - the accuracy curve for a different amount of labels available, and second - the level of maximum achievable performance. We give a name for this metric of Area Between Curves or, simply, ABC loss. Thus, models that can encode data without the loss of its expressiveness while needing less additional examples to achieve requested performance correspond to small values of ABC loss, while those not able to generalize well correspond to higher values of ABC.

Worth mentioning is that logistic regression has a closed-form solution allowing to access the optimal solution for the weights fast. This property of logistic regression appears to be extremely useful, as it gives us an opportunity to increase the significance of results by averaging across an arbitrary amount of random seeds without a high loss of time-efficiency, unlike the case for mutual information estimation.

The drawback of such an evaluation approach is that we might obtain the same values of the ABC loss for encoders of different strength and a level of quality. Hence, we need to incorporate these loss values with something else to distinguish models of bad initial (end-to-end) performance. While having accuracy values for every label partition available, we can measure the gap between the accuracy curve and the maximum performance ever possible - 100% accuracy. Having both of these values modifies the proposed metric to output not a single value, but a tuple consisting of two values. Visual representation of the gaps mentioned can be found in Figure 3.1. Such an approach allows us to state that model A is superior to model B for generalization, if for model A the first value in the ABC loss tuple is substantially



(a) Max achievable performance gap

(b) Max possible performance gap

Fig. 3.1: Visualization of the areas defining ABC loss

smaller than the one for model B while having second values to be more or less equal.

Of course, using such an approach for generalization evaluation implies having some amount of underlying variance we should either prevent or take into account. First possible source - weights initialization of the target linear classifier. To deal with this question properly we presume to keep initializations pre-defined and fixed for every time we set-up the evaluation model. Also, a certain amount of additional variance arises due to the sampling procedure in the compressed space - representations to evaluate are drawn as samples from the posterior distribution $p(Z|X)$. Using a single Monte Carlo sample might produce an intolerable level of noise in the evaluation. Empirically we figured out that using an average of 64 posterior samples for training logistic regression and using the likeliest embedding (posterior mean) during the test time produces the best capability for detecting generalization quality of representations[1].

In such a way, we provide a unified tool to quantitatively reason about the generalization capability of the models studied in this work.

[1] By this we refer to an ability to fairly treat knowingly better encoders with correspondingly low values of ABC, and knowingly bad - with high.

4 Experimental setup, Results and Discussion

4.1 Datasets

To validate the hypothesis outlined in Section 1.3 we investigate training deep neural models for image classification using 2 publicly available datasets.

Firstly, we perform a comprehensive benchmarking to compare the impact of the different design choices covered in this work we use the MNIST12k dataset. Each model is trained and evaluated on the version of the original MNIST (LeCun et al., 2010) dataset, which consists of 12000 training images and 50000 test images of hand-written digits. The motivation for such a choice is that by doing so we decrease the number of training labels available during the training time, thus, weakening the models to make spotting the desired behavior easier.

Further, we expand our analysis to a more complex task of predicting classes for real-world images using the CIFAR10 (Krizhevsky, 2009) dataset. Here we use the original version consisting of 50000 training and 10000 testing examples each corresponding to one of the following categories: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck.

In our conclusions we mostly rely on results for solving problems on CIFAR10 as the task of classifying real-world objects is generally more complex than hand-written images and we believe it to be more expressive about the true state of affairs. Nevertheless, results obtained for MNIST12k do not contradict general discussion. However, for MNIST12k experiments all models achieve near-perfect performance making hard to reason about superiority of one method over another due to possible insignificance of results caused by implicit randomness of training procedures. We are aware of this and other possible issues regarding variance of results and provide discussion on this in Section 5.1.

4.2 Models overview

To provide a comprehensive comparison we consider 4 major types of models: deterministic and stochastic neural networks trained with the application of common regularization techniques (weight decay (Krogh and Hertz, 1992) and dropout (Srivastava et al., 2014)), stochastic neural networks trained using VIB and CVIB. Investigation of our method, CVIB, involves the study of different extents of flexibility for the learned conditional prior (e.g. to learn the variance or to keep it fixed).

To operate with MNIST12k we adopt the architecture used in (Alemi et al., 2016). Namely, we model posterior distributions using MLPs of the form of encoder 784 - 1024 - 1024 - $2K$ for stochastic encoders and 784 - 1024 - 1024 - K for deterministic, where K is the size of the bottleneck, i.e dimensionality of the compressed space. Note that in stochastic variant last encoding layer has twice more parameters (first K outputs encode μ, while the remaining K outputs after softplus transformation, to ensure non-negativity, encode the diagonal of the covariance matrix Σ[1], denoted as σ) than its deterministic analog (intermediate representations are then characterized by a single embedding). The variational decoder is modeled by two successive fully-connected hidden layers both of size 64, resulting in the variational decoder of the form K - 64 - 64 - 10. We use a more complicated form of the decoder as opposed to the one used in (Alemi et al., 2016) on purpose, to learn high-level representations in not necessarily linearly separable space, since the usage of simple logistic regression is involved in the adaptation of linear evaluation protocol described in Section 3.2. So to avoid enforcing linear separability of the compressed space by construction, we do not use the same linear model on the down-stream task twice.

For learning on CIFAR10 we adopt the encoder to replicate the architecture of VGG11: 8 convolutional layers, each of which is applied in combination with the successive max-pooling operation followed by the activation, with one last dense layer to enter the embedding space. The architecture of the variational decoder was chosen to be consistent with the MNIST12k case described above: mapping of the form K - 64 - 64 - 10.

Across all experiments, we keep K=256. On top of that, there is something that is in common for all models investigated - in every case we use ReLU activations. Single-sided saturating nonlinearities, such as ReLUs, unlike $\tanh$, do not yield any compression as neural activations do not enter the saturating regime (Saxe et al., 2018). Hence, we do so, so not to induce any additional compression caused by the choice of architecture (unless specified explicitly).

[1] We presume features to be independent and having covariance being equal to 0.

Note that in neither case batch normalization was not used and was turned off on purpose in cases where using it is presumed by default (e.g. when using VGG11-motivated encoder). Same as for activations, we do so in order not to induce any additional generalization improvement.

For the study of CVIB for both tasks we introduce additional model of "backward" decoder that we set to be a linear mapping from label space to space of representations. Specifically, we create an additional model consisting of single fully connected layer, that takes target labels as input and produces corresponding embedding in the compressed space of representations. This model has its parameters fitted by the same optimizer as for the encoder network simultaneously during the training time.

As an architecture for the neural mutual information estimator, we use an MLP of the form $(S_1 + S_2)$ - 1024 - 1024 - 1, where S_1 and S_2 are sizes of flattened feature vectors of variables we aim to measure the statistical dependency between concatenated across the feature dimension.

All models were built using PyTorch framework (Paszke et al., 2017) and optimized using ADAM optimizer (Kingma and Ba, 2014). Detailed information regarding values of hyperparameters used in this work that were not directly related to the main focus of our research can be found in the Appendix.

4.3 Experiments

In this section we provide an experimental justification for the expediency to use the Information Bottleneck principle. We build our reasoning around the capability of various types of deep neural networks to compress the data into representations without the loss of its expressiveness and ability to generalize.

Specifically, the goal of all experiments is to discover the effect of different regularization techniques and information-theoretic based training objectives on compressing ability of the encoding procedure with respect to capability of sustaining generalization according to the ABC loss defined in Section 3.2. By this we aim to validate the Information Bottleneck principle and to show the ability of methods, derived from original IB (VIB, CVIB), not only to decrease the complexity of representations but to do this in a manner to actually increase their quality.

Since the achievement of the desired compression rate cannot be guaranteed in advance, we perform several optimizations for different values of parameters that stand for the amount of regularization applied. Namely, we build experiments over the set of parameters values of interest that we pre-define for every case considered. Note that the choice for the range of values for weight decay, dropout rate and trade-off parameter β might differ depending on the dataset, since different architectures were used to solve different tasks. We aim to push the expressiveness of our investigation to the limit by discovering models behaviour via gradually increasing hyperparameters values up until the point of information collapse.[2]

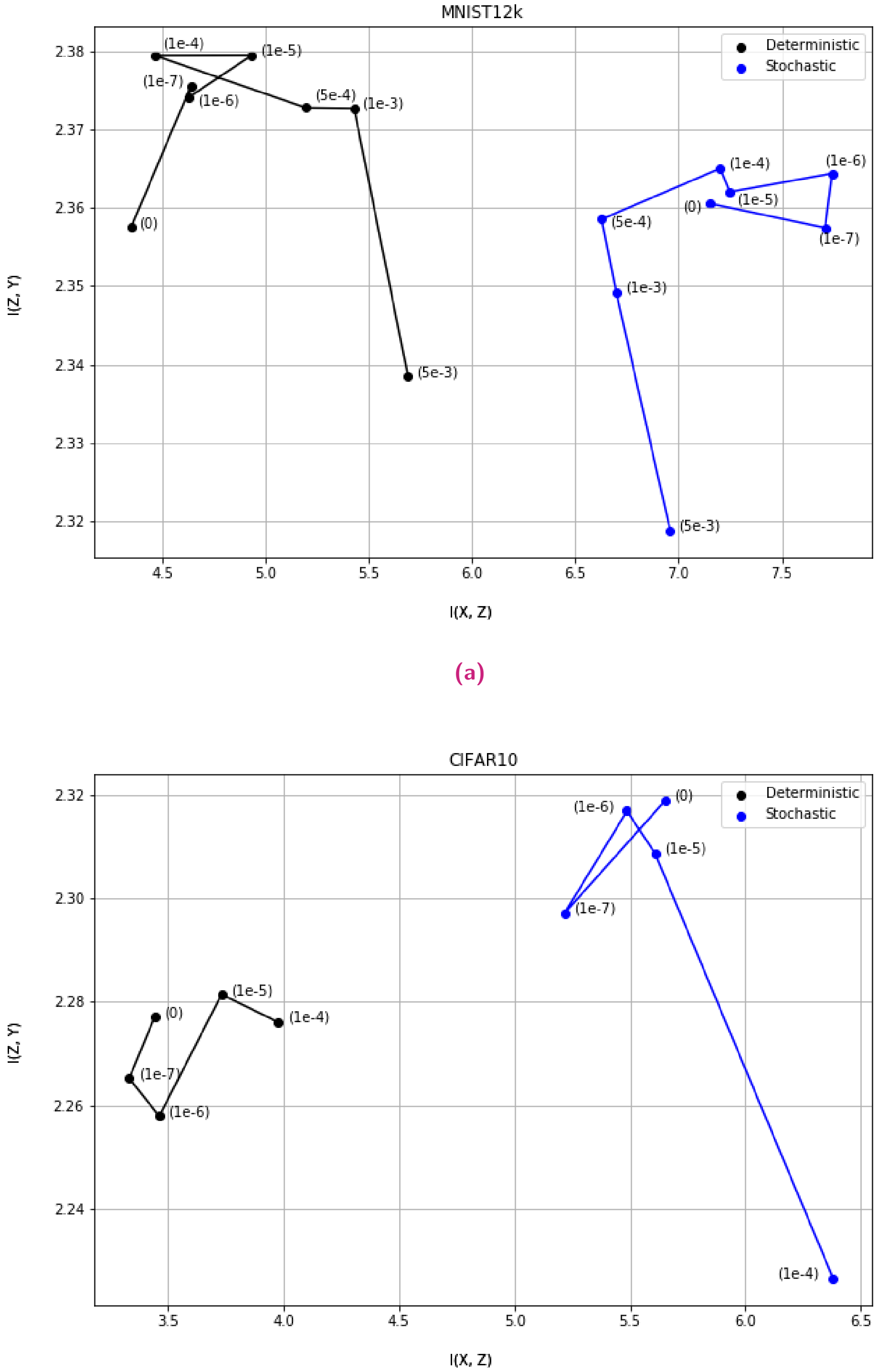
4.3.1 Measuring the effect of standard regularization techniques on the complexity of representations

Traditionally, the common choice for benchmarking objects of comparison to the IB-motivated encoders is made in a favour of simple deterministic encoders. Note that using VIB-based training procedure implies the stochastic nature of the encoder, since we aim to build a distribution over the compressed space and then provide representations as samples from it. To validate the idea of incorporating information-theoretic concepts to training deep neural networks, we first want to establish the relationship between the stochastic and deterministic natures of encoding procedures in terms of the ability to sustain the generalization for the compression rate achieved. We train stochastic and deterministic encoders trained under various levels of regularization applied. The phenomenon that we aim to observe can be described as increasing the strength of regularization should make the model compress the data more when compared to the completely unregularized versions of encoding procedures studied. Moreover, we expect the compression rate to increase monotonically when using higher values of weight decay or higher dropout rate.

First of all, in Figure 4.1 we present the study of the behavior of stochastic and deterministic encoders trained under various levels of weight decay regularization applied. Annotations present in the plot correspond to the value of the regularization hyperparameter (weight decay value in this case, dropout rate and trade-off parameter β respectively in later sections). Generally, usage of weight decay (also known as L_2 regularization) puts certain constraints on the values of the weights preventing the model from treating certain features much more important than others. Unexpectedly to us, given the information curves[3] we don't see any related compression for weight decay unless the information collapses (for overregularized model set-up) neither for MNIST12k (Figure 4.1a) nor for CIFAR10 (Figure 4.1b). Moreover, our results show that increasing the value of weight decay on average increases the complexity of representations. Need to say that such behavior was counterintuitive to us. At first, we supposed that the reason for this might have been

[2] Numerically this corresponds to the case of 0 mutual information preserved and the value of ABC loss of (0, 100).

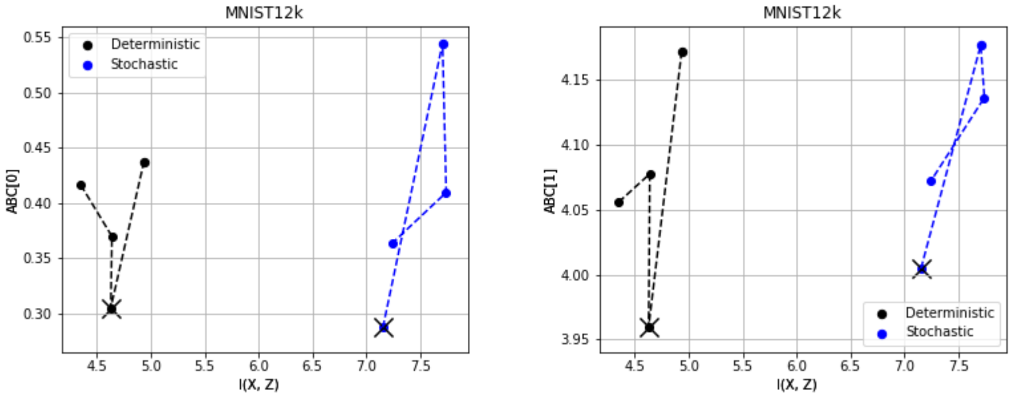
[3] By this here and further we refer to curves for mutual information estimation built for different hyperparameter values of our interest.



(a)

(b)

Fig. 4.1: Information curves for stochastic and deterministic encoders regularized via **weight decay**



(a) Max achievable performance gap (b) Max possible performance gap

Fig. 4.2: Generalization vs compression comparison via visualization of ABC loss values for $I(X;Z)$ for encoders regularized using **weight decay** on MNIST12k (black crosses denote points corresponding to the best generalization)

caused due to the problem of the scale of representations to the scale of initializations of weights of estimator degrading the quality of mutual information estimation. We tried to overcome this problem by leveraging the property of scale invariance of mutual information - for weight decay mutual information estimation was done for the whitened distribution of representations. However, it couldn't provide us with evidence of the phenomenon we expected. Given this, we conclude that usage of weight decay regularization does not imply any compulsory compression of the data.

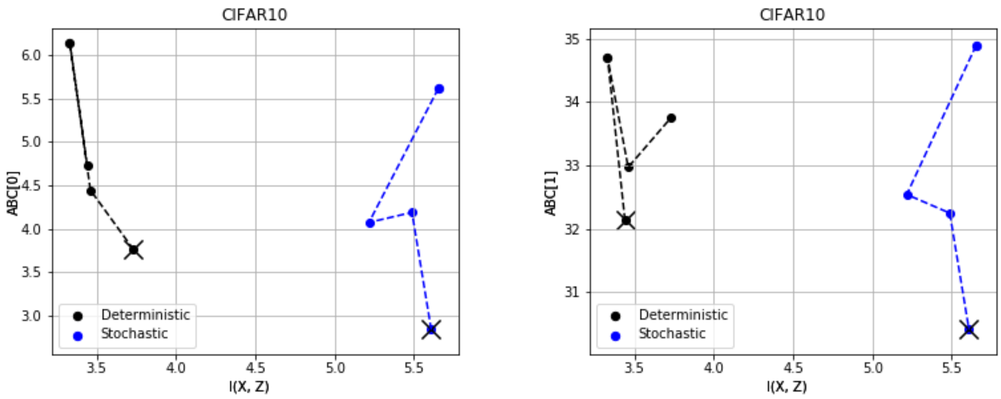
Nevertheless, the study of the generalization capability of such encoders, presented in Table 4.1, shows that in 3 out of 4 cases the usage of weight decay could improve the generalization capability of encoders.[4] Visualizations for corresponding ABC tuples, present in the Figures 4.2 and 4.3 show that the best generalization performance for both datasets (see Figures 4.2a and 4.3a) was achieved by the stochastic variant of the corresponding neural network. However, such improvement was achieved by the noticeable increase in the complexity of representations, meaning that the compression ability for such networks is worse as deterministic nature of the encoder ceases to convey a lot of attributes from the data.

On the other hand, when using dropout regularization the trend for correspondent compression of the data can be seen - increasing the dropout rate decreases the complexity of representations (Figure 4.4). However, there is no clear relation between compression and improvement of generalization. MNIST12k results, presented in Table 4.3, show that according to our target metric best generalization performance was obtained for unregularized versions of encoders, even though the compression is done without tangible loss of expressiveness in terms of information shared about

[4] Bold numbers correspond to the best model set-up concerning generalization capability via ABC loss

ABC on MNIST12k		
Weight decay value	Stochastic	Deterministic
0	**(0.287, 4.004)**	(0.416, 4.056)
1e-7	(0.544, 4.177)	(0.370, 4.077)
1e-6	(0.409, 4.136)	(0.305, 3.96)
1e-5	(0.364, 4.072)	(0.436, 4.172)
5e-5	(0.474, 4.369)	(0.299, 4.043)
1e-4	(0.686, 4.878)	**(0.266, 4.230)**
5e-4	(0.640, 5.252)	(0.561, 5.426)
1e-3	(1.598, 6.207)	(0.287, 4.977)
5e-3	(2.965, 8.631)	(0.744, 7)

Tab. 4.1: Results for generalization capability via ABC loss for encoders trained using **weight decay** on MNIST12k



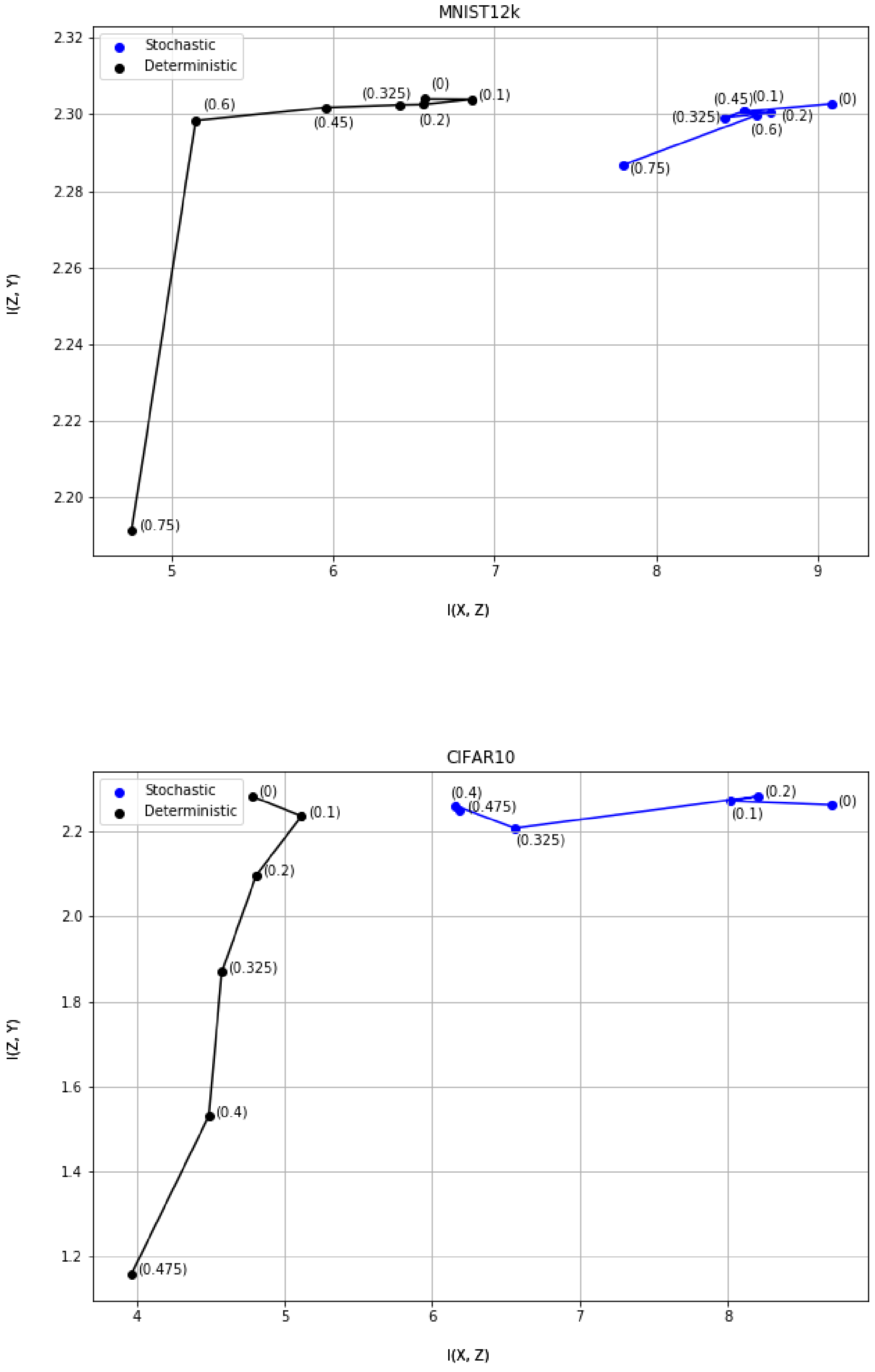
(a) Max achievable performance gap

(b) Max possible performance gap

Fig. 4.3: Generalization vs compression comparison via visualization of ABC loss values for $I(X;Z)$ for encoders regularized using **weight decay** on CIFAR10 (black crosses denote points corresponding to the best generalization)

ABC on CIFAR10		
Weight decay value	Stochastic	Deterministic
0	(5.618, 34.890)	(4.731, 32.132)
1e-7	(4.071, 32.540)	(6.143, 34.688)
1e-6	(4.187, 32.247)	(4.445, 32.970)
1e-5	(2.834, 30.410)	(3.762, 33.750)
5e-5	**(2.094, 31.767)**	(3.595, 29.605)
7.5e-5	(3.671, 35.436)	**(2.700, 30.812)**
1e-4	(3.873, 36.295)	(3.326, 31.804)

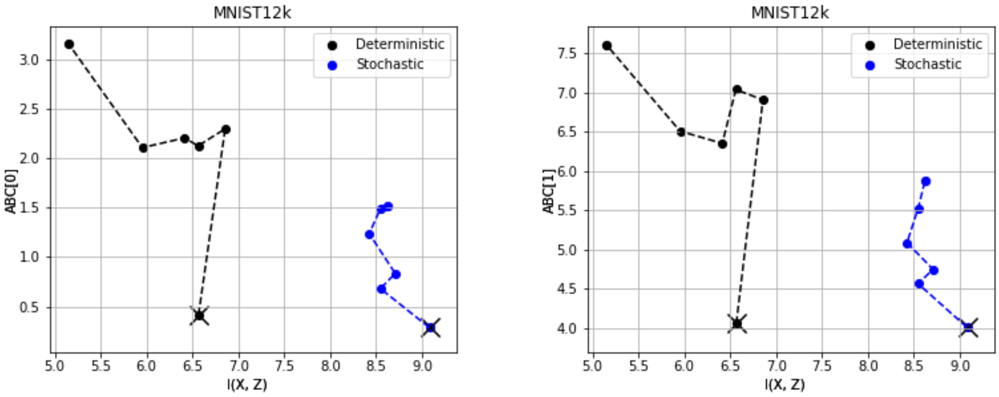
Tab. 4.2: Results for generalization capability via ABC loss for encoders trained using **weight decay** on CIFAR10



(a)

(b)

Fig. 4.4: Information curves for stochastic and deterministic encoders regularized via **dropout**



(a) Max achievable performance gap

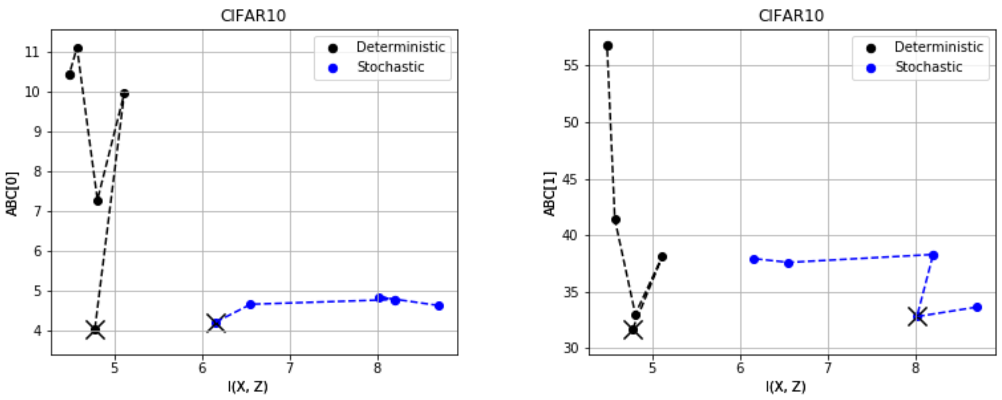
(b) Max possible performance gap

Fig. 4.5: Generalization vs compression comparison via visualization of ABC loss values for $I(X; Z)$ for encoders regularized using **dropout** on MNIST12k (black crosses denote points corresponding to the best generalization)

ABC on MNIST12k		
Dropout rate	Stochastic	Deterministic
0	**(0.287, 4.004)**	**(0.416, 4.056)**
0.1	(0.682, 4.562)	(2.298, 6.904)
0.2	(0.835, 4.747)	(2.126, 7.036)
0.325	(1.238, 5.084)	(2.206, 6.350)
0.45	(1.489, 5.521)	(2.108, 6.506)
0.6	(1.512, 5.874)	(3.157, 7.606)
0.75	(7.210, 16.757)	(13.997, 34.902)

Tab. 4.3: Results for generalization capability via ABC loss for encoders trained using **dropout** on MNIST12k

the task ($I(Z; Y)$). For CIFAR10 the results are even more interesting (Figures 4.4b and 4.6, Table 4.4). As the main concern of this section is to establish the relationship between stochastic and deterministic natures of neural networks, we see that given the choice of the nature using dropout directly affects the amount of task-related information $I(Z; Y)$ conveyed in representations for the case of deterministic neural encoders, while for stochastic it sustains desirable level. Moreover, values of ABC loss presented in Table 4.4, as well as generalization versus compression curves in Figure 4.6, confirm this - increasing the dropout rate decreases both an ability to generalize and the strength of the neural network itself (see the increase of the maximum performance gap). We hypothesize that such behavior is evidence of the fact that randomly discarding features degrades the quality of embeddings, but not the quality of the probability distribution built over the space of compressed representations. However, it is hard to say that such behavior must necessarily hold in the general case, as it requires a much more thorough investigation both experimentally and theoretically.



(a) Max achievable performance gap

(b) Max possible performance gap

Fig. 4.6: Generalization vs compression comparison via visualization of ABC loss values for $I(X;Z)$ for encoders regularized using **dropout** on CIFAR10 (black crosses denote points corresponding to the best generalization)

ABC on CIFAR10		
Dropout rate	Stochastic	Deterministic
0	(4.633, 33.625)	**(4.048, 31.704)**
0.1	(4.849, 32.785)	(9.964, 38.090)
0.2	(4.782, 38.284)	(7.287, 33.005)
0.325	(4.664, 37.584)	(11.116, 41.401)
0.4	**(4.213, 37.919)**	(10.434, 56.775)
0.475	(4.439, 45.430)	(0.618, 94.544)

Tab. 4.4: Results for generalization capability via ABC loss for encoders trained using **dropout** on CIFAR10

To sum up, it is hard to unequivocally reason about the generalization capability and ability to safely compress data (without loss of expressiveness) for neural networks trained using commonly used regularization techniques. However, in both cases the trend is obvious - introducing stochasticity in the architecture induces the increase in complexity of representations.

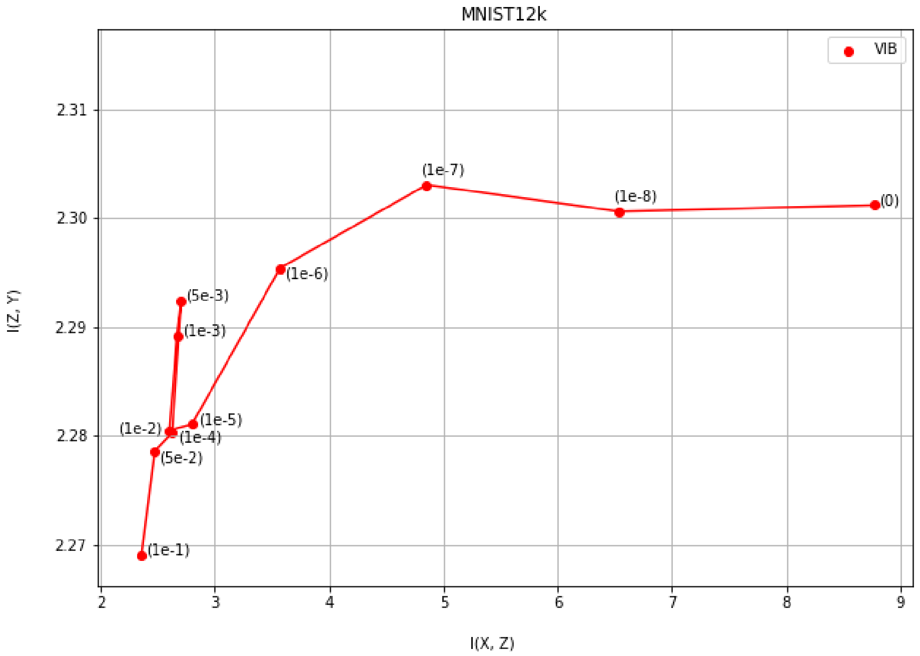
Hence, we can conclude that building probability distribution over the space of representations is more expressive and conveys more attributes of the data rather than via deterministic mappings. This result is important as it provides intuition on how to fairly treat the compressing strength of different representation learning algorithms that require the usage of stochastic nature when opposed to deterministic commonly-used benchmarks (e.g Vanilla MLPs).

4.3.2 Validating the ability of the Variational Information Bottleneck to compress without loss of generalization

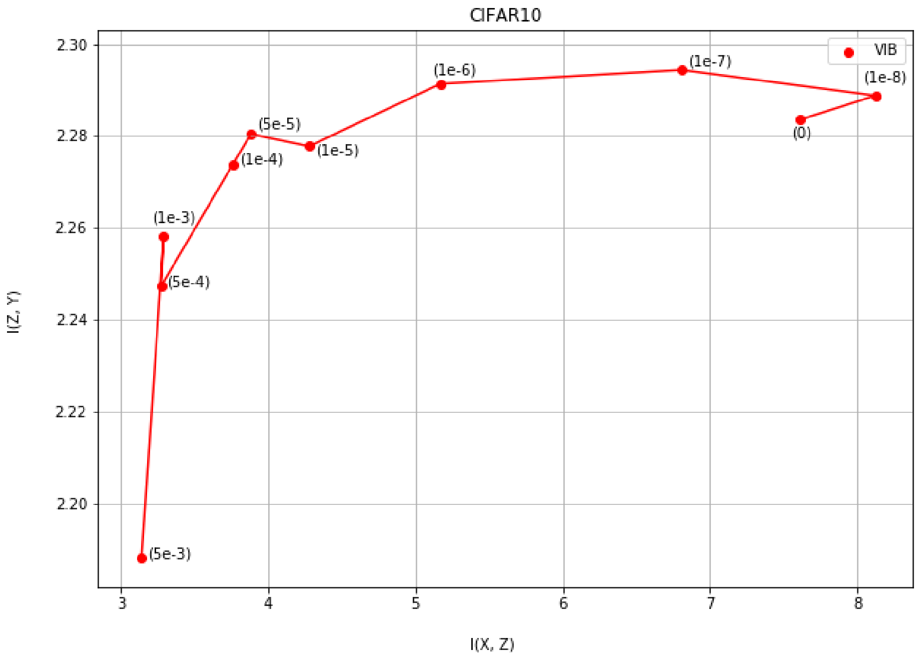
Further, we discover the effect of applying the adopted using variational inference version of the information bottleneck principle, described in Section 2.4 and referred to as VIB, to capture meaningful representations using stochastic neural networks. As it is focused explicitly on minimizing the mutual information between representation and input, we expect to spectate stronger compression as compared to the ones obtained in the previous section for the usage of weight decay and dropout. Moreover, in (Alemi et al., 2016) the authors provided results showing that the VIB model could outperform other regularization techniques in terms of misclassification rate. According to their results regarding the superiority of the VIB model, we expect it to show better generalization capability.

Information curves in Figure 4.7 show an undeniable trend for compression with the increase of the trade-off parameter β for both tasks investigated. Furthermore, correspondent values of ABC loss (see VIB column in Tables 4.5 and 4.6) show that by incorporating the Information Bottleneck principle we could not only compress the data effectively but indeed improve the generalization capabilities of representations. Namely, for MNIST12k we could improve the model's ability to generalize by 2.5 times, while for CIFAR10 we could do it by more than the factor of 5. Note that in both cases second values in ABC tuple, that stand for the maximum possible performance gap, got substantially improved, meaning that we not only could improve the generalization for representations but also the overall strength of the model.[5]

[5] It is worth acknowledging that this value has direct correlation with the prediction accuracy for the model trained in the end-to-end manner - higher initial accuracy (and, hence, strength) implies higher accuracy for the linear predictor fine-tuned using the entire finite sample.



(a)



(b)

Fig. 4.7: Information curves for the **VIB** model

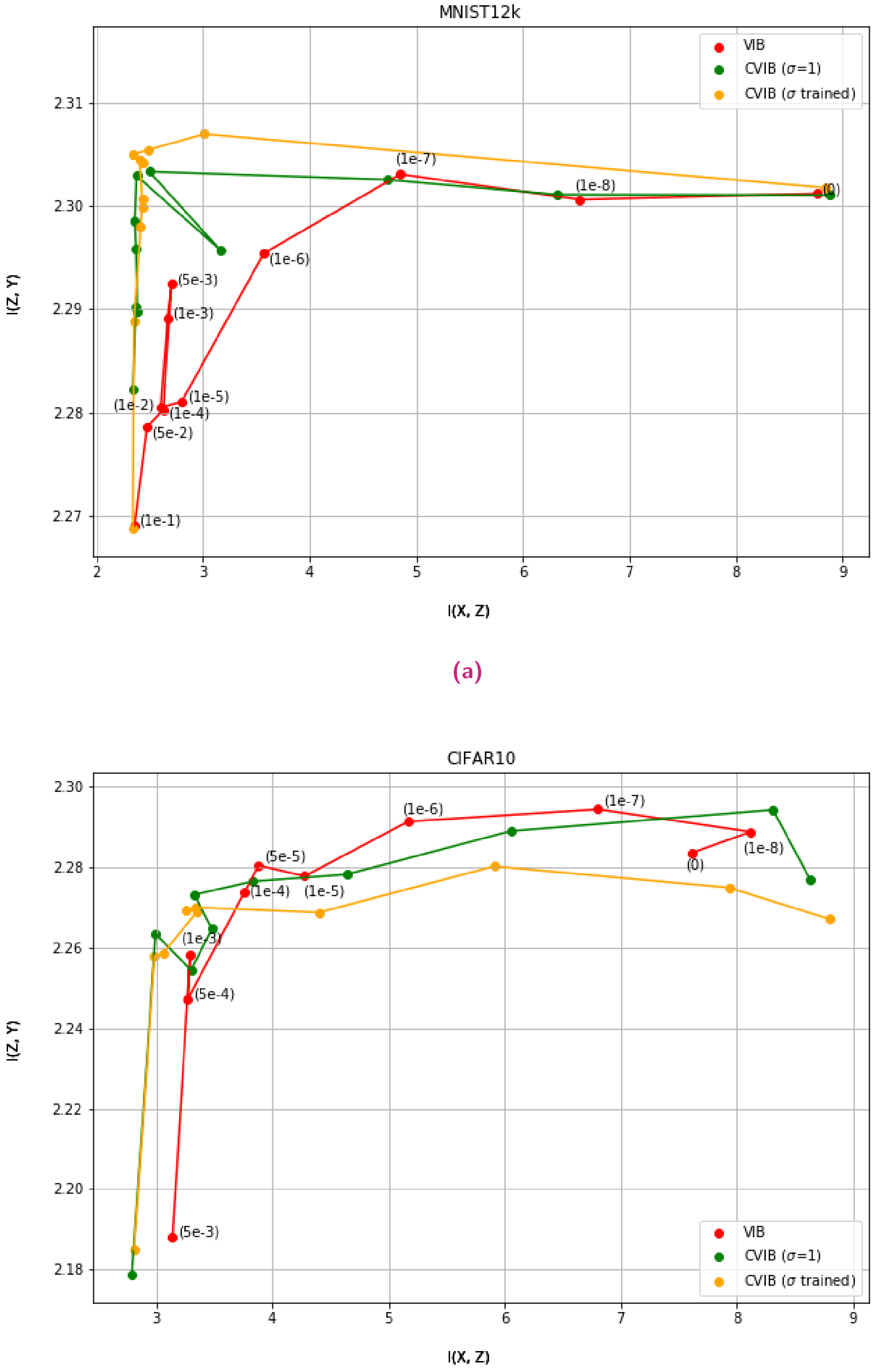
4.3.3 Study on using more complex variational priors and direct Superfluous information minimization

In this section, we finally came to the point of discussion of the effect on introducing more complex variational priors in the Variational Information Bottleneck formulation and switching our focus from minimizing information about the input conveyed in representations $I(X;Z)$ to minimizing superfluous information $I(X;Z|Y)$. We build our comparison on the study of two formulations of our method proposed in Section 2.5. Our focus has been made on the effects of learning, apart from the posterior $p(Z|X)$, different amount of parameters for statistics of $q(Z|Y)$. Namely, in the first case we only learn the mean modeled by $f_d^{\mu}(Y)$ for the approximation of the true conditional prior $p(Z|Y)$, while keeping the variance of approximation fixed and equal to 1 for every dimension. In the second case, we train the $f_d^{\sigma}(Y)$ for the variance of approximation as well. We refer to these cases as CVIB ($\sigma = 1$) and CVIB (σ trained) respectively. We expect that introducing additional flexibility to variational approximations of the true conditional prior distribution should improve the ability of the model both to compress and generalize.

Figure 4.8 shows the comparison of behavior on the information plane for the VIB model and different variations of the CVIB model.[6] As can be seen from the plot, CVIB in both cases shows itself competitively well. For instance, for the MNIST12k case (see Figure 4.8a) the study has shown the CVIB set-up to be dominating over the VIB ones in terms of the level of expressiveness of representations (higher values of estimations for $I(Z;Y)$). Moreover, the position of the nodes on the orange info curve - CVIB (σ trained) case - shows that minimizing superfluous information directly allows us to achieve higher levels of compression for reasonably broader range of β, making the model more invariant to the hyperparameter choice. But, unfortunately, the phenomenon of absolute superiority is not observed for the CIFAR10 case.

On the other hand, estimations for $I(X;Z|Y)$, present in the Figure 4.9, show that for both tasks using higher values of β improved the ability of the model to get rid of the useless attributes present in the raw data. Furthermore, empirical results show that usage of the proposed by us CVIB model tends to convey less amount of superfluous information when compared to the VIB model for each fixed hyperparameter setting. This result is reasonable since the former model focuses on the minimization of this quantity directly.

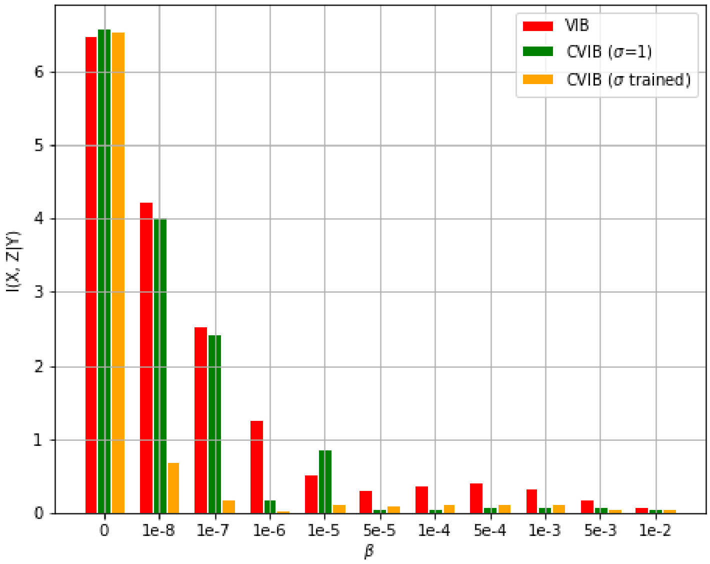
[6] For the sake of readability of plots annotations for CVIB curves are omitted, but the reader can infer β values by putting correspondence with the order of nodes in VIB and CVIB curves, as the range of hyperparameters studied are completely same for both models.



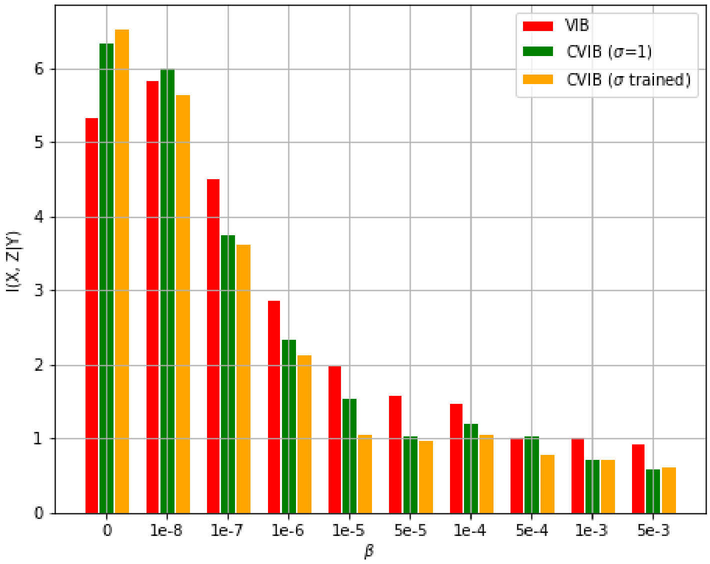
(a)

(b)

Fig. 4.8: Comparison of information curves for the **VIB** model and different versions of the **CVIB** model

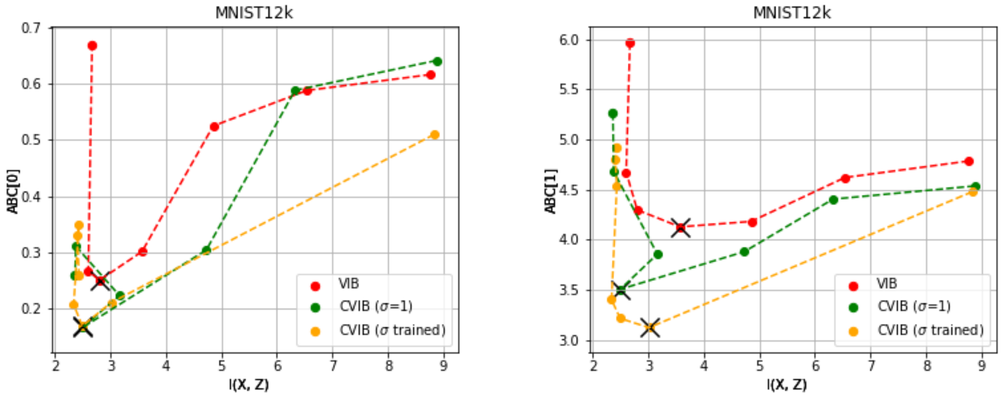


(a) MNIST12k



(b) CIFAR10

Fig. 4.9: Estimations for superfluous information preserved in representations obtained through the **VIB** and the **CVIB** models



(a) Max achievable performance gap

(b) Max possible performance gap

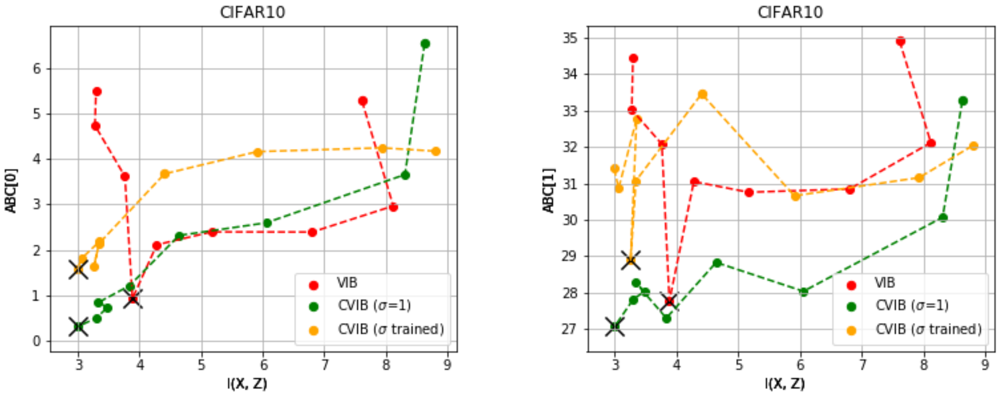
Fig. 4.10: Generalization vs compression comparison via visualization of ABC loss values for $I(X;Z)$ for encoders trained using **VIB** and different variations of **CVIB** on MNIST12k (black crosses denote points corresponding to the best generalization)

ABC on MNIST12k			
β	VIB	CVIB ($\sigma = 1$)	CVIB (σ trained)
0	(0.617, 4.784)	(0.642, 4.535)	(0.510, 4.478)
1e-8	(0.588, 4.619)	(0.589, 4.403)	(0.210, 3.122)
1e-7	(0.525, 4.181)	(0.304, 3.880)	**(0.168, 3.215)**
1e-6	(0.301, 4.127)	**(0.165, 3.504)**	(0.206, 3.401)
1e-5	**(0.249, 4.297)**	(0.222, 3.855)	(0.349, 4.534)
1e-4	(0.266, 4.674)	(0.310, 4.680)	(0.329, 4.796)
1e-3	(0.671, 5.971)	(0.259, 5.268)	(0.260, 4.914)
5e-3	(4.339, 9.955)	(0.253, 5.696)	(0.254, 5.274)
1e-2	(2.902, 8.247)	(0.329, 5.677)	(0.282, 5.385)
5e-2	(14.227, 19.889)	(0.387, 6.167)	(0.308, 6.348)
1e-1	(17.310, 23.573)	(0.289, 6.649)	(0.366, 7.131)

Tab. 4.5: Results for generalization capability via ABC loss for encoders trained using **VIB** and different variations of **CVIB** on MNIST12k

This result gives us the intuition that Results present in the Figure 4.9 show that by increasing the value of trade-off hyperparameter β can indeed help minimizing superfluous information.

The study for the generalization capability (see Tables 4.5 and 4.6) showed that the incorporation of the CVIB principle leads to substantial improvement in generalization when compared to the state-of-the-art VIB model. However, it is hard to say precisely what effect does the flexibility of the variance (i.e should we or should not involve learning the variance of $q(Z|Y)$) has on the overall ability to improve the quality of quantization. Nevertheless, the CVIB (σ trained) model succeeded in showing numerical improvement in the values of ABC for the MNIST12k case when compared to both the VIB and the CVIB (σ=1) cases. Whatever it is, according to visualizations for the generalization/compression trade-off, the VIB model was



(a) Max achievable performance gap

(b) Max possible performance gap

Fig. 4.11: Generalization vs compression comparison via visualization of ABC loss values for $I(X;Z)$ for encoders traines using **VIB** and different variations of **CVIB** on CIFAR10 (black crosses denote points corresponding to the best generalization)

ABC on CIFAR10			
β	VIB	CVIB ($\sigma = 1$)	CVIB (σ trained)
0	(5.273, 34.913)	(6.534, 33.268)	(4.169, 32.036)
1e-8	(2.964, 32.121)	(3.652, 30.071)	(4.241, 31.161)
1e-7	(2.390, 30.850)	(2.592, 28.040)	(4.157, 30.662)
1e-6	(2.393, 30.760)	(2.315, 28.833)	(3.669, 33.461)
1e-5	(2.110, 31.052)	(1.208, 27.304)	(2.139, 31.075)
5e-5	**(0.929, 27.781)**	(0.840, 28.291)	(1.623, 28.903)
1e-4	(3.614, 32.080)	(0.737, 28.034)	(2.189, 32.766)
5e-4	(4.733, 33.022)	(0.503, 27.826)	(1.813, 30.868)
1e-3	(5.495, 34.439)	**(0.309, 27.069)**	**(1.572, 31.416)**
5e-3	(6.978, 34.531)	(0.513, 30.277)	(2.104, 31.403)
5.875e-3	(0, 100)	(0, 100)	(2.388, 31.998)

Tab. 4.6: Results for generalization capability via ABC loss for encoders trained using **VIB** and different variations of **CVIB** on CIFAR10

outperformed by the CVIB method. Although, the superiority of the case with learned variance is only proved for the MNIST12k case, for both tasks CVIB ($\sigma = 1$) could show results both for compression and generalization, better than the VIB did.

Remember that one of the advantages both the VIB and the CVIB models is that they can be used to solve tasks of representation learning without the need to directly compute the amount of preserved information, we can easily overcome difficulties and issues outlined in 3.1. Because of the variational nature of both methods, it is possible to infer the quality of the approximation via the tightness of bounds on the value of the mutual or superfluous information we aim to optimize.

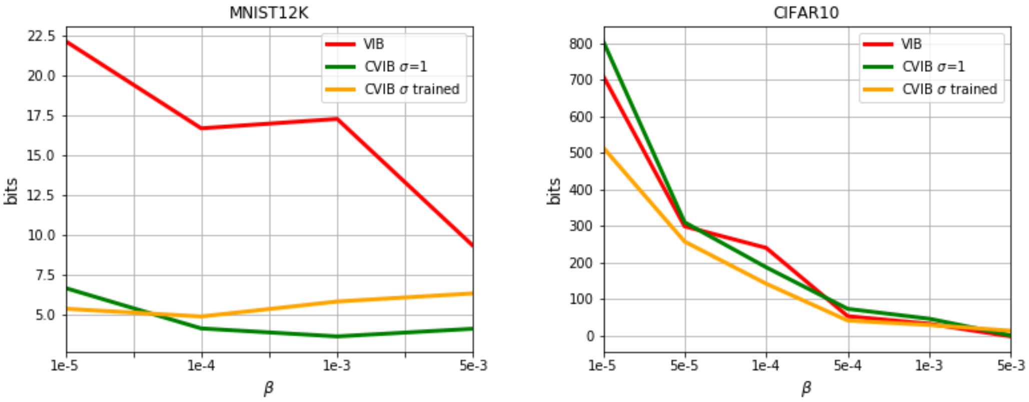
Thus, we can take advantage of another criterion for comparison between these two variations of the Information Bottleneck.[7] In Figure 4.12 we plot the brought together estimations for the tightness of both methods. For both cases we compute this gap as a difference between the variational upper-bound with the correspondent value of the optimization objective. Generally, having this gap to be minimal testifies the quality of the variational approximation approached by the model. Thereby, we can see that the CVIB model shows higher power to bound the superfluous information than the VIB to bound the mutual information with input. In particular, for solving the MNIST12k task (Figure 4.12a) our method proves itself to stick at the margin of up to 10 bits, while its opponent, VIB, is experiencing the gradual tightness increase for the value of β. On the other hand, for CIFAR10 (Figure 4.12b) the margin is not fixed, although for some hyperparameters choice the CVIB method produces tighter gaps. Given this and the intuition that having tighter bounds leads to more accurate estimates on the target value for optimization and, hence, more efficient training, we can hypothesize that the CVIB method is more reliable when opposed to the VIB in terms of the quality of the variational approximation built.

4.4 Overall discussion

Considering the hypothesis and research questions outlined in Section 1.3 and given the empirical and theoretical results presented in Section 4.3, we drive to certain insights about the Information Bottleneck theory and deep learning in general.

The very first insight is that introducing stochasticity to the neural networks leads to a substantial increase in the complexity of representations. We hypothesize that such a result could be treated as a shred of evidence for the fact that building

[7] These methods bound on different quantities making the direct comparison for loss values invalid. However, this doesn't restrict us on evaluating the tightness as the difference between the bound and the estimation for the value of the interest ($I(X;Z)$ or $I(X;Z|Y)$ respectively)



(a) MNIST12k

(b) CIFAR10

Fig. 4.12: Visualization for tightness of the bound on the true value of the mutual information for the **VIB** and the **CVIB** models

probability distribution of features is generally more expressive than choosing the representations to be obtained as embeddings of the data.

The study of this phenomenon led us to our second opening: standard regularization techniques of weight decay and dropout could not compress the data without worsening the generalization ability of the model. The evidence showed that for these methods, unfortunately, compression is accompanied by the drop of generalization performance.

Fortunately, such behavior was not observed for the variations of the Information Bottleneck. All results showed that by following this principle we actually could improve the generalization of the representation learning algorithm while achieving a desirable compression rate. Moreover, given the evidence we obtained we can say that discarding the attributes of the data from representations via mutual or superfluous information minimization led to significantly better results than other regularization methods studied. To us, it means that the VIB and the CVIB models could show themselves to work as decent regularizers by themselves.

Furthermore, we showed that discarding task-irrelevant features directly using the CVIB model could improve on various aspects of learning representations. First, by throwing away knowingly useless features we ensure higher levels of compression while retaining the expressiveness of the features conveyed in representations, as well as overall ability to generalize. Second, by doing so, the overall training procedure becomes more stable and invariant to the hyperparameters choice. For us, the information collapse happened at higher levels of the reconstruction-regularization trade-off. Third, the proposed training procedure showed to use tighter and more effective bounds showing higher effectiveness of the target objective chosen.

Thus, the outcomes of this project succeeded in successfully validating the direction of the research for the incorporation of the Information Bottleneck principle to build stronger feature extractors in the context of solving tasks of deep learning. We believe that further investigation on this topic, as well as building more solid bridges between the information theory and deep learning, will take the humanity one step closer to the development of Artificial General Intelligence and making the world a better place to be.

For standard regularization techniques compression is achieved by means of losing an ability to generalize - in the plots path for the curves is accompanied by the drop of generalization performance.

5 Conclusion

> *Pass on what you have learned.*
>
> — **Master Yoda**

In this work we did an extensive study for validity of this direction of the research via the thorough investigation of the Information Bottleneck principle. Results present in this work proved the relevance of discarding irrelevant information from the input to improve the robustness of features learned from the data. We created an unified framework that can be reused by the average user to quantitatively reason about the quality of the information-based methods, as well as any other representation learning algorithm. Moreover, ideas and concepts explored led us to the derivation of a novel method for representation learning that showed to be predominant to the previous state-of-the-art. Apart from this, we managed to address the relationship establishment for compression and generalization abilities of deep neural networks with the nature of its architecture, as well as, main concerns and issues associated with the incorporation of information-theoretic objectives to solving machine learning tasks.

5.1 Future Work

> *Impossible to see, the future is.*
>
> — **Master Yoda**

> *To infinity and beyond.*
>
> — **Buzz Lightyear**

Although we tried to be as thorough as possible in our investigation, there are still some points that could and should be addressed to improve the persuasiveness of reasoning regarding the validity of the Information Bottleneck, as well as the overall efficiency of the methods chosen to address this question.

First, we believe that a more precise discovery of the underlying variance of the procedures used in this work has to be done. Namely, we believe that our results might not be 100% reliable due to some level of implicit variance present. Of course, we did everything we could to improve the significance of results. However, a couple of steps could have been made to ensure our rightness. We are aware that the reliability of the linear evaluation metric of ABC could be improved if we measure the variance of the evaluations under the fixed initialization set-up. Specifically, one could study the variance of the curves for generalization by not only running an evaluation for a high enough number of seeds but also by redoing so given a fixed set of weight initializations to measure the possible variance of the optimization procedure stand-alone.

Another worth-investing-time direction of the research could be the discovery of the effect of switching finite sample distributions for training the encoder model and evaluation of its ability to compress via mutual information estimation. In theory, there should be no visible effect of having two models for optimization and evaluation being trained on different sets of data points (e.g. training vs held-out sets), as both sets theoretically are to be drawn from the same marginal. This could be helpful for gaining a better understanding of the problem of overfitting finite sample distributions and limitations of mutual information estimation in general.

Also, to improve the arguments for the validity of the Information Bottleneck principle, the investigation of other regularization techniques, such as confidence penalty and label smoothing, could have been done.

On top of that, we did not manage to establish the direct relationship between the flexibility of variation conditional priors, since our results of the study on learning the variance were not unambiguous. We hypothesize that a more precise investigation of the problem of overfitting for these two variants of the proposed method of CVIB should be done. The most obvious step to be done next would be to study the linear separability of compressed spaces using density-based dimensionality reduction algorithms (for instance, UMAP (McInnes et al., 2018)).

Moreover, our validation procedure (framework) can be applied both in supervised and unsupervised problem settings, opening the door to a whole new bunch of methods for deep representation learning (e.g. (Hjelm et al., 2018, Oord et al., 2018, Federici et al., 2020, Löwe et al., 2019). Also, it would be certainly useful to establish a connection for the proposed generalization/compression evaluation metric with other metrics to assess the quality of the model (robustness towards adversarial attacks, etc.).

Bibliography

Achille, Alessandro and Stefano Soatto (2018). “Information Dropout: Learning Optimal Representations Through Noisy Computation”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 40.12, pp. 2897–2905 (cit. on p. 3).

Alemi, Alexander A., Ian Fischer, Joshua V. Dillon, and Kevin Murphy (2016). *Deep Variational Information Bottleneck* (cit. on pp. 3, 7, 12, 13, 24, 33).

Belghazi, Mohamed Ishmael, Aristide Baratin, Sai Rajeshwar, et al. (2018). “Mutual Information Neural Estimation”. In: ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. Stockholmsmässan, Stockholm Sweden: PMLR, pp. 531–540 (cit. on pp. 4, 18).

Bengio, Yoshua, Aaron Courville, and Pascal Vincent (2012). *Representation Learning: A Review and New Perspectives*. cite arxiv:1206.5538 (cit. on p. 1).

Bishop, Christopher M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag (cit. on pp. 3, 14).

Cheng, Hao, Dongze Lian, Shenghua Gao, and Yanlin Geng (2018). “Evaluating Capability of Deep Neural Networks for Image Classification via Information Plane”. In: *European Conference on Computer Vision (ECCV)* (cit. on p. 4).

Cover, Thomas M. and Jay A. Thomas (1991). *Elements of Information Theory*. Wiley (cit. on p. 8).

Dinh, Laurent, Jascha Sohl-Dickstein, and Samy Bengio (2016). “Density estimation using Real NVP”. In: cite arxiv:1605.08803Comment: 10 pages of main content, 3 pages of bibliography, 18 pages of appendix. Accepted at ICLR 2017 (cit. on p. 14).

Federici, Marco, Anjan Dutta, Patrick Forré, Nate Kushman, and Zeynep Akata (2020). “Learning and generalization with the information bottleneck”. In: *ICLR* (cit. on pp. 15, 44).

Fischer, Ian (2019). “The Conditional Entropy Bottleneck”. In: (cit. on p. 16).

Hinton, G., L. Deng, D. Yu, et al. (2012). “Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups”. In: *IEEE Signal Processing Magazine* 29.6, pp. 82–97 (cit. on pp. 1, 10).

Hjelm, R Devon, Alex Fedorov, Samuel Lavoie-Marchildon, et al. (Aug. 2018). *Learning deep representations by mutual information estimation and maximization*. arXiv: `1808.06670` (cit. on pp. 4, 18, 44).

Kingma, Diederik P. and Jimmy Ba (2014). *Adam: A Method for Stochastic Optimization*. cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015 (cit. on p. 25).

Kingma, Diederik P. and Max Welling (2014). "Auto-Encoding Variational Bayes." In: *ICLR*. Ed. by Yoshua Bengio and Yann LeCun (cit. on pp. 3, 13).

Kolesnikov, A., X. Zhai, and L. Beyer (2019). "Revisiting Self-Supervised Visual Representation Learning". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1920–1929 (cit. on pp. 4, 20).

Krizhevsky, Alex (2009). *Learning multiple layers of features from tiny images*. Tech. rep. (cit. on p. 23).

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton (2012). "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 25*. Ed. by F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger. Curran Associates, Inc., pp. 1097–1105 (cit. on p. 1).

Krogh, Anders and John A. Hertz (1992). "A Simple Weight Decay Can Improve Generalization". In: *Advances in Neural Information Processing Systems 4*. Ed. by John E. Moody, Steve J. Hanson, and Richard P. Lippmann. San Francisco, CA: Morgan Kaufmann, pp. 950–957 (cit. on pp. 2, 24).

LeCun, Yann, Corinna Cortes, and CJ Burges (2010). "MNIST handwritten digit database". In: *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist* 2 (cit. on p. 23).

Löwe, Sindy, Peter O'Connor, and Bastiaan S. Veeling (2019). "Putting An End to End-to-End: Gradient-Isolated Learning of Representations." In: *NeurIPS*. Ed. by Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, et al., pp. 3033–3045 (cit. on pp. 4, 18, 44).

McAllester, David and Karl Stratos (2020). "Formal Limitations on the Measurement of Mutual Information". In: ed. by Silvia Chiappa and Roberto Calandra. Vol. 108. Proceedings of Machine Learning Research. Online: PMLR, pp. 875–884 (cit. on pp. 4, 19).

McInnes, Leland, John Healy, and James Melville (2018). *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*. cite arxiv:1802.03426Comment: Reference implementation available at http://github.com/lmcinnes/umap (cit. on p. 44).

Molchanov, Dmitry, Arsenii Ashukha, and Dmitry Vetrov (2017). "Variational Dropout Sparsifies Deep Neural Networks". In: ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. International Convention Centre, Sydney, Australia: PMLR, pp. 2498–2507 (cit. on p. 3).

Müller, Rafael, Simon Kornblith, and Geoffrey E Hinton (2019). "When does label smoothing help?" In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, et al. Curran Associates, Inc., pp. 4694–4703 (cit. on p. 3).

Neal, Radford M. (1995). "Bayesian Learning for Neural Networks". AAINN02676. PhD thesis. CAN (cit. on p. 3).

Nguyen, XuanLong, Martin J Wainwright, and Michael I. Jordan (2008). "Estimating divergence functionals and the likelihood ratio by penalized convex risk minimization". In: *Advances in Neural Information Processing Systems 20*. Ed. by J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis. Curran Associates, Inc., pp. 1089–1096 (cit. on p. 18).

Nowozin, Sebastian, Botond Cseke, and Ryota Tomioka (2016). “f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization”. In: *Advances in Neural Information Processing Systems 29*. Ed. by D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett. Curran Associates, Inc., pp. 271–279 (cit. on p. 18).

Oord, Aäron van den, Yazhe Li, and Oriol Vinyals (2018). “Representation Learning with Contrastive Predictive Coding”. In: *CoRR* abs/1807.03748. arXiv: 1807.03748 (cit. on pp. 4, 44).

Paninski, L (2003). “Estimation of entropy and mutual information”. In: *Neural Computation* 15, pp. 1191–1253 (cit. on p. 17).

Paszke, Adam, Sam Gross, Soumith Chintala, et al. (2017). “Automatic differentiation in PyTorch”. In: (cit. on p. 25).

Pereyra, Gabriel, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey E. Hinton (2017). “Regularizing Neural Networks by Penalizing Confident Output Distributions”. In: *CoRR* abs/1701.06548. arXiv: 1701.06548 (cit. on p. 3).

Poole, Ben, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker (2019). “On Variational Bounds of Mutual Information”. In: ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. Long Beach, California, USA: PMLR, pp. 5171–5180 (cit. on pp. 4, 18).

Rezende, Danilo and Shakir Mohamed (2015). “Variational Inference with Normalizing Flows”. In: ed. by Francis Bach and David Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, pp. 1530–1538 (cit. on p. 14).

Rodríguez Gálvez B.; Thobaben, R.; Skoglund M. (2020). “The Convex Information Bottleneck Lagrangian”. In: *Entropy* 22.98 (cit. on p. 11).

Saxe, Andrew M., Yamini Bansal, Joel Dapello, et al. (2018). “On the Information Bottleneck Theory of Deep Learning.” In: *ICLR (Poster)*. OpenReview.net (cit. on pp. 1, 4, 24).

Sharim, Ohad, Sivan Sabato, and Naftali Tishby (2010). “Learning and generalization with the information bottleneck”. In: *Theoretical Computer Science* 411.2696-2711 (cit. on p. 11).

Shwartz-Ziv, Ravid and Naftali Tishby (Mar. 2017). “Opening the Black Box of Deep Neural Networks via Information”. In: arXiv: 1703.00810 [cs.LG] (cit. on pp. 1, 4).

Singh, Saurabh, Derek Hoiem, and David Forsyth (2016). “Swapout: Learning an ensemble of deep architectures”. In: *Advances in Neural Information Processing Systems 29*. Ed. by D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett. Curran Associates, Inc., pp. 28–36 (cit. on p. 3).

Srivastava, Nitish, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov (2014). “Dropout: a simple way to prevent neural networks from overfitting.” In: *Journal of Machine Learning Research* 15.1, pp. 1929–1958 (cit. on pp. 3, 24).

Tishby, Naftali, Fernando C. Pereira, and William Bialek (1999). “The Information Bottleneck method”. In: *The 37th annual Allerton Conference on Communication, Control, and Computing*, pp. 368–377 (cit. on pp. 2, 3, 9, 10).

Tishby, Naftali and Noga Zaslavsky (2015). “Deep learning and the information bottleneck principle”. In: *2015 IEEE Information Theory Workshop (ITW)*, pp. 1–5 (cit. on pp. 1, 3, 11).

Tschannen, M., J. Djolonga, P. K. Rubenstein, S. Gelly, and M. Lucic (Apr. 2020). “On Mutual Information Maximization for Representation Learning”. In: *8th International Conference on Learning Representations (ICLR)* (cit. on p. 4).

Vera, M., P. Piantanida, and L. R. Vega (2018). “The Role of the Information Bottleneck in Representation Learning”. In: *2018 IEEE International Symposium on Information Theory (ISIT)*, pp. 1580–1584 (cit. on p. 11).

Yeung, R.W. (2008). *Elements of Information Theory*. Springer, pp. 211–228 (cit. on pp. 3, 10).

Zolna, Konrad, Devansh Arpit, Dendi Suhubdy, and Yoshua Bengio (2018). “Fraternal Dropout”. In: *International Conference on Learning Representations* (cit. on p. 3).

List of Figures

List of Tables