SHORTCOMINGS AND NEW PERSPECTIVES ON MUTUAL INFORMATION FOR REPRESENTATION LEARNING

Lukas Gosch
Scientific Internship Report 10/2019-04/2020
gosch.lukas@gmail.com

ABSTRACT

Approaching self-supervised learning through maximizing the mutual information (MI) of a representation with its input data or a transformation thereof has shown great empirical success. However, the employed MI estimators are still poorly understood and uncharacterised for high MI regimes possible in this context. Additionally, learned representations show a critical dependence on the choice of the estimating model family and it has been questioned if the success of these approaches can truly be attributed to MI maximization alone. In this work, we study commonly employed MI estimators in high MI regimes and find that they either diverge or saturate at a threshold connected to the logarithm of the sample size. Additionally, we provide a novel example in which MI fails to explain the behaviour of learned representations. Finally, using the concept of predictive $\mathcal{V}$-information, we make the bias in choosing a MI estimating model family explicit. We use this to purposely control properties in the representations through selecting appropriate estimating model families and show that it easily combines with existing self-supervised methods. This not only overcomes the hardness of estimating mutual information but also showcases the potential of alternative information measures for representation learning.

CONTENTS

1 INTRODUCTION

Features learned in a supervised fashion with deep convolutional neural networks (Lecun et al. (1998), CNNs) on large labeled image datasets such as ImageNet (Russakovsky et al. (2015)) have proven to transfer well to several other image classification datasets and vision tasks (Huh et al. (2016)). However, large amounts of labeled data can be expensive or even impossible to obtain. Therefore, recent research has focused on devising ways to learn useful representations from unlabeled data. One particularly promising approach is called self-supervised learning (SSL) in which auxiliary learning tasks - called *pretext* tasks - are introduced, which are independent of any provided labeling of the data. Self-supervised learning has been shown to achieve impressive results on different vision tasks such as image classification (Kolesnikov et al. (2019), Chen et al. (2020)) or object detection (Goyal et al. (2019)).

Recently, self-supervised learning methods based on the concept of mutual information (MI) maximization have shown promising empirical results (van den Oord et al. (2018), Tian et al. (2019), Hjelm et al. (2018), Bachman et al. (2019)). They are usually based on the *InfoMax principle* (Linsker (1988)), or a multi-view formulation thereof, where the mutual information $I(e(X); X)$ between a learned representation $e(X)$[1] and the corresponding input X is maximized. These self-supervised learning methods promise to be more general than domain specific pretext tasks such as spatial context prediction (Doersch et al. (2015)) or predicting image rotation (Gidaris et al. (2018)).

However, maximizing MI in the context of representation learning requires MI estimation between high dimensional random variables which is challenging (Paninski (2003), McAllester & Stratos (2018)). Therefore, current representation learning methods leverage advancements in neural MI estimation techniques such as MINE (Belghazi et al. (2018)), NWJ (Poole et al. (2019)) or InfoNCE (van den Oord et al. (2018)) which are based on maximizing variational lower bounds to the mutual information (Donsker & Varadhan (1975), Barber & Agakov (2003), Nguyen et al. (2010), Nowozin et al. (2016)).

The behaviour of these estimation methods has mainly been characterised in low mutual information regimes up to ten nats (Poole et al. (2019), Song & Ermon (2019)) and they suffer from serious statistical drawbacks if the true mutual information exceeds certain bounds (McAllester & Stratos (2018)). Additionally, Tschannen et al. (2020) provided empirical evidence that optimizing for tighter bounds on the mutual information can result in worse representations, calling doubt on whether the success of current SSL methods can be explained by mutual information alone.

Furthermore, Tschannen et al. (2020) argue that MI may not be the best measure of information for representation learning and call for a development of alternative measures of information. One such notion could be predictive $\mathcal{V}$*-information* (Xu et al. (2020)). Through including computational constraints of an observer, it promises to capture observed phenomena such as the creation of "usable" information through hierarchical computations as exhibited in deep neural networks while being estimable.

The contribution of this report is threefold:

- We show that in high mutual information regimes, as can be expected to be encountered in visual representation learning, MINE diverges and the related NWJ estimator shows a saturation phenomenon connected to the logarithm of the batch size. This could in part explain the unstable training behaviour reported when optimizing using these estimators in an InfoMax setting (van den Oord et al. (2018), Hjelm et al. (2018)). Furthermore, training a log density ratio estimator through Jensen-Shannon divergence maximization behaves inconsistent for direct plug-in mutual information estimation and depending on the negative sampling strategy, it does not yield improved properties for the NWJ-bound based estimator.
- We information theoretically argue, if the data is not augmented, there cannot be a worse pretraining task than the one whose introduced auxiliary label distribution $p(y')$ has no mutual information with the label distribution $p(y)$ of a downstream task of interest. Then, we

[1] With capital letters (e.g. X) we refer to random variables where with lower-case letters (e.g. x) we denote their realizations. We understand a representation g as a function which maps from the input space of X to an arbitrary output space. As a function of a random variable, g(X) is again a random variable.

empirically show that there are pretraining tasks which succeed even though $I(Y';Y) = 0$. Additionally to Tschannen et al. (2020)'s arguments that high $I(e(X);X)$ is not necessarily predictive of the downstream performance of a learned representation, this questions whether the concept of mutual information is too encompassing or too coarse to capture the reasons behind successful representation learning.

- In reformulating the InfoMax principle using $\mathcal{V}$-information (Xu et al. (2020)), which we call *$\mathcal{V}$-InfoMax*, we make the bias in choosing a function family for mutual information estimation explicit. We then show that the observer measuring information in a representation and the models used in the evaluation protocol have to be chosen dependently namely corresponding to similar function classes. Based on this insight, we show that the $\mathcal{V}$-InfoMax principle can be combined with domain specific pretext tasks resulting in an asymmetric supervised autoencoder architecture (Le et al. (2018)) which can boost the generalization capabilities of the learned representations.

2 Background

2.1 Mutual Information

Mutual information (MI) $I(X,Y)$ between two random variables X[2] and Y is defined as the expected value of the log-ratio of their joint density $p(x,y)$ with the product of their marginal densities $p(x)p(y)$:

$$I(X;Y) := \mathbb{E}_{p(x,y)}\left[\frac{p(x,y)}{p(x)p(y)}\right] = D_{KL}(p(x,y)||p(x)p(y)) \tag{1}$$

Similarly, the conditional mutual information $I(X;Y|Z)$ is defined by

$$I(X;Y|Z) := \mathbb{E}_{p(x,y,z)}\left[\frac{p(x,y|z)}{p(x|z)p(y|z)}\right] = D_{KL}(p(x,y|z)||p(x|z)p(y|z)) \tag{2}$$

As MI can be seen as the Kullback-Leibler (KL) divergence between these two distributions, it is non-negative. Furthermore, it is invariant to reparametrization, i.e. $I(X;Y) = I(g(X),h(Y))$ as long as g and h are invertible maps. Lastly, the *data processing inequality* states no processing of X can increase its information about Y, i.e. $I(g(X);Y) \leq I(X;Y)$. Due to the fact that MI can be rewritten as $I(X;Y) = H(Y) - H(Y|X)$, it can be understood as the reduction of uncertainty in Y when knowing X (and vice versa) and as such is characterizing the shared information between both random variables (Cover & Thomas (2006)). For a naive application in the representation learning context, this means that the processing of the input data X through a hierarchy of layers in a deep neural network can only decrease its overall information carried about a label distribution Y. While true, this clearly fails to describe the reasons and effectiveness of deep representation learning.

2.2 Generalization of Mutual Information to Multiple Random Variables

Multivariate mutual information and interaction information are two generalizations of the concept of mutual information to arbitrary numbers of random variables. In the case of three variables, they coincide and can be defined using conditional mutual information (McGill (1954), Hu (1962)):

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

The definition holds for any ordering of X, Y and Z. Multivariate MI can be hard to interpret as it can be negative (see Appendix A). However, if the three random variables form a Markov chain $X \to Y \to Z$ then $I(X;Z|Y) = 0$ and $I(X;Y;Z) = I(X;Z) \geq 0$.

[2]If not stated otherwise, we consider the input data X to be a discrete random variable. Then, a representation $g(X)$ is a function with a discrete input domain and therefore again a discrete random variable.

2.3 MUTUAL INFORMATION ESTIMATION

Neural mutual information estimators are based on maximizing different variational lower bounds to the mutual information $I(X;Y)$ (Poole et al. (2019)). Most notable for representation learning are *MINE* (Belghazi et al. (2018)), *NWJ* (Poole et al. (2019)) and *InfoNCE* (van den Oord et al. (2018)).

MINE is based on a lower bound (4) to the Kullback-Leibler (KL) divergence formulation of MI derived by Donsker & Varadhan (1975)

$$I(X;Y) \geq \mathbb{E}_{p(x,y)} T_\omega(x,y) - \log(\mathbb{E}_{p(x)p(y)} e^{T_\omega(x,y)}) \tag{4}$$

It is a lower bound for any choice of function T. In practice, T is usually parametrized by a small neural network. Hence, we parametrize it by ω and denote it T_ω. Then, to maximize (4), it has to learn to distinguish samples from the joint distribution $p(x,y)$ and the product of the marginal distributions $p(x)p(y)$. Therefore, T_ω is usually called a *critic*. 4 can be shown to be tight for a certain class of optimal functions and for any choice of critic bigger or equal to the NWJ lower bound (Belghazi et al. (2018)). However, due to the expectation in the logarithm, using Monte Carlo estimation results in biased evaluation and biased gradient estimates. Therefore, Belghazi et al. (2018) proposed to replace the estimate of $\mathbb{E}_{p(x)p(y)} e^{T_\omega(x,y)}$ in the occurring denominator of the gradient of (4) by an exponential moving average.

NWJ is similarly a lower bound to the KL divergence form of MI derived by Nguyen, Wainwright and Jordan (Nguyen et al. (2010)):

$$I(X;Y) \geq \mathbb{E}_{p(x,y)} T_\omega(x,y) - \mathbb{E}_{p(x)p(y)} e^{T_\omega(x,y)-1} \tag{5}$$

Contrary to MINE it allows to take unbiased gradients. Equation (5) is tight for $T^* = \log \frac{p(y|x)}{p(y)} + 1$. Therefore, Poole et al. (2019) argues to first train a log density ratio estimator and plug it into the NWJ bound to improve stability and variance of the estimator compared to directly optimizing (5).

A log density ratio is usually obtained through following the f-GAN formulation in Nowozin et al. (2016) to train a discriminator D_ω between $p(x,y)$ and $p(x)p(y)$ using the objective

$$\mathbb{E}_{p(x,y)} \log D_\omega(x,y) + \mathbb{E}_{p(x)p(y)} \log(1 - D_\omega(x,y)) \tag{6}$$

where $D_\omega(x,y) = \sigma(T_\omega(x,y)) = \frac{1}{1+e^{-T_\omega(x,y)}}$. For maximizing (6), the optimal solution is $T^* = \log \frac{p(y|x)}{p(y)}$ (see Appendix B.1). Additionally, this log density ratio estimate can be used for direct plug-in MI estimation (Poole et al. (2016), Mescheder et al. (2017), Achille & Soatto (2017)) but then, loses its lower bound guarantee.

Objective (6) is also a lower bound to the Jensen-Shannon (JS) divergence between $p(x,y)$ and $p(x)p(y)$ (Goodfellow et al. (2014)) and can alternatively be written as

$$\mathbb{E}_{p(x,y)} - \zeta(-T_\omega(x,y)) - \mathbb{E}_{p(x)p(y)} \zeta(T_\omega(x,y)) \tag{7}$$

with $\zeta(T_\omega(x,y)) = \log(1 + e^{T_\omega(x,y)})$ being the softplus function (see Appendix B.2). Hjelm et al. (2018) uses this reformulation as an alternative objective to maximize for representation learning with the InfoMax principle (see Section 2.4).

InfoNCE (van den Oord et al. (2018)) is defined by the lower bound (8)

$$I(X;Y) \geq \mathbb{E} \frac{1}{N} \sum_{i=1}^{N} \log \frac{e^{T_\omega(x_i,y_i)}}{\frac{1}{N}\sum_{j=1}^{N} e^{T_\omega(x_i,y_j)}} \tag{8}$$

where the expectation is taken over N independent samples from the joint distribution $p(x,y)$. It can theoretically be shown that the InfoNCE estimator saturates at $\log(N)$ making it impracticable for

high MI regimes. Therefore, in characterizing the behaviour of neural MI estimators for high underlying MI, we focus on MINE and NWJ (see Section 3). Interestingly, training representations using the InfoNCE estimator shows particular empirical success. However, Tschannen et al. (2020) have linked training representations through employing the InfoNCE objective (8) with metric learning and questioned if viewing it through the lense of MI maximization is the correct conceptual model. Indeed, recent work on contrastive learning Chen et al. (2020) shows impressive results without motivating their ideas through MI maximization.

2.4 INFOMAX PRINCIPLE

Given a family of encoder functions $\mathcal{E} = \{e_\theta : \mathcal{X} \to \mathcal{Z} | \theta \in \Theta\}$ (e.g. CNNs) mapping from the input space $\mathcal{X}$ to a space of representations $\mathcal{Z}$, the InfoMax principle (Linsker (1988)) reads

$$\max_\theta I(e_\theta(X); X) \tag{9}$$

Through maximizing the mutual information, it tries to find a set of parameters θ which maximizes the information stored in the learned representations about the inputs.

A related principle states to maximize the MI between two separate representations $e_1(X_1)$, $e_2(X_2)$ obtained from two different views X_1 and X_2 of the input data X:

$$\max_{e_1 \in \mathcal{E}_1, e_2 \in \mathcal{E}_2} I(e_1(X_1); e_2(X_2)) \tag{10}$$

Through choosing $X_2 = X$ and $e_2 = id(\cdot)$ (10) reduces to (9). Tschannen et al. (2020) showed that (10) can also be understood as a lower bound to $I(e_1(X_1), e_2(X_2); X)$. Equation (10) is the basis for MI based contrastive learning approaches (van den Oord et al. (2018), Tian et al. (2019)). However, the basic idea behind contrastive learning can be formulated to maximize the agreement between $e_1(X_1)$ and $e_2(X_2)$ without necessarily framing it as MI maximization (Becker & Hinton (1992), Chen et al. (2020)). On the other hand, Hjelm et al. (2018) combines (9) with (10) to learn a global representation which maximizes MI with the input data as well as local representations of it and call their approach *Deep InfoMax*.

It has been criticised if the success of representations learned through optimizing for equations (9) and (10) can actually be attributed to MI maximization alone. Especially, there seems to be an intricate interplay between the model families used as critics in MI estimation and the properties of the resulting representations Tschannen et al. (2020) (see Section 5.1). For investigating the general mechanisms behind these observations, we focus on the most simple case of the classic InfoMax principle given through equation (9)

2.5 EVALUATING REPRESENTATIONS

It is unclear how to best evaluate representations learned unsupervised. Therefore, Zhang et al. (2016) introduced a linear evaluation protocol which has been widely adopted in the self-supervised literature (van den Oord et al. (2018), Kolesnikov et al. (2019), Goyal et al. (2019), Tschannen et al. (2020), Chen et al. (2020)). It states to freeze the network and extract representations from various layers through down pooling. Then, a linear classifier is trained upon the extracted features to solve a downstream task such as image classification if the representations have been obtained through training on image datasets such as CIFAR-10 (Krizhevsky (2012)) or ImageNet (Goyal et al. (2019)).

Experiments are performed using ResNet and PlainNet[3] architectures (He et al. (2015)) as skip connection have a significant impact on the learned representations (Kolesnikov et al. (2019)). We follow the linear evaluation protocol and extract features after each convolution block of the ResNet denoted $B1$, $B2$, $B3$ and $B4$ as well as after the first convolutional layer denoted $B0$. As a PlainNet has an identical architecture to a ResNet just missing its skip connections, the convolutional blocks

[3]A PlainNet has exactly the same architecture as a ResNet but without skip-connections. Therefore, it corresponds to a plain convolutional network giving it its name.

are defined identically. Additionally, the $prelogits$ obtained through global average pooling are used as features. $B0 - B4$ are obtained through max-pooling to the same dimensionality as the $prelogits$.

Additionally, we measure the normalized mutual information $NMI(e(X); Y)$ between the above representations $e(X)$ and downstream labeling Y as defined in equation (11).

$$NMI(e(X); Y) = \frac{I(X; Y)}{H(Y)} \tag{11}$$

Therefore, $NMI(e(X); Y)$ can be interpreted as what fraction of information in Y can be predicted through knowing X. In contrast to employing a linear classifier, $NMI(e(X); Y)$ corresponds to an upper bound on the extractable information contained in the representations about another random variable Y.

$NMI(e(X); Y)$ is estimated using a log-ratio density estimator obtained through maximizing for the Jensen-Shannon divergence (see Section 2.3) and using it as a plug-in estimator for $I(X; Y)$. For a comparison of the different estimators from section 2.3 for evaluating the quality of a representation see appendix D.1.

2.6 $\mathcal{V}$-INFORMATION

Predictive $\mathcal{V}$-information is a new notion of information introduced in Xu et al. (2020) incorporating computational and modelling constraints. Therefore, it is a notion of *useable* information. Let $\mathcal{X}$ be the sample space of X and $\mathcal{P}(\mathcal{Y})$ denote the set of all possible probability distributions over the sample space of Y. Then a *predictive family* is defined as a set of functions $\mathcal{V} \subseteq \Omega = \{f : \mathcal{X} \cup \{\varnothing\} \to \mathcal{P}(\mathcal{Y})\}$ that satisfies optional ignorance[4]. In this definition, X can be interpreted as side information for predicting Y and $\varnothing$ as having no side information. Then, *predictive conditional $\mathcal{V}$-entropy* is defined as

$$H_{\mathcal{V}}(Y|X) = \inf_{f \in \mathcal{V}} \mathbb{E}_{x,y \sim X,Y} \left[-\log f[x](y)\right]$$
$$H_{\mathcal{V}}(Y|\varnothing) = \inf_{f \in \mathcal{V}} \mathbb{E}_{y \sim Y} \left[-\log f[\varnothing](y)\right]$$

where $H_{\mathcal{V}}(Y|X)$ can be interpreted as an approximation to $H(Y|X)$ as we cannot represent the true $p(y|x)$ due to computational constraints limiting us to models in $\mathcal{V}$.

Predictive $\mathcal{V}$-information is then defined as

$$I_{\mathcal{V}}(X \to Y) := H_{\mathcal{V}}(Y|\varnothing) - H_{\mathcal{V}}(Y|X)$$

which can be interpreted as the information in X about Y as measurable with $\mathcal{V}$.

$\mathcal{V}$-information owns properties which make it an interesting information measure, among others (for more on these we refer the reader to Xu et al. (2020)):

- For $\mathcal{V} = \Omega$ reduces to $I(X; Y)$ [5]
- It can be created through computation being consistent with preprocessing of input data through forwarding at through a hierarchy of layers.
- For certain choices of $\mathcal{V}$, PAC learning bounds (Valiant (1984)) can be derived for $\mathcal{V}$-information estimation. This is in stark contrast to the hardness or even impossibility of MI estimation in high MI settings (McAllester & Stratos (2018)).

[4] $\forall f \in \mathcal{V}, \forall P \in range(f), \exists f' \in \mathcal{V}, s.t. \forall x \in \mathcal{X}, f'[x] = P, f'[\varnothing] = P$

[5] Indeed, Xu et al. (2020) show that depending on the choice of $\mathcal{V}$ many common notions of uncertainty can be obtained as a special case of $\mathcal{V}$-information.

3 MI ESTIMATION IN HIGH MI-REGIONS

The neural mutual information estimators employed in self-supervised representation learning have only been systematically studied in mutual information regions up to ten nats (Poole et al. (2019), Song & Ermon (2019)). However, in the image domain, the mutual information between a representation and its original data could easily exceed this threshold (Alemi et al. (2016), van den Oord et al. (2018))[6]. As the InfoNCE estimator is theoretically bounded at $\log(N)$ where N is the batch size used for evaluation van den Oord et al. (2018), it becomes impractical to apply this estimator for higher MI regions. Therefore, this section characterizes the high MI behaviour of MINE (see Section 3.1) and NWJ (see Section 3.2). We find that both estimators are well behaved for mutual information regions up to $\log(N)$ but that above this threshold, MINE shows a divergence and NWJ a saturation phenomenon.

For all experiments, we used a joint critic architecture Belghazi et al. (2018) in which the inputs x, y are concatenated and fed in as one big input vector to the critic network $T(x, y)$. This approach has been found by Poole et al. (2019) to consistently outperform a separate critic architecture $T(x, y) = h(x)^T g(y)$ as proposed in van den Oord et al. (2018). Furthermore, we apply the negative sampling scheme proposed by Belghazi et al. (2018) to first sample a batch $(x, y) \sim p(x, y)$ and then shuffle y yielding $(x, \tilde{y})$. This results in equally many positive and negative samples and requires only $2N$ forward passes through the network. Therefore, this approach significantly speeds up computation time compared to using all $N(N-1)$ possible negative (x_i, y_j) pairs as done in Poole et al. (2019) and Song & Ermon (2019). We observe that using the simpler sampling scheme proposed by Belghazi et al. (2018) results in decreased variance and increased stability of the estimators (see Appendix D.2). Further details about the experiment architectures and training procedures can be found in Appendix C.1.

3.1 MINE

To evaluate the behaviour of MINE we use the commonly used test dataset of two 20-dimensional Gaussian and choose their correlation to match different true underlying MI's (Belghazi et al. (2018), Poole et al. (2019), Song & Ermon (2019)).

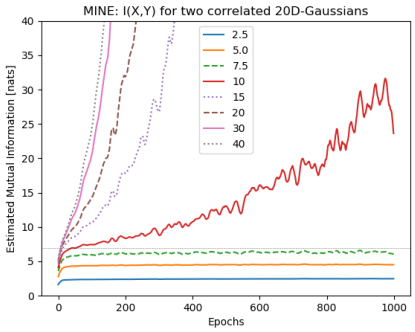
In figure 1a, one can observe that MINE behaves stable up to certain true MI threshold above which, it diverges. The divergence occurs consistently across different critic architectures, non-linearities and regularization schemes. Interestingly, it seems to be connected to the logarithm of the batch size $\log(N)$ which corresponds to the grey dotted line in figure 1a. If the batch size is chosen so that $\log(N)$ is below the estimated MI values for a true MI of 7.5 nats, MINE diverges (see Figure 1b). On the other hand, estimating a true MI of 10 nats can be stabilized by choosing a large enough batch size (see Figure 1c). Additionally, these figures show that the speed of divergence is dependent on i) the true underlying MI and ii) the used batch size.

Figure 1d shows the effects of using a very small critic network as done by Belghazi et al. (2018). Then, the speed of divergence is significantly slowed leading to the third dependency on the critic capacity. Therefore, if the training of a smaller critic network is stopped too early its divergence can be easily overlooked. Using the same critic network and learning scheme as done by Belghazi et al. (2018), we could show that their reported results for two correlated 20D-Gaussians are an artefact of stopping the training too early.

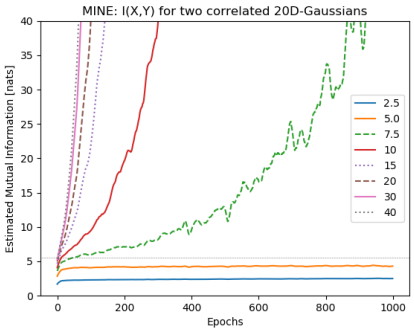
Therefore, too high underlying MI for given choice of training scheme could explain the unstable training behaviour of MINE reported by several authors when applying MINE in an InfoMax setting (van den Oord et al. (2018), Hjelm et al. (2018)).

Additionally, this casts doubt on whether MINE can sensibly be used to evaluate the quality of a representation through calculating $I(e(X); X)$ as done by Hjelm et al. (2018) who reports MI estimates up to a hundred nats.

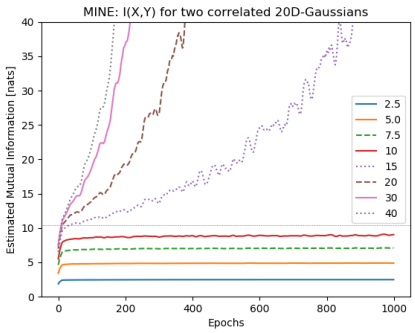
[6]Also consider that $I(e(X); X) \leq H(X) \leq 3 \cdot d \cdot \log_e(256)$ [nats] for RGB images where $e(X)$ is a representation of X and d the number of input pixels (e.g. $256 \cdot 256$ for ImageNet). Due to the structure in images it is natural to assume that this bound is loose. However, this showcases that higher MI values are easily possible.



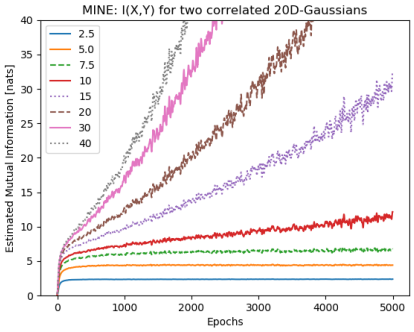
(a) Batch size: 1024, $\log_e(1024) = 6.93$



(b) Batch size: 256, $\log_e(256) = 5.55$



(c) Batch size: 32768, $\log_e(32768) = 10.40$



(d) Using the same configuration as in Belghazi et al. (2018).

Figure 1: MI estimation with MINE between two 20D-Gaussians. Each line corresponds to a different correlation chosen and is labeled based on the resulting underlying MI in nats. One epoch corresponds to 100 independently drawn batch-samples. a-c) using a two layer MLP critic with 512 hidden units in each layer. d) using a small MLP critic with one hidden layer with 64 units and a batch size of 512.

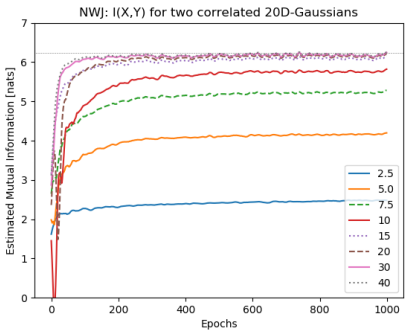
3.2 NWJ

Figure 2a shows the result of applying NWJ to same test dataset of two correlated 20D-Gaussian as MINE. Interestingly, instead of a divergence phenomenon it shows a saturation phenomenon. NWJ behaves well for small MI regions but its bias increases with the underlying true MI and at some point the estimates saturate. Figure 2a shows two effects i) NWJ cannot distinguish MI regimes bigger than 15 nats. ii) NWJ's saturation point coincides with the logarithm of the used batch size N for training (grey dotted line)

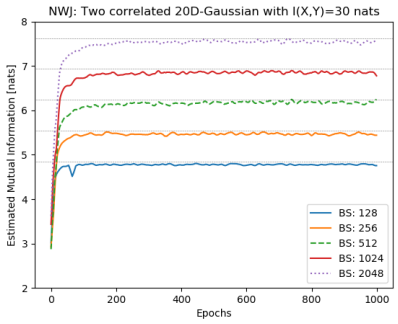
Figure 2b shows that NWJ's saturation at $\log(N)$ did not occur by chance. Fixing the underlying MI at 30 nats and training NWJ with varying batch sizes, the estimate always saturates at the corresponding $\log(N)$ (grey dotted lines).

Additionally, we consistently observe across a large number of models (see Appendix C.1) that NWJ with the simple negative sampling procedure proposed in Belghazi et al. (2018) enjoys stable training[7] and significantly less variance than for other negative sampling procedures (see Appendix D.2). This makes the alternative two step training scheme to first train a density estimator through maximizing the Jensen-Shannon divergence and then plugging it into the NWJ bound to improve its variance and stability (Poole et al. (2019)) not necessary.

[7]For very high capacity critics with three layers and 1024 hidden units per layer or more, we sometimes observe divergence for the very low underlying MI of 2.5 nats which may be due to numerical issues.



(a) Batch size: 512, $\log_e(512) = 6.24$



(b) Holding the underlying MI constant but varying the batch size N and comparing each to $\log(N)$ (grey dotted lines).

Figure 2: MI estimation with NWJ between two 20D-Gaussians. Each line corresponds to a different correlation chosen and is labeled based on the resulting underlying MI in nats. One epoch corresponds to 100 independently drawn batch-samples. Estimates obtained by training a two layer MLP critic with 512 hidden units in each layer. a) Shows that NWJ's bias increases with the underlying MI up to the saturation point $\log(N)$ drawn with a grey dotted line. b) By fixing the correlation to result in a MI of 30 nats and varying the batch size N for training, shows that the saturation point indeed is defined by $\log(N)$.

3.2.1 JS

Figure 3 shows the results of the alternative two step training scheme where first a density estimator is trained through maximizing the Jensen-Shannon divergence and then plugging it into the NWJ bound as well as used for direct plug-in MI estimation (see also Mescheder et al. (2017)).

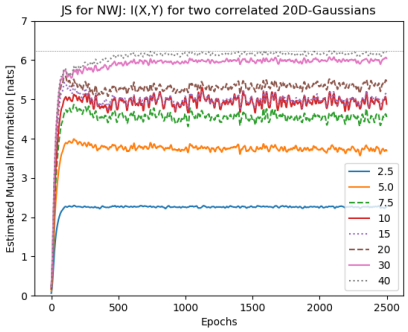
Figure 3a shows that plugging the density estimator into the NWJ bound could yield odd training behaviour where in the first few epochs the estimates for $I(X;Y)$ increase and then decrease. We find that this behaviour worsens if the critic becomes more complex (see Figure 13 in Appendix D.3) and can be controlled through regularization (see Figure 3c). It can be again observed that using the density estimator for evaluating the bound results in a saturation at $\log(N)$ and that introducing regularization increases the bias.

Figure 3b shows the behaviour of using the density estimator as a direct plug-in estimator for the MI. It can be observed that the estimator works well for MI regions up to 10 nats. However, for higher underlying MI's its produces lower and lower estimates. Therefore, direct plug-in MI estimation with JS maximization loses the crucial property of monotonicity for estimation MI in high MI regions. Without regularization, we find that this behaviour to consistently happen across critic architectures. Introducing regularization (see Figure 3d) this problem does not occur. However, it now shows a spiking behaviour for high underlying MI's early in training and loses its discriminative power w.r.t. higher MI's. This odd training behaviour at high MI's questions whether the estimates for direct plug-in estimation can be trusted at these MI ranges.

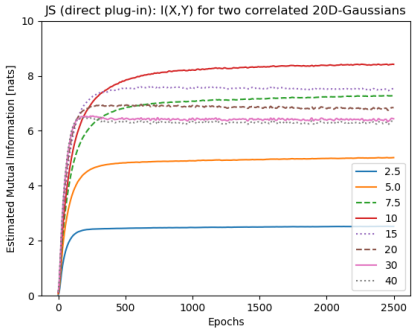
Using the JS density estimator in the NWJ lower bound, we observe increased variance compared to directly optimizing for NWJ. Therefore, using the negative sampling strategy introduced in Belghazi et al. (2018), using JS instead of directly NWJ does not yield advantages but even introduces the problem of finding a good regularization scheme. This picture reverses if the negative sampling procedure introduced by Poole et al. (2019) is used (see Appendix D.2).

3.3 Further Notes and Implications

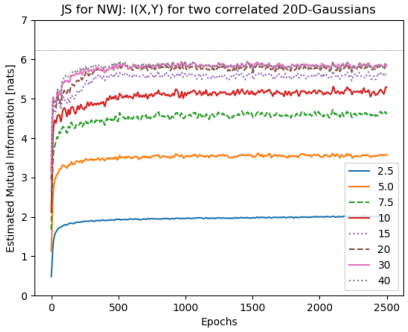
That MINE diverges and NWJ saturates at an underlying MI related to $\log(N)$ draws in interesting connection to work done by McAllester & Stratos (2018) who proved that no high confidence lower bound to the mutual information exists, which is higher than $\log(N)$. Investigating this theoretical connection is a highly interesting direction for future work. How different negative sampling



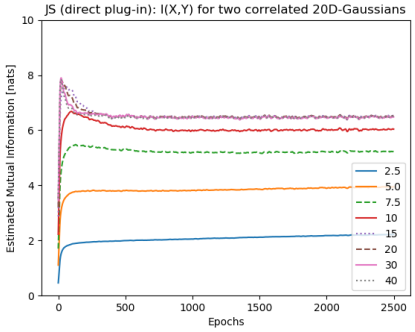
(a) Critic: 1-Layer MLP with 64 hidden units



(b) Critic: 1-Layer MLP with 64 hidden units



(c) Critic: 3-Layer MLP with 512 hidden units, dropout and weight decay



(d) Critic: 3-Layer MLP with 512 hidden units, dropout and weight decay

Figure 3: MI estimation through Jensen-Shannon divergence maximization between two 20D-Gaussians. *JS for NWJ* denotes using the density estimator to plug into the NWJ lower bound and *JS (direct plug-in)* using the density estimator as a direct plug-in estimator for $I(X;Y)$. a) and b) show the results of an unregularized critic and c) and d) of a strongly regularized critic with dropout probabilities $p_1 = 0.5, p_2 = 0.3, p_3 = 0.2$ and weight decay of $2.5 \cdot 10^{-5}$. For training, a batch size of 512 is used - the grey dotted line corresponds to $\log(512)$.

schemes effect the behaviour of these estimators and especially NWJ's trade-off between bias and variance is another interesting and connected research direction.

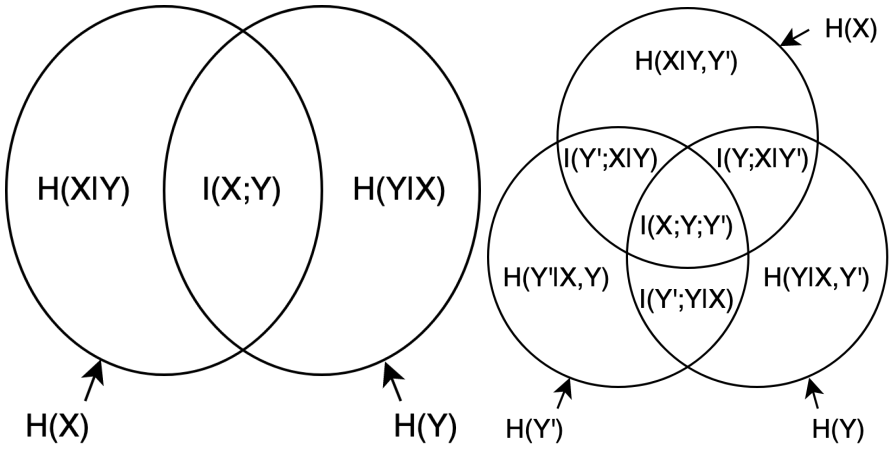
For all figures in this section, the critics employed ELU non-linearities. We observe exactly the same qualitative behaviour for ReLU non-linearities except that for some test instances choosing ReLUs can result in slightly higher variance and slightly faster divergence than ELUs.

4 MI Between a Training & Test Label Distribution

Given a downstream label distribution $p(y|x)$ and the goal to learn useful features extracted from an input X for predicting Y through training on an auxiliary label distribution $p(y'|x)$ in a self-supervised fashion. In this section we give an information theoretic argument that the usefulness of $p(y'|x)$ for self-supervised learning can be judged by the mutual information $I(Y;Y')$ between the marginal label distributions $p(y)$ and $p(y')$. Then we show that the concept of mutual information cannot explain the observed downstream behaviour of self-supervised learned representations. This is achieved through training on two different pretraining task on CIFAR-10 which both fulfil $I(Y';Y) = 0$ but one yields generalizing[8] representations while the other does not.

[8] As measured with the linear evaluation protocol and mutual information with the downstream label distribution.

Consider a family of encoder functions $\mathcal{E} = \{e : \mathcal{X} \to \mathcal{Z}\}$ and a family of classifiers $\mathcal{G} = \{g : \mathcal{Z} \to \mathcal{Y}\}$ which take a representation $e(x) \in \mathcal{Z}$ with $e \in \mathcal{E}$ and $x \in \mathcal{X}$ as input and output a prediction in some label space $\mathcal{Y}$. That a classifier $g \in \mathcal{G}$ can generalize in predicting Y given $e(X)$, the encoder e has to make information in X which is predictive about Y accessible to g. The predictive information in X about Y corresponds to $I(X;Y)$ as visualized in the Venn diagram in figure 4a. In self-supervised learning, $\mathcal{E}$ could be a class of deep CNNs and $\mathcal{G}$ corresponding to different logistic regression models applied on top of the deep representations.



(a) Relationship between entropy and mutual information. $I(X;Y)$ corresponds to the intersection of information in X with the information in Y (see Cover & Thomas (2006)).

(b) Information theoretic measures for the three random variables X, Y and Y'. For the definitions of conditional MI and the multivariate MI, see Section 2.1 and 2.2 respectively.

Figure 4: Relationships between the information content between two and three random variables X, Y and Y'. We identify X with the input data and Y and Y' as two different labelings of X. An exposition of a set-theory interpretation of entropy can be found in Reza (1961).

Consider now an auxiliary labeling Y' of X following the conditional distribution $p(y'|x)$ and that $e \in \mathcal{E}$ and $g \in \mathcal{G}$ have been chosen so as to maximize predictability of Y' given X. Then, e has to make shared information between X and Y', corresponding to a subset of $I(X;Y')$, accessible to the model family $\mathcal{G}$.

Now, identify Y with a downstream labeling we want to be predictive about. Then, the mutual information $I(X;Y')$ can be decomposed into (see figure 4b)

$$I(X;Y') = I(Y';X|Y) + I(X;Y;Y')$$

Information extracted by e and predictive about Y' but not Y lies in $I(Y';X|Y)$ whereas information predictive about both Y' and Y is captured by the multivariate mutual information $I(X;Y;Y')$.

For general random variables $I(X;Y;Y')$ can be negative (see Appendix A), but assuming that the two labelings Y and Y' only depend on the data, X, Y and Y' form a Markov chain $Y \leftrightarrow X \leftrightarrow Y'$. Then,

$$\begin{aligned} I(X;Y;Y') &= I(Y;Y') - I(Y;Y'|X) \\ &= I(Y;Y') \geq 0 \end{aligned}$$

as $I(Y;Y'|X) = 0$ due to Markovity.

Therefore, choosing Y' to have high $I(X;Y;Y') = I(Y;Y')$ should increase the information extracted by e which is predictive for Y[9] and therefore aid downstream performance if it is measured with a similar function family to $\mathcal{G}$.

[9]This of course only holds true if the encoder function family $\mathcal{E}$ can make use of the information increase.

Additionally, if $I(Y;Y')=0$ the encoder e optimally chosen for Y' has no incentive to extract any information in $I(X;Y)$. In this sense, choosing a Y' fulfilling $I(Y;Y')=0$ corresponds to choosing a maximally unhelpful auxiliary labeling.

We now construct two different auxiliary labelings fulfilling $I(Y;Y')=0$ with respect to a downstream labeling Y and show that one labeling results in representations with good downstream performance whereas the other does not. Let the input data corresponds to CIFAR-10 enlarged by adding for each image a $90°$, $180°$ and $270°$ rotated version and Y corresponds to the ten CIFAR-10 object categories. As a first auxiliary labeling Y_1' the rotations of the images are chosen (similar to what $Gidariset\ al.$ (2018) proposed as a pretext task). If $i \in \{0°, 90°, 180°, 270°\}$ and y is a CIFAR-10 image label, then $p(Y_1'=i|Y=y)=p(Y_1'=i)=\frac{1}{4}$[10] resulting in $I(Y;Y_1')=0$. As a second auxiliary labeling Y_2' consider random labels with the same support as Y but chosen uniformly. As Y_2' is again per construction independent of Y, $I(Y,Y_2')=0$.

Table 1 shows downstream performance results of training a ResNet18 on this dataset without data augmentation (not altering the distribution of X for training or evaluation) as well as for randomly initializing the weights of the network as a baseline. It clearly highlights that even though $I(Y;Y_2')=I(Y;Y_1')=0$, predicting rotations clearly outperforms random labels and the baseline by large margins whereas training on random labels is destructive. The same qualitative behaviour holds for a plain convolutional network without skip connections (see Appendix D.4).

This shows that $I(Y;Y')$ is not necessarily predictive of downstream performance and can't explain the downstream behaviour of learned representations. In addition to $Tschannenet\ al.$ (2020), who showed that large mutual information between a representation and the input data is not necessarily predictive of downstream performance, this highlights limitations of capturing the success of self-supervised learning through mutual information.

	ResNet18					
$N\hat{M}I(e_\theta(X);Y)$	B0	B1	B2	B3	B4	Prelogits
Rotation Labels Y_2'	0.50	**0.61**	**0.65**	**0.67**	**0.62**	**0.62**
Random Labels Y_1'	**0.52**	0.53	0.44	0.28	0.06	0.11
Random Initialization	0.50	0.47	0.39	0.32	0.27	0.29
Linear Down. Accuracy	B0	B1	B2	B3	B4	Prelogits
Rotation Labels Y_2'	41.65	**52.51**	**55.96**	**57.48**	**56.80**	**51.04**
Random Labels Y_1'	**44.10**	45.91	39.60	31.51	19.98	20.83
Random Initialization	42.12	42.06	39.47	34.91	33.00	32.95

Table 1: Evaluating the representations learned after different convolutional blocks of a ResNet18 trained on rotation or random labels of the enlarged CIFAR-10 dataset. Evaluation done using the normalized mutual information score $N\hat{M}I(e_\theta(X);Y)$ between the learned representations and the downstream labels and the common linear evaluation protocol (see Section 2.5). Even though, $I(Y;Y_2')=I(Y;Y_1')=0$, predicting rotations clearly outperforms random labels and the random initialization baseline by large margins whereas training on random labels even worsens the representation compared to the baseline.

5 On the Marriage of MI-based approaches and $\mathcal{V}$-Information

5.1 $\mathcal{V}$-InfoMax Principle

Recall the InfoMax principle (Equation 9) from section 2.4 which tries to find a set of parameters θ which maximizes the information stored in a learned representation $e_\theta(X)$ about the input X. In practice, optimizing for the mutual information is accomplished through neural mutual information estimation (see Section 2.3) in which a critic network $T_\omega : \mathcal{X} \times \mathcal{Z} \to \mathbb{R}$ of limited capacity, parametrized by ω, is trained to discriminate between the joint distribution $p(x,z)$ and the product of the marginals $p(x)p(z)$. Then, the InfoMax objective optimized for reads (Hjelm et al. (2018))

[10] That is, Y_1' and Y are independent.

$$\max_{\omega,\theta} I_\omega(e_\theta(X); X) \tag{12}$$

However, high mutual information alone does not necessarily state anything about its accessibility with respect to certain function classes. As an example, consider choosing $e_\theta(X) = id(X)$ to be the identity mapping or an invertible encryption scheme which both would preserve the mutual information. Furthermore, Tschannen et al. (2020) showed that representations learned through optimizing for high mutual information and evaluated using the common linear evaluation protocol employed in self-supervised learning (see Section 2.5), are highly sensitive to the critic architecture employed. Especially they showed that employing simple model classes as a bilinear critic $T(x, z) = x^\top W z$ significantly outperforms more complex critics such as a multi-layer perceptron (MLP) in downstream linear accuracy even though yielding looser MI bounds. While this behaviour cannot be explained through the lens of mutual information, it is the expected behaviour of and can be explained through a $\mathcal{V}$-information reformulation of the InfoMax principle given below.

Using $\mathcal{V}$-information to reformulate equation 12, one has to take into account that contrary to MI, $\mathcal{V}$-information is asymmetric. Consider $\mathcal{V}$ to be the class of models used to evaluate the learned representation. Then, the goal of representation learning is to make the learned features $e_\theta(X)$ as useful to the models in $\mathcal{V}$ as possible. Therefore, the natural directionality is such that the predictive $\mathcal{V}$-information $e_\theta(X)$ contains about the input X, measured with respect to the model class $\mathcal{V}$, should be maximized:

$$\max_{\theta} I_{\mathcal{V}}(e_\theta(X) \to X) \tag{13}$$

We call objective (13) the *$\mathcal{V}$-InfoMax principle*. Contrary to the classic InfoMax principle (9), it explicitly takes the model class intended for downstream usage of the representation into account.

For a practical implementation of the $\mathcal{V}$-InfoMax principle, a predictive family $\mathcal{V}$ has to be selected. Following Xu et al. (2020), $\mathcal{V}$ is chosen represent functions mapping from the representation space $\mathcal{Z}$ to the mean of a Gaussian distribution.

$$\mathcal{V} = \{f : z \to \mathcal{N}(\phi_\omega(z), \Sigma), z \in \mathcal{Z}, \varnothing \to \mathcal{N}(\mu, \Sigma) | \mu \in \mathbb{R}^D; \Sigma = \tfrac{1}{2} I_{D\times D}, \phi_\omega \in \Phi\}$$

where $z = e_\theta(x) \in \mathcal{Z}$ is a representation of a given input $x \in \mathbb{R}^D$[11] and $\Phi = \{\phi_\omega : \mathcal{Z} \to \mathbb{R}^D\}$ is a set of decoder functions, parametrized by ω. Now, Φ can be chosen based on the targeted model class to process further the learned representations. Then,

$$\begin{aligned}
\max_\theta I_{\mathcal{V}}(e_\theta(X) \to X) &= \max_\theta \{H_{\mathcal{V}}(X|\varnothing) - H_{\mathcal{V}}(X|e_\theta(X))\} \\
&= \max_\theta \left\{ \inf_{\mu\in\mathbb{R}^D} \mathbb{E}_{p(x)} \left[-\log \frac{1}{(2\pi)^{\frac{D}{2}} |\Sigma|^{\frac{1}{2}}} e^{-||x-\mu||_2^2} \right] \right. \\
&\quad \left. - \inf_{\phi_\omega\in\Phi} \mathbb{E}_{p(x,e_\theta(x))} \left[-\log \frac{1}{(2\pi)^{\frac{D}{2}} |\Sigma|^{\frac{1}{2}}} e^{-||x-\phi_\omega(e_\theta(x))||_2^2} \right] \right\} \\
&= \underbrace{\inf_{\mu\in\mathbb{R}^D} \mathbb{E}_{p(x)} ||x-\mu||_2^2}_{tr(Cov(X))} + \max_\theta \left\{ - \inf_{\phi_\omega\in\Phi} \mathbb{E}_{p(x,e_\theta(x))} ||x - \phi_\omega(e_\theta(x))||_2^2 \right\} \\
&\Leftrightarrow \min_\theta \left\{ \min_\omega \mathbb{E}_{p(x,e_\theta(x))} ||x - \phi_\omega(e_\theta(x))||_2^2 \right\} \\
&= \min_{\theta,\omega} \mathbb{E}_{p(x,e_\theta(x))} ||x - \phi_\omega(e_\theta(x))||_2^2
\end{aligned} \tag{14}$$

Given a sample of data points $\mathcal{D} = \{x_i \in \mathcal{X}\}_{i=1}^N$, based on equation (14), the empirical $\mathcal{V}$-InfoMax principle takes the form of the following optimization problem:

[11] For images, D corresponds to the number of pixel times the number of color channels.

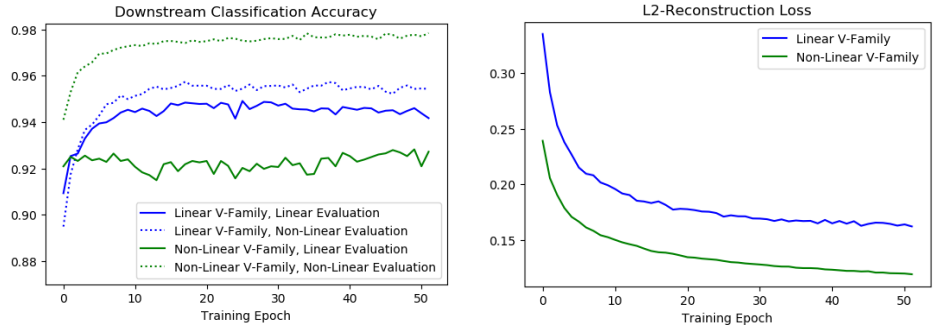
$$\max_{\theta} \hat{I}_{\mathcal{V}}(e_\theta(X) \to X; \mathcal{D}) \Leftrightarrow \min_{\theta,\omega} \frac{1}{N} \sum_{i=1}^{N} ||x_i - \phi_\omega(e_\theta(x_i))||_2^2 \tag{15}$$

Therefore, maximizing the $\mathcal{V}$-information in a representation reduces to training a deterministic autoencoder LeCun (1987) using the $L2$-norm as reconstruction loss. However, as equation (15) is derived from an information theoretic viewpoint to maximize the usable information with respect to a certain downstream model class, it motivates a highly asymmetric autoencoder design. For modern self-supervised learning with the linear evaluation protocol, the encoder e_θ will correspond to a very deep CNN whereas the decoder will be chosen to be a linear function class. In this sense, the corresponding autoencoder shows maximal asymmetry.

5.1.1 Experiments

To verify the dependence of the learned representations on the choice of function family $\mathcal{V}$, we employ the $\mathcal{V}$-InfoMax principle to train representations on MNIST (see Figure 5).

It showcases that indeed when training self-supervised using objective 15, adapting the class of decoder functions Φ to the used downstream evaluation model class is a crucial design choice. Figure 5a shows that representations learned through choosing a linear decoder model significantly outperforms a non-linear decoder model using a linear evaluation protocol and vice versa. Additionally, figure 5b shows that unexpectedly, using a non-linear decoder class yields a lower L2-reconstruction loss and therefore higher predictive $\mathcal{V}$-information. This concludes that higher $\mathcal{V}$-information does not necessarily translate into better downstream performance if the representations are evaluated with the wrong model class.



(a) Validation accuracies of the learned representations using the $\mathcal{V}$-InfoMax principle.

(b) L2-loss reconstruction loss of the decoder on the validation set.

Figure 5: Training a MLP encoder with two hidden layers of 300 units and a linear output layer with 100 units on MNIST using objective 15. The linear evaluation protocol is a logistic regression model achieving 92.35% accuracy in pixel space. The non-linear evaluation protocol is again defined by a MLP with two hidden layers of 300 units each achieving 98.16% accuracy in pixel space. The difference between the $\mathcal{V}$-families and the evaluation models is that the first have a last layer of 784 units whereas the later ones end in 10 output units. (a) Shows that downstream linear accuracy is significantly boosted by choosing a linear decoder family whereas a non-linear decoder family performs worse using a linear evaluation protocol but boosts non-linear downstream performance. (b) Shows that the non-linear $\mathcal{V}$-family achieves higher predictive $\mathcal{V}$-information about X.

Table 2 shows that this insight carries to a more complex setting on CIFAR-10. There, as an encoder a ResNet18 is chosen and trained using objective 15 for 50 epochs. Again, one clearly sees that higher predictive $\mathcal{V}$-information does not directly translate into better downstream performance but that matching the decoder models and the evaluation models to belong to similar classes of functions achieves best downstream results.

These results show that using the $\mathcal{V}$-InfoMax principle, the decoder function family and the models used for evaluation have to be seen as dependent and can be co-designed in a simple manner. In addition to making the bias introduced through the choice of critic architecture explicit, $\mathcal{V}$-information

ResNet18 (Prelogits)	Linear Decoder	Non-Linear Decoder	Random Init
Linear Down. $\mathcal{V}$-information	**0.29**	0.19	0.23
Linear Down. Accuracy	**44.8**	38.24	40.12
Non-Linear Down. $\mathcal{V}$-information	0.44	**0.46**	0.38
Non-Linear Down. Accuracy	56.14	**57.54**	50.9
$L2$ Reconstruction Loss	0.28	**0.14**	-

Table 2: Statistics about the prelogits layer of a ResNet18 trained using objective (15) on CIFAR-10 for 50 epochs. The linear evaluation protocol is a logistic regression model the non-linear decoder family are MLPs with three hidden layers with 512 units each. It clearly highlights that even though a non-linear decoder family achieves lower $L2$ reconstruction loss and in that higher $\hat{I}_{\mathcal{V}}(e_\theta(X) \to X)$, this does not automatically translate in better downstream performance. Choosing a linear decoder family significantly outperforms a non-linear decoder w.r.t. to the linear evaluation protocol whereas the non-linear decoder outperforms the linear decoder w.r.t. a non-linear evaluation protocol. Choosing a non-linear decoder can result in the representation worsening w.r.t. to random initialization (which is known to result in oriented high-frequency filters and therefore can be a sensibly applied for preprocessing Saxe et al. (2011)), if measured with the wrong downstream model class.

enjoys PAC-learnability (Valiant (1984)) for certain choices of $\mathcal{V}$ (Xu et al. (2020), see Section 2.6) - independent of the actual MI. This theoretically justifies the practical applicability of the $\mathcal{V}$-InfoMax principle to high dimensional, high MI settings as encountered in representation learning whereas correct MI estimation becomes questionable (McAllester & Stratos (2018), see Section 3).

5.2 Combination with Domain Specific Pretext Tasks

Due to the generality of the $\mathcal{V}$-InfoMax principle and the predictable dependence of the learned representations on $\mathcal{V}$, it can be easily combined with domain-specific pretraining tasks yielding an asymmetric supervised autoencoder architecture visualized in figure 6.

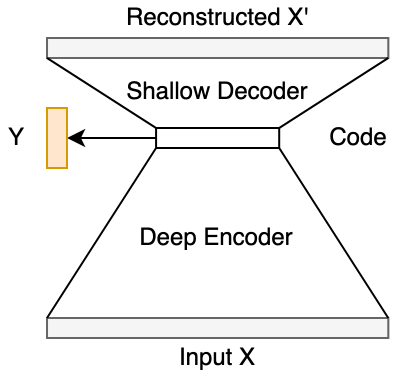


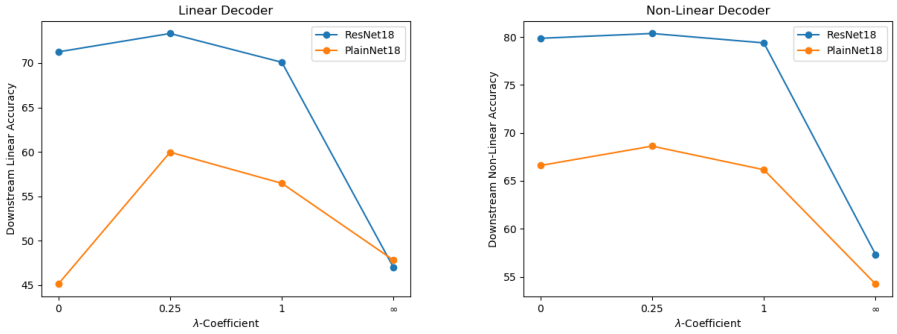
Figure 6: Asymmetric supervised autoencoder for self-supervised learning with a domain specific pretraining task defining the input X and labels Y and $\mathcal{V}$-information maximization through a simple decoder function class.

Assume $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ to be a data sample defined by an arbitrary pretext task with a corresponding loss $\ell : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$. The supervised autoencoder then optimizes for equation (16):

$$\min_{\theta,\omega} \sum_{i=1}^{N} \ell(x_i, y_i) + \lambda \sum_{i=1}^{N} ||x_i - \phi_\omega(e_\theta(x_i))||_2^2 \tag{16}$$

where λ is a hyperparameter defining the weight given to the $\mathcal{V}$-InfoMax objective. One expects a natural trade-off between how specialized the pretraining task is to the downstream problem and the correct size for λ.

Figure 7 shows the results of training CNNs with and without skip-connections on CIFAR-10 using the common pretraining task of predicting rotations (Gidaris et al. (2018)) and combining it with the $\mathcal{V}$-InfoMax objective for different choices of λ. Figure 7a shows that by choosing a linear $\mathcal{V}$-family and a right λ, the downstream linear accuracy of the learned representations can be significantly increased compared to only training for the pretext task ($\lambda = 0$). Figure 7b shows that non-linear downstream accuracy is pushed by choosing a non-linear decoder model, though less pronounced. This is due to models in figure 7 are trained using hyperparameter settings optimized for training on the rotation task only. Table 3 details the results for choosing $\lambda = 0.25$ and showcases that the used learning scheme results in suboptimal training performance of the non-linear decoder achieving a worse $L2$-reconstruction loss than using a linear decoder model. Therefore, it can be expected that the non-linear downstream accuracy performance of the non-linear decoder can be significantly increased by appropriately adapting the learning scheme. Furthermore, the effect of increased downstream linear accuracy through a linear decoder is more pronounced in the PlainNet-architecture. As CNNs without skip-connections are known to increasingly specialize to the pretext task compared to CNNs with skip-connections Kolesnikov et al. (2019), this shows that adding the $\mathcal{V}$-InfoMax objective reduces pretext specialization of a representation.



(a) Downstream Linear Accuracies when choosing a Linear $\mathcal{V}$-Family

(b) Downstream Non-Linear Accuracies when choosing a Non-Linear $\mathcal{V}$-Family

Figure 7: Training a supervised autoencoder (SAE, equation (16)) with ResNet18 or PlainNet18 as encoders on CIFAR-10 with different choices for λ. Either using the rotation pretext task only ($\lambda = 0$) or using a linear or non-linear decoder model with $\lambda = 0.25$ and $\lambda = 1.0$ (same model families as in table 2). $\lambda = \infty$ refers to only training for the $\mathcal{V}$-InfoMax objective.

	ResNet18 (Prelogits)			**PlainNet18** (Prelogits)		
Decoder	$\lambda = 0$	Linear	Non-Linear	$\lambda = 0$	Linear	Non-Linear
Linear Down. Accuracy	71.25	**73.32**	72.42	45.12	**59.96**	52.68
Non-Linear Down. Accuracy	79.86	79.24	**80.35**	66.6	68.48	**68.62**
$L2$ reconstruction loss	-	**0.32**	0.44	-	**0.34**	0.46
Accuracy on Pretext Task	75.94	**76.26**	76.11	75.55	**75.80**	75.65

Table 3: Details the results achieved by the representations obtained by the models corresponding to $\lambda = 0.25$ in figure 7 and contrasts it with pretraining on the pretext task only.

The best results in figure 7 are achieved by setting $\lambda = 0.25$ and further increasing λ leads to worse representations. This is not surprising as predicting rotations is a very domain specific pretext task especially optimal for image classification. However, training on the less optimal pretraining task of predicting random labels, higher λ values yield better representations (see Table 4). This confirms the trade-off character of λ as having to be chosen based on how specialized the pretext task is to the intended downstream task.

	ResNet18 (Prelogits)				**PlainNet18** (Prelogits)			
Linear Decoder	$\lambda = 0$	$\lambda = 0.25$	$\lambda = 1$	$\lambda = \infty$	$\lambda = 0$	$\lambda = 0.25$	$\lambda = 1$	$\lambda = \infty$
Linear Down. Accuracy	27.38	36.72	45.74	**46.96**	15.26	32.58	44.56	**47.80**
Non-Linear Down. Accuracy	32.14	43.46	53.54	**60.70**	19.82	35.44	50.16	**61.44**
$N\hat{M}I(e_\theta(X); Y)$	0.18	0.33	0.45	**0.49**	0.04	0.25	0.42	**0.50**

Table 4: Training a supervised autoencoder (SAE, equation (16)) with ResNet18 or PlainNet18 as encoders on CIFAR-10. Either using the random pretext task only ($\lambda = 0$) or using a linear decoder model with $\lambda = 0.25$ and $\lambda = 1$. $\lambda = \infty$ refers to only training for the $\mathcal{V}$-InfoMax objective. It confirms that if the pretext task is not optimal for a downstream task, increasing the weight for the $\mathcal{V}$-InfoMax objective aids generalization and prevents overspecialization.

6 Conclusion

In this work, we studied the commonly employed neural MI estimators MINE and NWJ in high MI regimes possibly encountered in visual representation learning. MINE diverges and NWJ saturates at a threshold related to $\log(N)$ where N is the sample size used for evaluating the bounds. This finding adds to the question if the success behind MI maximization based self-supervised learning methods can actually be attributed to MI maximization alone if the true underlying MI exceeds $\log(N)$. Investigating a connection to work done by McAllester & Stratos (2018) who proved that no high confidence lower bound to MI can exceed $\log(N)$, is a natural direction for future research. How different negative sampling schemes effect the behaviour of these estimators, the $\log(N)$ threshold and especially NWJ's trade-off between bias and variance is another interesting and connected research direction.

More generally, we found that from an information theoretic viewpoint, a representation has no incentive to keep information in the dataset about a downstream labeling Y if it is trained on an auxiliary labeling Y' satisfying $I(Y; Y') = 0$. Then, we showed that the empirical behaviour of learned representations contradicts this information theoretic viewpoint and $I(Y; Y')$ is not necessarily predictive of downstream performance of the representations. As $I(e(X); X)$ is neither (Tschannen et al. (2020)), this highlights limitations of capturing the success of self-supervised learning through the concept of mutual information.

Through reformulating MI maximization using the concept of predictive $\mathcal{V}$-information (Xu et al. (2020)), we introduced the $\mathcal{V}$-InfoMax principle and showed that it makes the bias in choosing a MI estimating model family explicit. We used this principle to purposely control properties in the resulting representations by selecting appropriate estimating model families and showed that it easily combines with existing self-supervised methods. The $\mathcal{V}$-InfoMax principle does not only overcomes the hardness of estimating MI for visual representation learning, even giving PAC-learning bounds for a correct choice of $\mathcal{V}$, but also shows that the MI estimating model families and the models used for evaluating representations have to be seen as dependent and can be co-designed in a simple manner.

This concludes that alternative information measures, and in particular $\mathcal{V}$-information, have the power to overcome limitations of the mutual information concept in self-supervised representation learning and constitute an interesting direction for further research. Additionally, we expect $\mathcal{V}$-information to be easily applicable to other lines of representation learning research involving mutual information in particular the information bottleneck (Tishby & Zaslavsky (2015)).

Acknowledgments

We want to thank Mohamed Ishmael Belghazi for providing his MINE implementation to reproduce the results for two correlated 20D-Gaussians reported in Belghazi et al. (2018). Furthermore, I want to thank Mary Phuong for making me aware of the work done by Xu et al. (2020) on "useable" information and Paul Henderson for pointing out interesting work done on MI estimation. Especially, I want to thank Christoph Lampert for his supervision of this project and continuous support. Last but not least, I want to thank all group members of the Machine Learning and Computer Vision Group at IST Austria for their interesting inputs and discussion on the project and machine learning in general.

REFERENCES

Alessandro Achille and Stefano Soatto. Emergence of invariance and disentanglement in deep representations, 2017.

Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information bottleneck. *CoRR*, abs/1612.00410, 2016. URL http://arxiv.org/abs/1612.00410.

Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views, 2019.

David Barber and Felix V. Agakov. The im algorithm: A variational approach to information maximization. In *NIPS*, 2003.

Suzanna Becker and Geoffrey E. Hinton. Self-organizing neural network that discovers surfaces in random-dot stereograms. *Nature*, 355:161–163, 1992.

Ishmael Belghazi, Sai Rajeswar, Aristide Baratin, R. Devon Hjelm, and Aaron C. Courville. MINE: mutual information neural estimation. *CoRR*, abs/1801.04062, 2018. URL http://arxiv.org/abs/1801.04062.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. *ArXiv*, abs/2002.05709, 2020.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, USA, 2006. ISBN 0471241954.

Aaron Defazio, Francis R. Bach, and Simon Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. *CoRR*, abs/1407.0202, 2014. URL http://arxiv.org/abs/1407.0202.

Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction. *CoRR*, abs/1505.05192, 2015. URL http://arxiv.org/abs/1505.05192.

Monroe D Donsker and SR Srinivasa Varadhan. Asymptotic evaluation of certain markov process expectations for large time, i. *Communications of Pure and Applied Mathematics*, 28(1):1–47, 1975.

A. Elad, D. Haviv, Y. Blau, and T. Michaeli. Direct validation of the information bottleneck principle for deep nets. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pp. 758–762, 2019.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *CoRR*, abs/1803.07728, 2018. URL http://arxiv.org/abs/1803.07728.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

Priya Goyal, Dhruv Mahajan, Harikrishna Mulam, and Ishan Misra. Scaling and benchmarking self-supervised visual representation learning. pp. 6390–6399, 10 2019. doi: 10.1109/ICCV.2019.00649.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL http://arxiv.org/abs/1512.03385.

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization, 2018.

K.T. Hu. On the amount of information. *Theory Probab. Appl.*, 7:439–447, 1962.

Mi-Young Huh, Pulkit Agrawal, and Alexei A. Efros. What makes imagenet good for transfer learning? *CoRR*, abs/1608.08614, 2016. URL http://arxiv.org/abs/1608.08614.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.

Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. Revisiting self-supervised visual representation learning. *International Conference on Computer Vision*, 2019.

Alex Krizhevsky. Learning multiple layers of features from tiny images. *University of Toronto*, 05 2012.

Lei Le, Andrew Patterson, and Martha White. Supervised autoencoders: Improving generalization performance with unsupervised regularizers. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pp. 107–117, Red Hook, NY, USA, 2018. Curran Associates Inc.

Y. LeCun. Modèles connexionistes de l'apprentissage. *Ph.D. thesis, Université de Paris VI. 17, 499, 511*, 1987.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998. ISSN 1558-2256. doi: 10.1109/5.726791.

Ralph Linsker. Self-organization in a perceptual network. *IEEE Computer*, 21:105–117, 03 1988. doi: 10.1109/2.36.

David McAllester and Karl Stratos. Formal limitations on the measurement of mutual information. *CoRR*, abs/1811.04251, 2018. URL http://arxiv.org/abs/1811.04251.

W. McGill. Multivariate information transmission. *Psychometrika*, 19(1):97–116, 1954.

Lars M. Mescheder, Sebastian Nowozin, and Andreas Geiger. Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. *CoRR*, abs/1701.04722, 2017. URL http://arxiv.org/abs/1701.04722.

X. Nguyen, M. J. Wainwright, and M. I. Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11):5847–5861, Nov 2010. ISSN 1557-9654. doi: 10.1109/TIT.2010.2068870.

XuanLong Nguyen, Martin J. Wainwright, and Michael I. Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11):5847–5861, Nov 2010. ISSN 1557-9654. doi: 10.1109/tit.2010.2068870. URL http://dx.doi.org/10.1109/TIT.2010.2068870.

Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization, 2016.

Liam Paninski. Estimation of entropy and mutual information. *Neural Comput.*, 15(6):1191–1253, June 2003. ISSN 0899-7667. doi: 10.1162/089976603321780272. URL https://doi.org/10.1162/089976603321780272.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Ben Poole, Alexander A. Alemi, Jascha Sohl-Dickstein, and Anelia Angelova. Improved generator objectives for gans, 2016.

Ben Poole, Sherjil Ozair, Aäron van den Oord, Alexander A. Alemi, and George Tucker. On variational bounds of mutual information. *CoRR*, abs/1905.06922, 2019. URL http://arxiv.org/abs/1905.06922.

Fazlollah M. Reza. *An Introduction to Information Theory*. Originally published: New York: McGraw-Hill, USA, 1961. ISBN 0-486-68210-2.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Andrew Saxe, Pang Koh, Zhenghao Chen, Maneesh Bhand, Bipin Suresh, and Andrew Ng. On random weights and unsupervised feature learning. pp. 1089–1096, 04 2011.

Jiaming Song and Stefano Ermon. Understanding the limitations of variational mutual information estimators, 2019.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *CoRR*, abs/1906.05849, 2019. URL http://arxiv.org/abs/1906.05849.

Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. *CoRR*, abs/1503.02406, 2015. URL http://arxiv.org/abs/1503.02406.

Michael Tschannen, Josip Djolonga, Paul K. Rubenstein, Sylvain Gelly, and Mario Lucic. On mutual information maximization for representation learning. *ArXiv*, abs/1907.13625, 2020.

L. G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, November 1984. ISSN 0001-0782. doi: 10.1145/1968.1972. URL https://doi.org/10.1145/1968.1972.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018. URL http://arxiv.org/abs/1807.03748.

Yilun Xu, Shengjia Zhao, Jiaming Song, Russell Stewart, and Stefano Ermon. A theory of usable information under computational constraints, 2020.

Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization. *CoRR*, abs/1603.08511, 2016. URL http://arxiv.org/abs/1603.08511.

A Negativity of Multivariate Mutual Information

Write $I(X;Y;Z) = I(X;Z) - I(X;Z|Y)$ and consider an XOR gate defined by table 5 for which X and Z are independent random inputs and Y is the output. As X and Z are independent $I(X,Z) = 0$. Furthermore, given we know Y, if now happen to gain knowledge about Z, we also know the outcome of X, therefore $I(X;Z|Y) > 0$ (1 bit) which results in $I(X;Y;Z) < 0$. □

X	Z	Y= X $\veebar$ Y
0	0	0
0	1	1
1	0	1
1	1	0

Table 5: Behaviour of an XOR gate

B Mutual Information Estimation

B.1 Optimal Solution to the f-GAN Objective is a Log Density Ratio Estimator

Restating the f-GAN objective with a data distribution $p_{data}(x,y) = p(x,y)$ and a fixed generator distribution $p_g(x,y) = p(x)p(y)$, equation (6) reads

$$\mathbb{E}_{p_{data}(x,y)} \log D(x,y) + \mathbb{E}_{p_g(x,y)} \log(1 - D(x,y)) \tag{17}$$

where $D(x,y) = \sigma(T(x,y)) = (1 + e^{-T(x,y)})^{-1}$. Goodfellow et al. (2014) and Nowozin et al. (2016) showed that the optimal solution for maximizing (17) with respect to the discriminator D is

$$D^*(x,y) = \frac{p_{data}(x,y)}{p_{data}(x,y) + p_g(x,y)} = \frac{p(x,y)}{p(x,y) + p(x)p(y)}$$

Therefore,

$$\begin{aligned}
\sigma(T^*(x,y)) &= \frac{p(x,y)}{p(x,y) + p(x)p(y)} \\
\Leftrightarrow (1 + e^{-T^*(x,y)})^{-1} &= \frac{p(x,y)}{p(x,y) + p(x)p(y)} \\
\Leftrightarrow 1 + e^{-T^*(x,y)} &= \frac{p(x,y) + p(x)p(y)}{p(x,y)} \\
\Leftrightarrow e^{-T^*(x,y)} &= \frac{p(x)p(y)}{p(x,y)} \\
\Leftrightarrow -T^*(x,y) &= \log \frac{p(x)p(y)}{p(x,y)} \\
\Leftrightarrow T^*(x,y) &= \log \frac{p(x,y)}{p(x)p(y)}
\end{aligned}$$

concluding the proof.

B.2 Reformulation of the f-GAN Objective

The alternative formulation of the f-GAN objective used by Hjelm et al. (2018) and Poole et al. (2019) reads

$$\mathbb{E}_{p(x,y)} - \zeta(-T_\omega(x,y)) - \mathbb{E}_{p(x)p(y)}\zeta(T_\omega(x,y)) \tag{18}$$

with $\zeta(T_\omega(x,y)) = \log(1 + e^{T_\omega(x,y)})$ being the softplus function. Writing out the softplus function and using the sigmoid function $\sigma(T) = (1 + e^{-T})^{-1}$, equation (18) reads

$$\begin{aligned}
&\mathbb{E}_{p(x,y)} - \log(1 + e^{-T_\omega(x,y)}) - \mathbb{E}_{p(x)p(y)} \log(1 + e^{T_\omega(x,y)}) \\
&= \mathbb{E}_{p(x,y)} \log(1 + e^{-T_\omega(x,y)})^{-1} + \mathbb{E}_{p(x)p(y)} \log(1 + e^{T_\omega(x,y)})^{-1} \\
&= \mathbb{E}_{p(x,y)} \log(\sigma(T_\omega(x,y))) + \mathbb{E}_{p(x)p(y)} \log(\sigma(-T_\omega(x,y))) \\
&= \mathbb{E}_{p(x,y)} \log(\sigma(T_\omega(x,y))) + \mathbb{E}_{p(x)p(y)} \log(1 - \sigma(T_\omega(x,y))) \\
&= \mathbb{E}_{p(x,y)} \log(D_\omega(x,y)) + \mathbb{E}_{p(x)p(y)} \log(1 - D_\omega(x,y))
\end{aligned}$$

exactly yielding the f-GAN objective in equation (6).

C Experimental and Architectural Details

C.1 MI Estimation in High MI-Regions

We performed extensive experiments for each estimator across a large number of hyperparameters. For MINE we investigated choosing 1 or 2 hidden layers, {64, 128, 256, 512} hidden units per layer, ReLU or ELU non-linearity and a learning rate of 0.0001. For NWJ and JS we investigated choosing 1-3 hidden layers, {64, 128, 256, 512, 1024, 2048} hidden units per layer, ReLU or ELU non-linearity and learning rates [0.0001, 0.00025, 0.0005]. Each architecture was trained using Adam (Kingma & Ba (2014)) and a batch size of 512 for 1000 epochs. For selected architectures, we investigated batch sizes {64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768} and different dropout and weight decay regularizations. For batch sizes of 512 and above, one epoch corresponds

to 100 independent batches sampled from the joint distribution $p(x, y)$ leading to the same number of training steps for different batch sizes. For batch sizes below 512, one epoch corresponds to $\frac{51200}{\text{batch size}}$ independent batches drawn from $p(x, y)$.

All MI estimation plots are created from smoothed estimates. These are obtained by applying a weighted average scheme to each epoch in which the estimates in the neighbourhood of the trailing and leading 10 epochs (as far as existing) are averaged over with the weight being defined by the distance d to the original epoch: $\frac{11-d}{11}$.

C.2 MI Between a Training & Test Label Distribution

The dataset is based on the official training set of CIFAR-10 which has first been split in a custom training set of 45.000 images and a custom validation set of 5.000 images. The dataset is normalized using the channel-wise mean and standard deviation of the training set otherwise, no data augmentation is used. The ResNet and PlainNet follow the architecture outlined in He et al. (2015). However, to adapt it to CIFAR-10, the first max-pooling layer is deactivated as well as the first convolutional layer changed to a 3x3 convolution with a stride of 1 and zero padding of 1. The models are trained using SGD for 182 epochs with a momentum of 0.9 where the learning rate is decayed by a factor of 10 after epochs 91 and 137. On rotation labels, a batch size of 512 and a starting learning rate of 0.1 is used. For random labels, a batch size of 128 and a starting learning rate of 0.01 is chosen. Training on random labels fully memorizes the dataset (near 100% training accuracy, validation accuracy equalling random guessing on 10 labels).

Representations are evaluated i) using the linear evaluation protocol for which a logistic regression model is trained using Adam with for 500 epochs with a learning rate of $5 \cdot 10^{-4}$ and a weight decay set as in Kolesnikov et al. (2019); ii) using a three layer MLP with 512 hidden units per layer and ELU activations functions, dropout ($p_1 = 0.5$, $p_2 = 0.3$, $p_3 = 0.2$) and trained using Adam for 1000 epochs with a learning rate of $5 \cdot 10^{-4}$ and weight decay set to $2.5 \cdot 10^{-5}$; iii) Estimating $I(e(X); X)$ and calculating $N\hat{M}I(e(X); X)$ by training a three layer MLP critic through Jensen-Shannon maximization and using the obtained density estimator for direct plug-in MI estimation (see Section 2.3 and Appendix D.4). The critic has 512 hidden units per layer and ELU activations functions. Furthermore, dropout is employed ($p_1 = 0.5$, $p_2 = 0.3$, $p_3 = 0.2$) and weight decay set to $2.5 \cdot 10^{-5}$. Training is performed for 1000 epochs using Adam with a learning rate of $5 \cdot 10^{-4}$. All evaluation models have a batch size of 512.The evaluation hyperparameters are consistent across all experiments with ResNets and PlainNets on CIFAR-10 found in this report. For computational efficiency and following Hjelm et al. (2018) and Goyal et al. (2019), we do not use data augmentation for evaluating representations.

C.3 $\mathcal{V}$-InfoMax Principle: Experiments

Inspired by the layer wise training scheme for the information bottleneck (Elad et al. (2019)), equation 15 is optimized by alternatingly optimizing the decoder ϕ_ω to minimize the $L2$ reconstruction loss for $c_1 = 10$ steps and then optimizing the encoder for $c_2 = 1$ steps. For MNIST, the decoder is initialized by training it for 150 epochs using the fixed and randomly initialized encoder and for CIFAR-10 for 200 epochs. All of the following models are trained with a batch size of 128.

For the experiments on MNIST the encoder is chosen to be a two layer MLP with 300 hidden units per layer and an output layer of 100 units and ReLU activation functions. Optimization is performed using Adam with a learning rate of $2.5 \cdot 10^{-4}$, weight decay set to 10^{-5} and dropout ($p_1 = 0.5$, $p_2 = 0.5$). The linear decoder model is a logistic regression model, the non-linear decoder model has the same architecture as the encoder but without dropout layers and 10 units in the last layer. For MNIST, no data augmentation is performed. Representations are evaluated i) using the linear evaluation protocol for which we use the SAGA Defazio et al. (2014) implementation in *scikit-learn* Pedregosa et al. (2011) and ii) using a two layer MLP with 300 hidden units per layer trained for 50 epochs using Adam with a learning rate of 10^{-4}, weight decay 10^{-5}.

For the experiments on CIFAR-10, the ResNet18 is equivalent to the one described in C.2. The ResNet is trained using Adam with a learning rate and weight decay of 10^{-4}. The decoder is equivalently trained using Adam with a learning rate of 10^{-4} but a weight decay of 10^{-5}. The decoder is either a logistic regression model or a three layer MLP with 512 hidden units per layer

and ELU activation functions (output layer has dimension of CIFAR-10). Training is performed with the commonly used data augmentations employed on CIFAR-10 (following He et al. (2015)) which includes random horizontal flipping and random cropping. Images are normalized using the channel-wise mean and standard deviation. Representations are evaluated using the linear evaluation protocol and a non-linear evaluation model as outlined in appendix C.2.

C.4 $\mathcal{V}$-INFOMAX PRINCIPLE: COMBINATION WITH DOMAIN SPECIFIC PRETEXT TASKS

The ResNet and PlainNet models as well as the decoder models which together form the supervised autoencoder are chosen exactly as detailed in appendix C.2. Then, the models are jointly trained using the SGD learning scheme likewise outline in appendix C.2. The representations are evaluated again following appendix C.2 where we choose to use the representations from the epoch achieving best accuracy on the pretext task (Kolesnikov et al. (2019)). For $\lambda = \infty$ we choose representations from the epoch corresponding to the best $L2$ reconstruction loss.

D FURTHER RESULTS

D.1 EVALUATING REPRESENTATIONS THROUGH ESTIMATING $I(e(X); Y)$

To investigate the effect of different neural MI estimators for measuring $I(e(X); Y)$, a LeNet-5 variant Lecun et al. (1998) is trained using Adam with a learning rate of $5 \cdot 10^{-4}$ for 50 epochs. Three representations are extracted, $L0$ and $L1$ corresponding to the first and second max pooling layer consisting of 1176 and 784 features and $L2$ corresponding to the 120 dimensional fully connected hidden layer which directly connects to the 10-dimensional output layer. As input data MNIST is chosen with $p \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$ percentage of pixels randomly set to 0 or 1 (salt & pepper noise) in the training and test set. The classification results are summarized in table 6.

p	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Accuracy in %	99.15	98.51	97.92	96.59	94.25	89.09	81.51	67.74	46.13	23.77	11.37

Table 6: Classification results on official test split of MNIST with or without salt & pepper noise.

Table 6 suggests that $I(e(X); Y)$ for $p = 0.0$ should be close to $H(Y) = log(10) = 2.30$ nats and gradually decrease to 0 nats for $p = 1.0$.

The results of training MINE & NWJ using 2-layer MLP-critic with 512 hidden units per layer and Adam with $lr = 0.0001$ on this dataset can be seen in figure 8. These plots show multiple things:

- MINE & NWJ behave very similar for evaluating representations
- MINE & NWJ can be employed for continuous-discrete problems with low underlying MI
- As expected, $\hat{I}(e(X); Y)$ for $p = 0.0$ is close to 2.3 nats and zero for $p = 1.0$. In between the estimators behave **monotonicly** (except for a slight deviation at $p = 0.4$ in $L2$)
- The estimators scale well to higher input dimensions
- For very low MI and large input dimension, the estimators can peak at a certain estimation value and then decrease. The peak is of same height as the observed saturation point in $\hat{I}(L2; Y)$. Additionally, so not shown, for the training set $\hat{I}(e(X); Y)$ always continuously increases. Therefore, $\hat{I}(e(X); Y)$ can show a classic overfitting behaviour and the best estimate for $\hat{I}(e(X); Y)$ is chosen as the maximal $\hat{I}(e(X); Y)$ achieved on the validation set.

Additionally, the MI between the layers is comparable but could be marginally higher at a later layer (see Table 7). This is in disagreement of the data processing inequality and highlights the limitations of MI estimation in practice due to limitations on the estimating model families. As for evaluation purposes, a two layer MLP is trained on $L0$ but $L1$ includes a second convolutional processing of the input data. Therefore, the same two layer MLP critic could capture information in $L1$ for which its

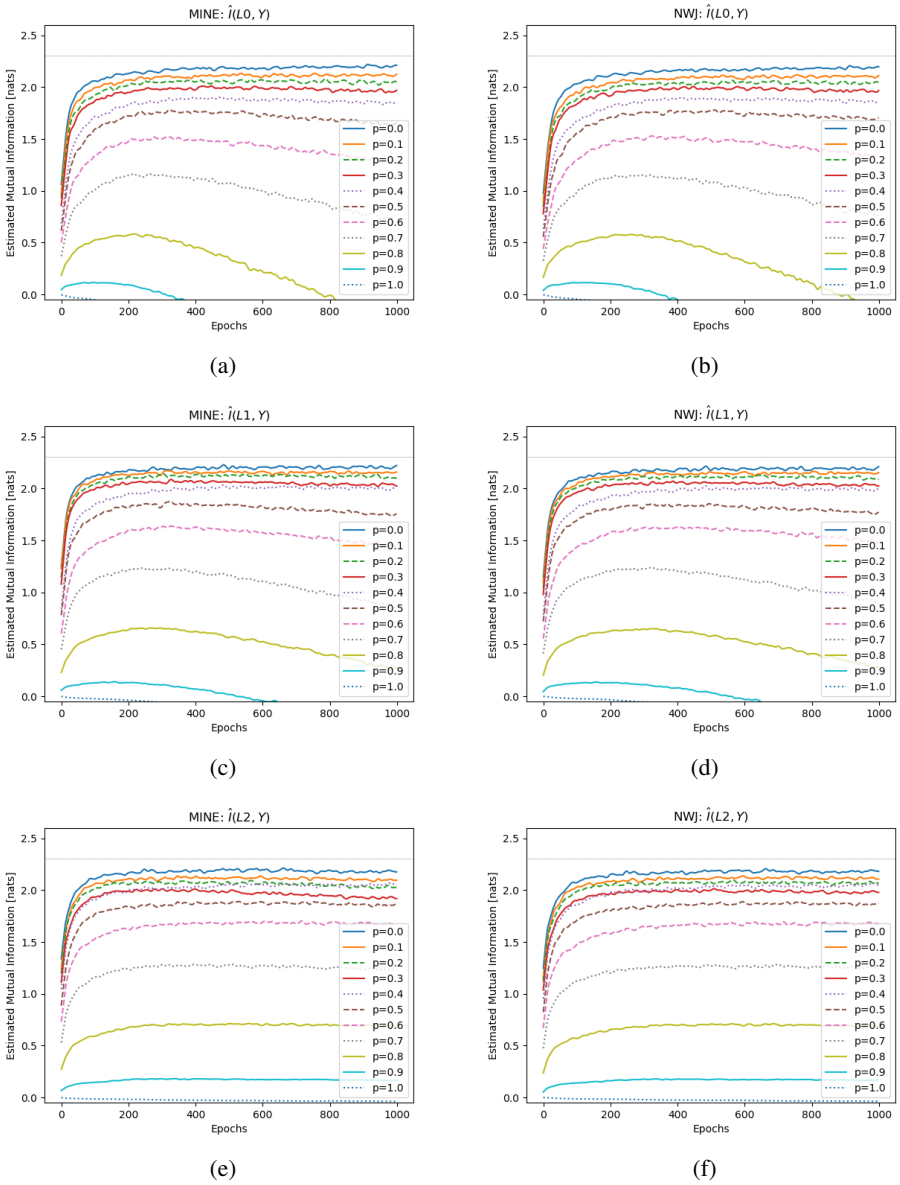


Figure 8: Training MINE and NWJ on representations of a LeNet-5 variant trained on MNIST with a different proportion p of salt & pepper noise applied to the training & test images. The dotted gray line corresponds to $H(Y) = log(10) = 2.30$ nats.

capacity to measure it in $L0$ is not sufficient (see $\mathcal{V}$-information discussion in Section 2.6 and 5.1). However, as can be seen in figure 8 and table 7, these effects are only marginally pronounced. The seen fluctuation could equally well be understood as random fluctuations and the "lost" information through layer-wise transformations being so small that it defies the resolution of the MI-estimators. If only random fluctuations in the MI estimator would cause the small lower value for $L0$ than $L1$ in table 7, one would expect equally many cases for which $L0 > L1$. However, we find looking at $L0$ and $L1$ for different p that $L0$ is consistently slightly below $L1$ indicating that the critic network indeed struggles to capture all the shared information in the representations about the labels for $L0$ compared to $L1$.

Figure 9 shows the same experiment but estimating MI in training a log-density ratio estimator through Jensen-Shannon divergence maximization and using it to plug it into the NWJ bound as

well as direct plug-in MI estimation. As a critic network, a three layer MLP with 512 units each with regularization (dropout $p_0 = 0.5$, $p_1 = 0.3$, $p_0 = 0.2$, weight decay = $2.5 \cdot 10^5$) using Adam (lr = 0.0001) is trained. For the NWJ bound, the exact same insights as for direct MINE and NWJ $I(e(X); Y)$ estimation hold. For plug-in estimation of $I(e(X); Y)$ one can observe that the maximum of $\hat{I}(e(X); Y)$ slightly overestimates mutual information. Therefore, using a log-density ratio estimator for the investigated MI range, these empirical results suggest that the true underlying MI can be given as an element of an interval lower bounded by plugging the estimator into the NWJ bound and upper bounded by direct plug-in estimation. For plug-in estimation one also observes that a small hump develops early in training for a true underlying MI close to 2.3 nats slightly exceeding the 2.3 nats threshold. Nevertheless, monotonicity holds and we find that this bump does not occur using the same critic when evaluating representations for CIFAR-10.

$p = 0.0$	$L0$	$L1$	$L2$
MINE	2.212	2.227	2.212
NWJ	2.207	2.213	2.206

Table 7: Maximal $\hat{I}(e(X); Y)$ obtained through MINE and NWJ for $p = 0.0$ in nats.

We did not explore the *InfoNCE* estimator for measuring $\hat{I}(e(X); Y)$. One can expect it to work very well for datasets where $|Y|$ is small as for MNIST or CIFAR-10 ($|Y| = 10$). However, due to contrasting each positive sample with all possible $N - 1$ negative samples, it is significantly more expensive to use InfoNCE for measuring the quality of a representation. If $|Y|$ increases to 1024 as for ImageNet a batch size of 1000 has to be chosen so as to guarantee that InfoNCE does not saturate below the true $I(e(X); Y)$. Then, one evaluation of the InfoNCE bound already takes roughly one million critic evaluation making it computationally infeasible[12].

D.2 Different Negative Sampling Scheme

We investigate the effects of changing the negative sampling scheme from shuffling y to choosing all possible negative (x_i, y_j) pairs for each bounds for architectures with 1 hidden layer and 64 hidden units and 2 hidden layers and 512 hidden units, ReLU and ELU non-linearity, learning rate of 0.0001 (Adam) and a batch size of 64 samples leading to $64 \cdot (64 - 1)$ negative samples and 64 positive samples. (Note: $log_e(64 \cdot (64 - 1)) = 8.30$ and $log_e(64) = 4.16$)

Figure 10 shows that the MINE became significantly more unstable showing rapid divergence even for a small critic network (see Figure 10a) and for a higher capacity critic even misbehaviour for small true MI's (see Figure 10b). No clear correlation to the logarithm of the positive or negative samples can be established.

Figure 11 shows that this negative sampling scheme leads NWJ to not show the saturation phenomenon. However, it does not only increase computation time significantly, it also leads to very high variance in the estimates even for a very small critic network (see Figure 11a). For a higher capacity critic (see Figure 11b), the variance problem even increases and for the small 2.5 nats regime, it even loses its property to be a lower bound, this may be due to numerical issues.

Figure 12 showcases that optimizing for the JS divergence instead and plugging the obtained estimator into NWJ yields a very unstable behaviour of the estimator even for a very small critic network (see Figure 12a). Choosing a more complex critic worsens the instability. We believe the instability can be reduced through regularization as shown in figure 3c. When the estimates are stable as for example for 30 nats, the estimator indeed shows reduced variance compared to direct optimization using NWJ as mentioned by Poole et al. (2019). Figure 12b shows that direct MI estimation suffers from the same loss of monotonicity in the estimate as using the sampling scheme from Belghazi et al. (2018) in which higher true MI values can lead to lower MI estimates. Interestingly, the stability of direct MI estimation is not worsened which is in line with findings by Hjelm et al. (2018) that optimizing for JS divergence results in representation which perform equally well w.r.t. to the number of negative samples.

[12] InfoNCE's efficiency scales quadratically with the used batch size.

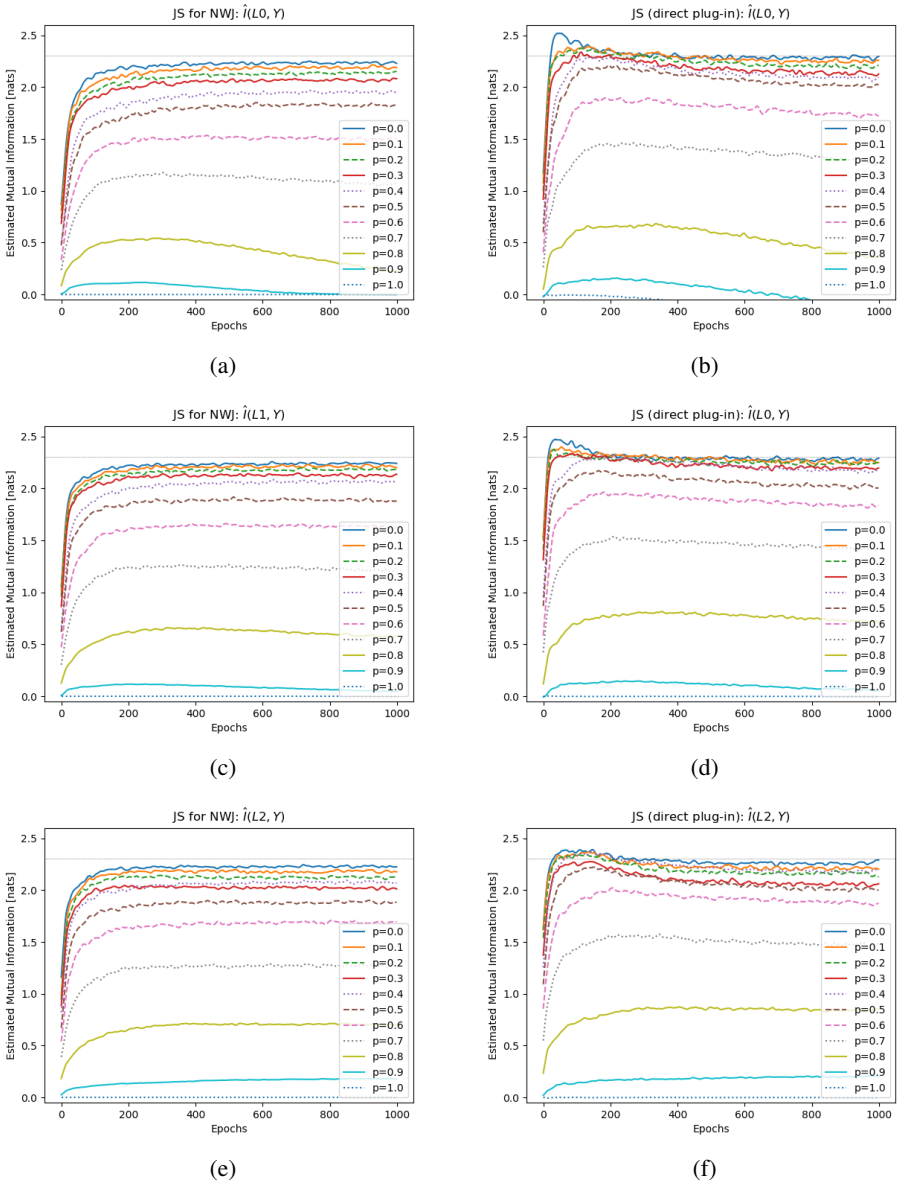


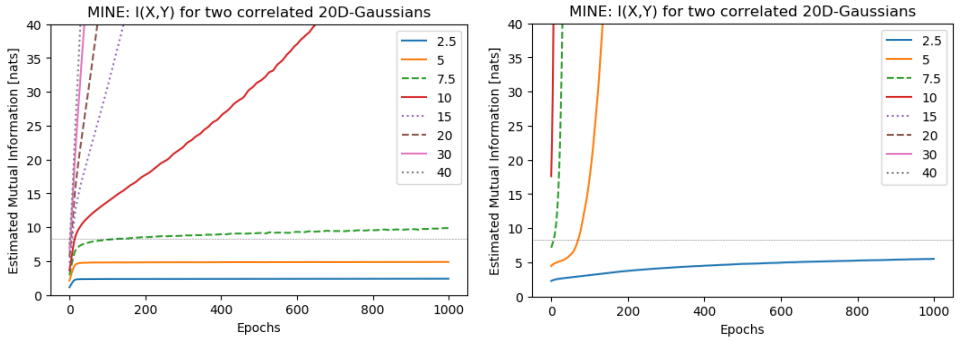
Figure 9: Training a log-density ratio estimator through Jensen-Shannon divergence maximization and using it to plug it into the NWJ bound as well as direct plug-in MI estimation. Representations are of a LeNet-5 variant trained on MNIST with a different proportion p of salt & pepper noise applied to the training & test images. The dotted grey line corresponds to $H(Y) = log(10) = 2.30$ nats.

D.3 JS

Figure 13 shows that without regularization higher capacity critics yield not well-behaved MI estimation results.

D.4 Mutual Information with a Downstream Label Distribution

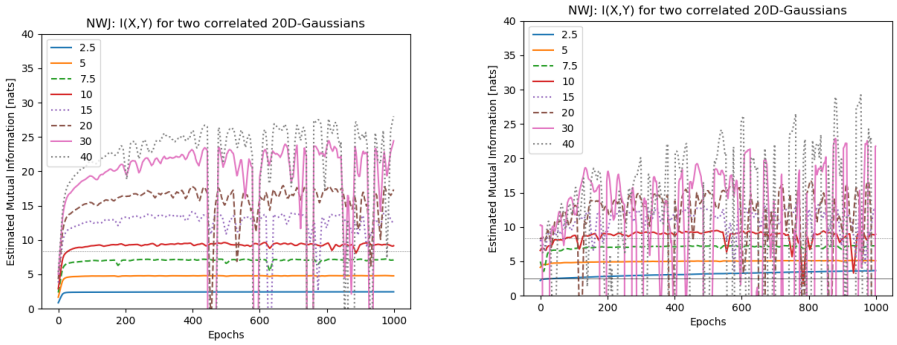
Table 8 shows the results of training a plain convolutional network on the auxiliary labelings.



(a) Critic: 1-Layer MLP with 64 hidden units and ReLU activations

(b) Critic Network: 2-Layer MLP with 512 hidden units per layer and ReLU activations

Figure 10: MI estimation with MINE between two 20D-Gaussians and negative sampling as used in Poole et al. (2019). The grey dotted line corresponds to $log_e(64 \cdot (64-1)) = 8.30$. Each line corresponds to a different correlation chosen and is labeled based on the resulting underlying MI in nats. One epoch corresponds to 100 independently drawn batch-samples.



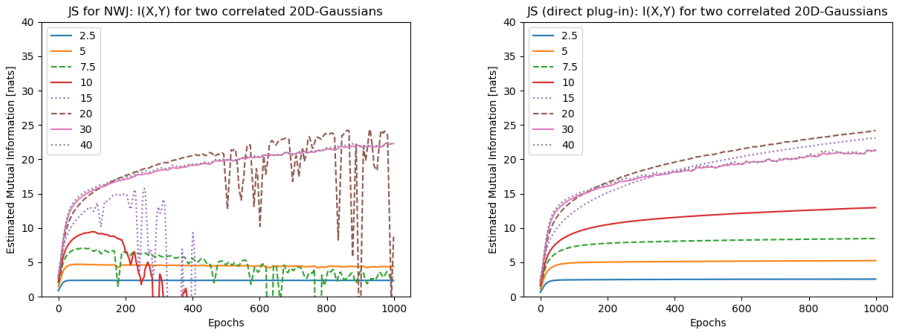
(a) Critic: 1-Layer MLP with 64 hidden units and ELU activations

(b) Critic Network: 2-Layer MLP with 512 hidden units per layer and ELU activations. Black horizontal line corresponds to 2.5 nats.

Figure 11: MI estimation with NWJ between two 20D-Gaussians and negative sampling as used in Poole et al. (2019). The grey dotted line corresponds to $log_e(64 \cdot (64-1)) = 8.30$. Each line corresponds to a different correlation chosen and is labeled based on the resulting underlying MI in nats. One epoch corresponds to 100 independently drawn batch-samples.

D.5 $\mathcal{V}$-INFOMAX PRINCIPLE

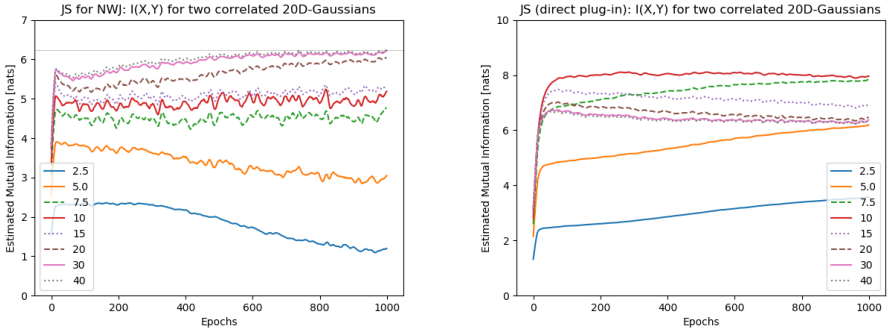
Figure 14 shows that if we choose the alternative training scheme of jointly taking the gradient with respect ot the parameters of the encoder and decoder, the qualitative behaviour of the representations learned using the $\mathcal{V}$-InfoMax principle (15) stays the same. Albeit less pronounced, they are highly dependent on the choice of decoder family in conjunction with the corresponding evaluation protocol. Figure 14 especially showcases that learning representations through maximizing information measured with a linear decoder family significantly pushes the representations to form linearly separable features for the downstream task of image classification.



(a) Critic: 1-Layer MLP with 64 hidden units and ELU activations

(b) Critic: 1-Layer MLP with 64 hidden units and ELU activations

Figure 12: MI estimation with JS between two 20D-Gaussians and negative sampling as used in Poole et al. (2019). a) Using the density estimator to plug into the NWJ lower bound. b) Using the density estimator as a direct plug-in estimator for $I(X;Y)$. Note that one epoch corresponds to 100 sampled batches. Therefore, the training behaviour reported in Poole et al. (2019) or Song & Ermon (2019) correspond exactly to the first 50 and very roughly to first 200 epochs as we do not incrementally increase MI during training.



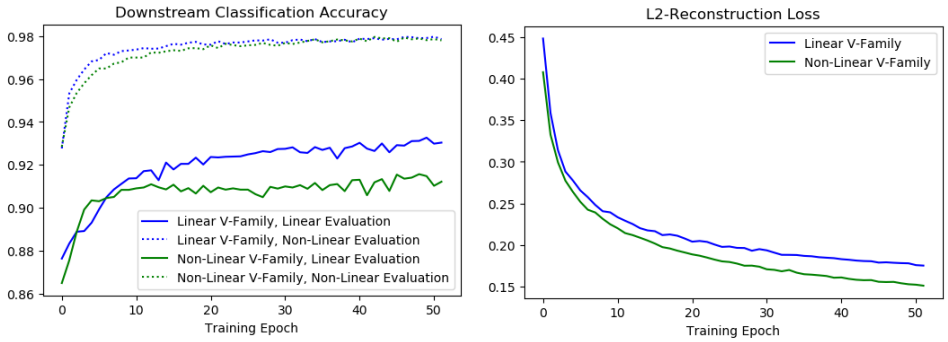
(a) Critic: 2-Layer MLP with 512 hidden units

(b) Critic: 2-Layer MLP with 512 hidden units

Figure 13: MI estimation through Jensen-Shannon divergence maximization between two 20D-Gaussians. *JS for NWJ* denotes using the density estimator to plug into the NWJ lower bound and *JS (direct plug-in)* using the density estimator as a direct plug-in estimator for $I(X;Y)$. The critic network is trained without regularization.

	PlainNet18					
$N\hat{M}I(e_\theta(X);Y)$	B0	B1	B2	B3	B4	Prelogits
Rotation Labels Y_2'	**0.53**	**0.59**	**0.69**	**0.69**	**0.45**	**0.46**
Random Labels Y_1'	0.43	0.42	0.37	0.21	0.01	0.04
Random Initialization	0.49	0.40	0.30	0.24	0.20	0.16
Linear Down. Accuracy	B0	B1	B2	B3	B4	Prelogits
Rotation Labels Y_2'	**43.83**	**51.78**	**57.45**	**59.94**	**35.12**	**32.53**
Random Labels Y_1'	36.34	38.10	35.64	27.60	17.12	17.28
Random Initialization	40.72	38.35	32.39	26.76	23.07	19.03

Table 8: Evaluating the representations learned after different convolutional blocks of a PlainNet18 (see He et al. (2015)) trained on rotation or random labels of the enlarged CIFAR-10 dataset. Evaluation done using the normalized mutual information score $N\hat{M}I(e_\theta(X);Y)$ between the learned representations and the downstream labels and the common linear evaluation protocol (see Section 2.5). It shows exactly the same behaviour as observed for residual networks (see Table 1). Even though, $I(Y;Y_2') = I(Y;Y_1') = 0$, predicting rotations clearly outperforms random labels and the random initialization baseline by large margins whereas training on random labels even worsens the representation compared to the baseline.



(a) Validation accuracies of the learned representations using the relaxed $\mathcal{V}$-InfoMax principle.

(b) L2-loss reconstruction loss of the decoder on the validation set.

Figure 14: Training a MLP encoder with two hidden layers of 300 units and a linear output layer with 100 units on MNIST using $\mathcal{V}$-InfoMax objective 15 but taking joint gradients during training. The linear and non-linear model classes are chosen as in figure 5. (a) Shows that downstream linear accuracy is significantly boosted by choosing a linear decoder family. The non-linear downstream accuracy of both decoder families is comparable. However, notice that the linear downstream accuracy of the non-linear decoder families stays on a plateau throughout most training epochs whereas non-linear performance consistently increases. (b) Shows that the non-linear $\mathcal{V}$-family again achieves higher predictive $\mathcal{V}$-information about X.