
Learning deep representations by mutual information estimation and maximization

R Devon Hjelm

MSR Montreal, MILA, UdeM, IVADO
devon.hjelm@microsoft.com

Alex Fedorov

MRN, UNM

Samuel Lavoie-Marchildon

MILA, UdeM

Karan Grewal

U Toronto

Adam Trischler

MSR Montreal

Yoshua Bengio

MILA, UdeM, IVADO, CIFAR

Abstract

Many popular representation-learning algorithms use training objectives defined on the observed data space (which we call “pixel-level”). This may be detrimental when only a small fraction of the bits of signal actually matter at a semantic level. We hypothesize that representations should be learned and evaluated more directly in terms of their information content and statistical or structural constraints. To address the first quality, we consider learning unsupervised representations by maximizing mutual information between part or all of the input and a high-level feature vector. To address the second, we control characteristics of the representation by matching to a prior adversarially. Our method, which we call Deep INFOMAX (DIM), can be used to learn representations with desired characteristics and which empirically outperform a number of popular unsupervised learning methods on classification tasks. DIM opens new avenues for unsupervised learning of representations and is an important step towards flexible formulations of representation-learning objectives catered towards specific end-goals.

1 Introduction

At a conscious level, intelligent agents make predictions and plans not at the level of pixels and other sensors, but at the level of abstract representations (Bubic et al., 2010). This may be more appropriate, because the number of semantically relevant bits (in speech, e.g., phonemes, the identity of the speaker, prosody, etc.) is a tiny fraction of the total number of bits in the raw signal (Bengio, 2017). However, most unsupervised machine learning is based on training objectives defined at least partially in input space (e.g., Vincent et al., 2008; Kingma & Welling, 2013; Makhzani et al., 2015; Dumoulin et al., 2016; Donahue et al., 2016). Since such objectives can be optimized well without capturing these few semantically relevant bits, they may not yield good representations. As one of deep learning’s core objectives is to discover “good” representations, we ask: *is it possible to learn representations with a training objective that is not defined in input space?* The simple idea explored here is to train the representation-learning function (i.e., the encoder) to maximize the mutual information between its inputs and outputs.

Mutual information is notoriously difficult to compute, particularly in continuous and high-dimensional settings. Thankfully, recent advances in *neural estimation* (Belghazi et al., 2018) enable effective computation of mutual information between high dimensional input / output pairs of deep neural networks, and in this work we leverage these techniques for representation learning. As we will show, however, maximizing mutual information between the complete input and its representation (i.e., *global* mutual information) alone is not sufficient for learning useful representations, depending on the downstream task. Rather, maximizing the average mutual information between the representation and *local* regions of the input can greatly improve the representation’s quality for, e.g.,

classification tasks, while global mutual information plays a stronger role in the ability to reconstruct the full input given the representation.

Usefulness of a representation is not just a matter of information content: representational characteristics like structure also play an important role (Gretton et al., 2012; Hyvärinen & Oja, 2000; Hinton, 2002; Schmidhuber, 1992; Bengio et al., 2013; Thomas et al., 2017). We therefore combine mutual information maximization with prior matching in a manner similar to adversarial autoencoders (AAE, Makhzani et al., 2015) or BiGAN (Donahue et al., 2016; Dumoulin et al., 2016), to attain representations with desired constraints as well as good downstream performance. Our full approach is close to the INFOMAX optimization principle (Linsker, 1988; Bell & Sejnowski, 1995), so we call our method *deep INFOMAX* (DIM).

Our contributions are as follows:

- We formalize deep INFOMAX (DIM), which uses mutual information neural estimation (MINE, Belghazi et al., 2018) to explicitly maximize the mutual information between input data and learned high-level representations.
- Our mutual information maximization can prioritize global or locally-consistent information, which we show can be used to tune the suitability of learned representations for classification or reconstruction-style tasks.
- We use adversarial learning (à la Makhzani et al., 2015) to constrain the representation to have desired statistical characteristics specific to a prior.
- We introduce two new measures of representation quality, one based on MINE and the other a dependency measure based on the work by Brakel & Bengio (2017), and we use these to compare representations of different unsupervised methods.

2 Motivation and Background

2.1 Related work

Generative models in representation learning Generative models are commonly used as methods for building representations (Vincent et al., 2010; Kingma et al., 2014; Salimans et al., 2016; Rezende et al., 2016; Donahue et al., 2016), and mutual information plays an important role in their representational quality. In generative models that rely on reconstruction (e.g., denoising, variational, and adversarial autoencoders, Vincent et al., 2008; Rifai et al., 2012; Kingma & Welling, 2013; Makhzani et al., 2015), the negative reconstruction error can be related to the mutual information in the encoder as,

$$\mathcal{I}_e(X, Y) \geq \mathcal{H}_e(Y) - \mathcal{R}_{e,d}(X), \quad (1)$$

where X and Y are random variables corresponding to the input and an intermediate representation (e.g., the bottleneck), $\mathcal{R}_{e,d}(X)$ is the reconstruction error, and $\mathcal{H}_e(Y)$ is the marginal entropy of the encoder output. Thus, models with reconstruction-type objectives provide some guarantees on the amount of information encoded in their intermediate representations. Similar guarantees exist for bi-directional adversarial models (Dumoulin et al., 2016; Donahue et al., 2016), which adversarially train an encoder / decoder to match their respective joint distributions (which typically increases the marginal entropy in Equation 1) or minimizes the reconstruction error (Chen et al., 2016).

In generative adversarial networks (GANs, Goodfellow et al., 2014), the representation of the discriminator has been shown to be suitable for some downstream tasks (Salimans et al., 2016; Kumar et al., 2017). While little explanation is provided on why GANs should learn good representations, we posit that high mutual information in the discriminator representation with the input is necessary to classify between real and fake images, particularly as the generator becomes very good, though we leave exploring this to future work.

The training objective of a decoder for reconstruction matches individual pixels (with the original input) while the training objective of a GAN generator matches the generated image according to the expectations of the discriminator. Learning a generative model can be difficult with high-dimensional data, such as Imagenet (Krizhevsky et al., 2012), audio, or video data, as well as in many reinforcement learning settings, which potentially limits the applicability of generative models in representation learning. The biggest issue is that, as the dimension of the data becomes large,

the objectives of these models become dominated by the pressure to represent all of the pixels well, whereas it may be that a smaller set of dimensions suffice to capture the important characteristics of the distribution. The models above must represent *all* statistically significant variability in the data in order to generate (or even discriminate) properly, and much of this variability could be unimportant and even detrimental to the suitability of a representation for downstream tasks.

Decoder-free representation learning There are many popular decoder-free methods for learning representations. Some of the more traditional methods, such as independent component analysis (Bell & Sejnowski, 1995), which is linear, and self-organizing maps (Kohonen, 1998), which require that the latent space lie on a grid, generally cannot represent complex relationships in data. Deep approaches that rely on maximum likelihood exist (Dinh et al., 2014, 2016), but these rely on strictly constraining the encoder and / or the output space in order to formulate a tractable likelihood objective. Deep clustering (Xie et al., 2016; Chang et al., 2017) learns decoder-free representations for the purposes of performing unsupervised clustering, and these models could be used for a wide variety of downstream tasks, though there is limited work in this direction.

Noise as targets (NAT, Bojanowski & Joulin, 2017) is an unsupervised learning algorithm that rephrases learning a representation as a supervised learning problem with noise targets. NAT does not use a generative model, and while NAT requires a cumbersome inference mechanism (using the Hungarian algorithm) to assign inputs to noise counterparts, it is able to learn representations that perform competitively against many unsupervised methods that use generative models in transfer learning and semi-supervised learning tasks. However, NAT requires that the target samples are drawn prior to training, the number of which needs to match the number of input samples, and it is unclear how NAT performance is affected by the size of the input dataset and the dimensionality of the representation.

Mutual-information estimation Mutual information-based objectives have a long tradition in unsupervised learning of features. The INFOMAX optimization principle (Linsker, 1988; Bell & Sejnowski, 1995), as prescribed for neural networks, advocates maximizing mutual information between the input and output. This is the basis of numerous ICA algorithms, some of which are nonlinear (Hyvärinen & Pajunen, 1999; Almeida, 2003), but none of which are general enough to apply to deep neural networks. Mutual information neural estimation (MINE, Belghazi et al., 2018) learns a neural estimate of the mutual information of continuous variables, is strongly consistent, and can be used to learn better implicit bi-directional generative models (e.g., with better reconstruction) by maximizing mutual information between the input and output of the encoder. MINE is a fundamental building block of deep INFOMAX (DIM) as presented here, though we show that good representations can be learned without the generator / decoder in this context. In addition, we find that using a simple classifier based on the Jensen-Shannon divergence (rather than the traditional KL-divergence) is both more stable and provides better results. Our method further leverages local structure in the input, which can be used to improve suitability of representations for classification.

Leveraging known structure in the input in mutual information maximization-based objectives is nothing new (Becker, 1992; Wiskott & Sejnowski, 2002), and some very recent works also follow this intuition. Unsupervised clustering and segmentation is attainable by maximizing the mutual information between encoded nearby patches, which can be done by using an autoencoder loss (Ji et al., 2018). Very recent work that is mostly independent to ours, contrastive predictive coding (CPC, Oord et al., 2018), uses a mutual information estimate-based approach to perform prediction on other patches in an image (and, by extension, other timesteps in a sequence). Their work shares similarities to DIM, as they leverage negative samples on locations of a feature map, although they estimate mutual information between patches using autoregression. DIM, in contrast, uses mutual information w.r.t. a global summary vector (i.e., a single representation for the complete input), and requires only 1×1 convolutions to perform mutual information estimation between local patches and said summary vector. In addition, our work looks at the suitability of representations across two different mutual information maximization objectives (local vs. global), as we believe representations have different utility based on the end-goal.

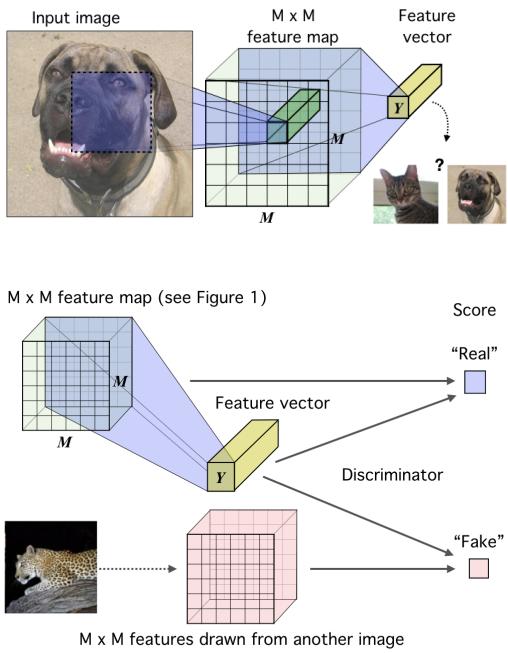


Figure 1: The base encoder model in the context of image data. An image (in this case) is encoded into a convolutional network until reaching a feature map of $M \times M$ feature vectors corresponding to $M \times M$ input patches. These vectors are summarized (for instance, using additional convolutions and fully-connected layers) into a single feature vector, Y . Our goal is to train this network such that relevant information about the input is extractable from the high-level features.

Figure 2: Deep INFOMAX (DIM) with a global $\text{MI}(X; Y)$ objective. Here, we pass both the high-level feature vector, Y , and the lower-level $M \times M$ feature map (See Figure 1) through a discriminator composed of additional convolutions, flattening, and fully-connected layers to get the score. Fake samples are drawn by combining the same feature vector with a $M \times M$ feature map from another image.

3 Deep INFOMAX

3.1 Encoder and DIM objective

Here we outline the general setting of training an encoder to maximize mutual information between its input and output. Let \mathcal{X} and \mathcal{Y} be the domain and range of a continuous and (almost everywhere) differentiable parametric function, $E_\psi : \mathcal{X} \rightarrow \mathcal{Y}$ with parameters ψ (e.g., a neural network). These parameters define a family of encoders, $\mathcal{E}_\Phi = \{E_\psi\}_{\phi \in \Phi}$ over Φ . Let us assume that we are given a set of training examples on an input space, \mathcal{X} : $\mathbf{X} := \{x^{(i)} \in \mathcal{X}\}_{i=1}^N$, with empirical probability distribution, \mathbb{P} . We can define $\mathbb{U}_{\psi, \mathbb{P}} = E_\psi \# \mathbb{P}$ to be the marginal *push-forward distribution* (Bottou et al., 2017) induced by pushing samples from \mathbb{P} through E_ψ , as well as the joint and the product of marginals:

$$\begin{aligned} \mathbb{U}_{\psi, \mathbb{P}}(Y = y) &:= \mathbb{P}(\{x^{(i)} \in \mathbf{X} \mid E_\psi(x^{(i)}) = y\}), \\ \mathbb{J}_{\psi, \mathbb{P}}(X = x, Y = y) &:= \mathbb{P}(X = x)\delta_y(E_\psi(x)), \\ \mathbb{M}_{\psi, \mathbb{P}}(X = x, Y = y) &:= \mathbb{P}(X = x)\mathbb{P}(\{x^{(i)} \in \mathbf{X} \mid E_\psi(x^{(i)}) = y\}), \end{aligned} \quad (2)$$

where $\delta_y(E_\psi(x))$ is the Dirac measure on Y .

An example encoder for image data is given in Figure 1, which will be used in the following sections, but our formulation can easily be adapted for temporal data. Similar to the INFOMAX optimization principle (Linsker, 1988), we assert that our encoder should be trained according to the following objectives:

- **Mutual information maximization:** Find the set of parameters, ψ , such that the mutual information, $\mathcal{I}(X; E_\psi(X))$, is maximized. Depending on the end-goal, this maximization can be done over the complete input, X , or some local subset.
- **Structural constraints:** Depending on the end-goal for the representation, the marginal $\mathbb{U}_{\psi, \mathbb{P}}$ should match a prior distribution, \mathbb{V} .

The formulation of these two objectives covered below we call *Deep INFOMAX (DIM)*.

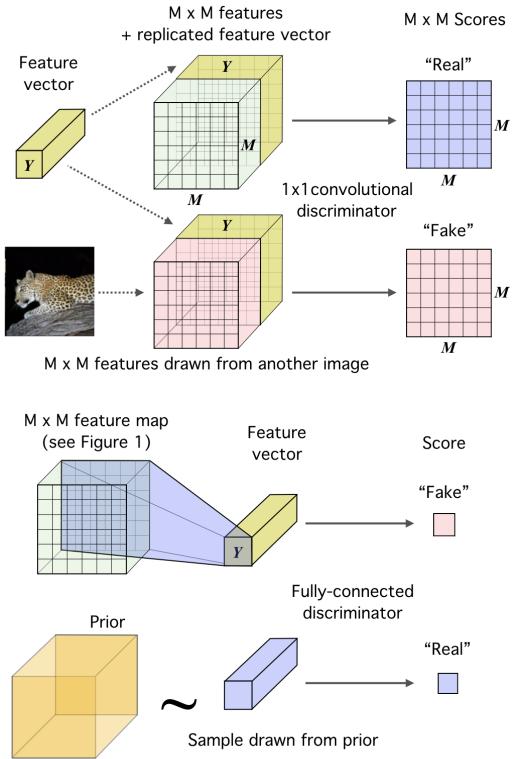


Figure 3: Maximizing mutual information between local feature and high level feature vector. First we encode the image to a feature map that maintains some structural aspect of the data, in this case location, and we summarize this feature map into a global feature vector (see Figure 1). This feature vector then is concatenated with the lower-level feature map *at every location*. A 1×1 convolutional discriminator is then used to score the “real” feature map / feature vector pair, while the “fake” pair is produced by pairing the feature vector with a feature map from another image.

Figure 4: Matching the output of the encoder to a prior. “Real” samples are drawn from a prior while “fake” samples from the encoder output are sent to a discriminator. The discriminator is trained to distinguish between (classify) these sets of samples. The encoder is trained to “fool” the discriminator.

3.2 Mutual information estimation and maximization

Our basic mutual information maximization framework is presented in Figure 2. The approach follows Mutual Information Neural Estimation (MINE Belghazi et al., 2018), which estimates mutual information by training a discriminator (a classifier) to distinguish between samples coming from the joint, \mathbb{J} , and the product of marginals, \mathbb{M} , of two random variables, X and Y . MINE relies on a lower-bound to the mutual information based on the Donsker-Varadhan representation (Donsker & Varadhan, 1983) of the KL-divergence,

$$\mathcal{I}(X; Y) := \mathcal{D}_{KL}(\mathbb{J} || \mathbb{M}) \geq \widehat{\mathcal{I}}_\omega(X; Y) := \mathbb{E}_{\mathbb{J}}[T_\omega] - \log \mathbb{E}_{\mathbb{M}}[e^{T_\omega}], \quad (3)$$

where $T_\omega : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ is a discriminator function modeled by a neural network with parameters ω .

At a high level, we basically use MINE to optimize the encoder E_ψ , simultaneously estimating and maximizing $\mathcal{I}(X, E_\psi(X))$,

$$(\hat{\omega}, \hat{\psi}) = \arg \max_{\omega, \psi} \widehat{\mathcal{I}}_\omega(X; E_\psi(X)). \quad (4)$$

However, there are some important differences. First, because the encoder and MINE are optimizing the same objective and have similar computational pathways (e.g., convolutional neural networks), we combine the initial layers of both networks, so that $E_\psi = f_\psi \circ C_\psi$ and $T_{\psi, \omega} = D_\omega \circ C_\psi$.¹

Second, rather than base our estimate of mutual information on the KL-divergence, we use the Jensen-Shannon divergence (JSD). Following the formulation provided in Nowozin et al. (2016), the combined encoder and MINE objective then becomes,

$$\begin{aligned} (\hat{\omega}, \hat{\psi})_G &= \arg \max_{\omega, \psi} \widehat{\mathcal{I}}_{\omega, \psi}^{(JSD)}(X; E_\psi(X)) \\ &= \arg \max_{\omega, \psi} \mathbb{E}_{\mathbb{J}_{\psi, \mathbb{P}}}[-\text{sp}(-T_{\psi, \omega}(x, y(x)))] - \mathbb{E}_{\mathbb{M}_{\psi, \mathbb{P}}}[\text{sp}(T_{\psi, \omega}(x', y(x)))], \end{aligned} \quad (5)$$

¹Here we slightly abuse the notation and use ψ for both parts of E_ψ .

where x is an input sample, $y(x)$ is the high-level representation, x' is another input sample (unrelated to $y(x)$), and $\text{sp}(z) = \log(1 + e^z)$ is the softplus function.

Our new estimator based on the JSD and the original based on the KL should behave similarly: both act like classifiers whose objectives maximize the expected log-ratio of the joint over the product of marginals. However, the JSD has some important properties that may make it more suitable for maximizing mutual information simultaneously with estimation. First, the mutual information is potentially unbounded for deterministic functions, which could lead to additional instabilities. The Jensen-Shannon, in contrast, has an upper-bound of $\log 2$, which in principle does not encourage our estimator nor encoder to produce very large numbers. Second, the log-expectation in the Donsker-Varadhan representation of the KL introduces biased gradients (which is admittedly correctable, see Belghazi et al. (2018) for details), while the gradients from Equations 5 are unbiased. More to the point, our initial experiments in maximizing information found using the JSD objective was consistently superior, so we use it in DIM.

3.3 Local mutual information maximization

The objective above can be used to maximize mutual information between input and output, but ultimately this may be undesirable depending on the task. For example, trivial noise local to pixels is useless for image classification, so a representation may not benefit from encoding this information if the end-goal is to classify (e.g., in zero-shot learning, transfer learning, etc). In order to obtain a representation model suitable for classification, we maximize the average mutual information between the high-level representation and local patches of the image.

Because the same representation is encouraged to have high mutual information with all the patches, this favours the aspects of the data which are shared across patches. Suppose the feature vector, Y , is of limited capacity (number of units and range) and assume the encoder does not support infinite output configurations. For maximizing the mutual information between the whole input and the representation, Y , the encoder can pick and choose what type of information in the input is passed through the encoder, such as noise specific to local patches or pixels. However, if the encoder is passing information specific to only some parts of the input, this *does not increase* the mutual information with any of the other patches that do not contain said noise. This encourages the encoder to prefer information that is *shared* across the input, and this hypothesis is supported in our experiments below.

Our local mutual information maximization framework is presented in Figure 3. First we encode the image to a feature map, $C_\psi(x) := \{C_\psi^{(i)}(x)\}_{i=1}^{M \times M}$ that maintains some structural aspect of the data (such as location), indexed in this case by i . Next, we summarize this feature map into a global feature vector, $y(x) = f_\psi \circ C_\psi(x) := E_\psi(x)$. This feature vector then is concatenated with the lower-level feature map at every location, i.e., $\{[C_\psi^{(i)}(x), y(x)]\}_{i=1}^{M \times M}$. A 1×1 convolutional discriminator is then used to score the (feature map, feature vector) pair,

$$T_{\psi, \omega}^{(i)}(x, y(x)) = D_\omega([C_\psi^{(i)}(x), y(x)]) \quad (6)$$

Fake samples are generated by combining feature vectors with feature maps coming from different images, x' ,

$$T_{\psi, \omega}^{(i)}(x', y(x)) = D_\omega([C_\psi^{(i)}(x'), y(x)]). \quad (7)$$

We then maximize the average JSD mutual information, as above:

$$\begin{aligned} (\hat{\omega}, \hat{\psi})_L &= \arg \max_{\omega, \psi} \frac{1}{M^2} \sum_{i=1}^{M^2} \hat{\mathcal{I}}_{\omega, \psi}^{(\text{JSD})}(X^{(i)}; E_\psi(X)) \\ &= \arg \max_{\omega, \psi} \frac{1}{M^2} \sum_{i=1}^{M^2} \mathbb{E}_{\mathbb{J}_{\psi, \mathbb{P}}}[-\text{sp}(-T_{\psi, \omega}^{(i)}(x, y(x)))] - \mathbb{E}_{\mathbb{M}_{\psi, \mathbb{P}}}[\text{sp}(T_{\psi, \omega}^{(i)}(x', y(x)))]. \end{aligned} \quad (8)$$

As an implementation detail that reduces computational overhead, it is reasonable simply to shuffle the feature map at each location across a batch independently when generating “fake” examples. This introduces a non-zero probability of mixing some “real” samples with our “fakes”, but we found that this did not noticeably affect results.

3.4 Matching representations to a prior distribution

Information content is only one desirable property of a representation; depending on the application, good representations can be compact (Gretton et al., 2012), independent (Hyvärinen & Oja, 2000; Hinton, 2002; Dinh et al., 2014), disentangled (Schmidhuber, 1992; Rifai et al., 2012; Bengio et al., 2013; Chen et al., 2018; Gonzalez-Garcia et al., 2018), or independently controllable (Thomas et al., 2017). As an example, independence is important in many neuroscience studies (Calhoun et al., 2008; Allen et al., 2012; Damaraju et al., 2014), where phrasing medical imaging data as a blind source separation (Bell & Sejnowski, 1995) is instrumental to interpretable analyses. In addition, interpolation is an important tool in analyzing interpretability and generalizability, which can be the result of building representations that are compact or are composed of independently controllable factors. As another example, in model-based RL, it may be desirable to have a representation composed of relatively independent components to simplify planning in representation space.

Our approach to imposing structure onto learned representations is illustrated in Figure 4, where we implicitly train the encoder so that the push-forward distribution, $\mathbb{U}_{\psi, \mathbb{P}}$, matches a prior, \mathbb{V} . As we do not have a density for $\mathbb{U}_{\psi, \mathbb{P}}$, we can achieve this by training a discriminator, $D_\phi : \mathcal{Y} \rightarrow \mathbb{R}$, to estimate the divergence, $\mathcal{D}(\mathbb{V} || \mathbb{U}_{\psi, \mathbb{P}})$, and then training the encoder to minimize this implicitly through back-propagation, i.e.,

$$(\hat{\omega}, \hat{\psi})_P = \arg \min_{\psi} \arg \max_{\phi} \hat{\mathcal{D}}_\phi(\mathbb{V} || \mathbb{U}_{\psi, \mathbb{P}}) = \mathbb{E}_{\mathbb{V}}[\log D_\phi(y)] + \mathbb{E}_{\mathbb{P}}[\log(1 - D_\phi(E_\psi(x)))] \quad (9)$$

This approach is very similar to what is done in adversarial autoencoders (AAE, Makhzani et al., 2015) and bi-directional adversarial models (Dumoulin et al., 2016; Donahue et al., 2016), but without a generator. It is also similar to noise as targets (Bojanowski & Joulin, 2017), defining the target distribution implicitly rather than from a noise distribution sampled *a priori*.

All three objectives – global and local mutual information maximization and prior matching – can be used together, and doing so we arrive at our complete objective for deep INFOMAX (DIM):

$$\arg \max_{\omega_1, \omega_2, \psi} (\alpha \hat{\mathcal{I}}_{\omega_1, \psi}(X; E_\psi(X)) + \frac{\beta}{M^2} \sum_{i=1}^{M^2} \hat{\mathcal{I}}_{\omega_2, \psi}(X^{(i)}; E_\psi(X))) + \arg \min_{\psi} \arg \max_{\phi} \gamma \hat{\mathcal{D}}_\phi(\mathbb{V} || \mathbb{U}_{\psi, \mathbb{P}}), \quad (10)$$

where ω_1 and ω_2 are the discriminator parameters for the global and local objectives, respectively, and α , β , and γ are hyperparameters. We will show below that choices in these hyperparameters affect the learned representations in meaningful ways. As an interesting aside, we also show in the Appendix that this prior matching can be used to train a high-quality generator of image data.

4 Experiments

4.1 Experimental setup

Datasets We test Deep INFOMAX (DIM) on four imaging datasets to evaluate its representational properties:

- CIFAR10 and CIFAR100 (Krizhevsky & Hinton, 2009): two small-scale labeled datasets composed of 32×32 images with 10 and 100 classes respectively.
- Tiny ImageNet: A reduced version of ImageNet (Krizhevsky & Hinton, 2009) images scaled down to 64×64 with a total of 200 classes.
- STL-10 (Coates et al., 2011): a dataset derived from ImageNet composed of 96×96 images with a mixture of 100000 unlabeled training examples and 500 labeled examples per class. We use *data augmentation* with this dataset, taking random 64×64 crops and flipping horizontally during unsupervised learning.

Encoder We used a network similar to a deep convolutional GAN (DCGAN, Radford et al., 2015) discriminator for CIFAR10 and CIFAR100, and for all other datasets we used an Alexnet (Krizhevsky et al., 2012) architecture similar to that found in Donahue et al. (2016). ReLU activations and batch norm (Ioffe & Szegedy, 2015) were used on every hidden layer. For all experiments, the output of all encoders was a 64 dimensional vector.

Discriminators The discriminator for MINE was a fully-connected network for the global objective and a 1×1 convolutional network for the local objective, each with two 512-unit hidden layers. The discriminator used to match the prior in DIM was a fully-connected network with two hidden layers of 1000 and 200 units. There was no batch norm for the discriminators.

Other implementation details DIM with the local objective (Equation 8) can be performed on any level feature map of the convolutional encoder. For classification tasks, we found for Alexnet architectures that using the third feature map in the convolutional network generally worked best, while with the DCGAN architecture, it was the second. In addition, we found that using a nonlinearity (such as sigmoid for a uniform prior) on the output representation, Y , was important for performance on higher-dimensional datasets. Finally, ReLU activations from the feature map needed to be included in the mutual information estimation step for the approach to work well.

4.2 Model comparisons

We compare DIM against the following unsupervised models:

- Variational autoencoders (VAE, Kingma & Welling, 2013),
- Adversarial autoencoders (AAE, Makhzani et al., 2015),
- BiGAN (a.k.a., adversarially learned inference with a deterministic encoder: Donahue et al., 2016; Dumoulin et al., 2016),
- Noise as targets (NAT, Bojanowski & Joulin, 2017).

For generative models, we used a similar setup as that found in Donahue et al. (2016) and Bojanowski & Joulin (2017) for the generators / decoders, where we used a generator from DCGAN in all experiments. Batch norm and ReLU activations were used in all experiments.

All models were trained using Adam with a learning rate of 1×10^{-4} for 1000 epochs for CIFAR10 and CIFAR100 and for 200 epochs for all other datasets.

4.3 How do we evaluate the quality of a representation?

Evaluation of representations is not well-understood and often relies on various proxies. Linear separability of known classes has been used as a proxy for disentanglement and mutual information between representations and class labels. Unfortunately, these approaches do not illuminate if the representation has high mutual information with the class labels when the representation is not disentangled. Other works (Bojanowski & Joulin, 2017; Donahue et al., 2016) have looked at *transfer learning* classification tasks, by freezing the weights of the encoder and training a small fully-connected neural network classifier using the representation as input. Others still have also more directly measured the mutual information between the labels and the representation (Rifai et al., 2012; Chen et al., 2018), which can also reveal the representation’s amount of entanglement.

Class labels have limited use in evaluating representations, as we are often interested in information encoded in the representation that is unknown to us. Implicit models give us two new metrics for evaluating representation quality. First, we can use mutual information neural estimation (MINE, Belghazi et al., 2018) to more directly measure the mutual information between the input and output of the encoder.

Next, we can measure the independence of the representation itself also using a discriminator. Given a batch of representations, we can generate an independent version of these representations by randomly shuffling each factor independently across the batch axis. The new representation is then composed of independent factors. This is similar to an approach that used independent shuffling to learn a maximally independent representation for sequential data (Brakel & Bengio, 2017). We can train a discriminator to measure the KL-divergence between the representation distribution (joint distribution of the factors) and the shuffled representation distribution (product of the marginals), using the Donsker-Varadhan representation as with MINE. The higher the KL divergence, the more dependent the factors, and we call this evaluation method *Neural Dependency Measure* (NDM).

To summarize, we use the following metrics for evaluating representations. For each of these, the encoder is held fixed unless noted otherwise:

Table 1: Classification accuracy (top 1) results on CIFAR10 and CIFAR100. DIM(L) (i.e., with the local-only objective) outperforms all other unsupervised methods presented by a wide margin. In addition, DIM(L) approaches or even surpasses a fully-supervised classifier with similar architecture. DIM with the global-only objective is competitive with some models across tasks, but falls short when compared to generative models and DIM(L) on CIFAR100. Fully-supervised classification results are provided for comparison.

Model	CIFAR10			CIFAR100		
	conv	fc (1024)	Y(64)	conv	fc (1024)	Y(64)
Fully supervised	75.39			42.27		
VAE	60.71	60.54	54.61	37.21	34.05	24.22
AAE	59.44	57.19	52.81	36.22	33.38	23.25
BiGAN	62.57	62.74	52.54	37.59	33.34	21.49
NAT	56.19	51.29	31.16	29.18	24.57	9.72
DIM(G)	52.2	52.84	43.17	27.68	24.35	19.98
DIM(L)	70.1	70.21	63.97	48.46	46.09	36.51

- **Linear classification** using a support vector machine. This is simultaneously a proxy for mutual information of the representation with linear separability.
- **Non-linear classification** using a single hidden layer neural network (200 units) with dropout. This is a proxy on mutual information of the representation with the labels separate from linear separability as measured with the SVM above.
- **Semi-supervised learning**, that is, fine-tuning the complete encoder by adding a small neural network on top of the last convolutional layer (matching architectures with a standard classifier) for further evaluation on a semi-supervised task (STL-10, here).
- **MS-SSIM** (Wang et al., 2003), using a decoder trained on the L_2 reconstruction loss. This is a proxy for the total mutual information between the input and the representation and can indicate the amount of encoded pixel-level information.
- **Mutual information neural estimate (MINE)**, $\hat{I}_\rho(X, Y)$, between the input, X , and the output representation, Y , by training a discriminator with parameters ρ to maximize the DV representation of the KL-divergence.
- **Neural dependency measure (NDM)** using a second discriminator that measures the KL between Y and a batch-wise shuffled version of Y making the different dimensions be independent.

For the neural network classification evaluation above, we performed experiments on all datasets, while for other measures we looked at only CIFAR10. For both the neural and SVM classification measures, we built separate classifiers on the high-level vector representation (Y), the output of the previous fully-connected layer (fc) and the last convolutional layer (conv). Model selection for the classifiers was done by averaging the last 100 epochs of optimization, and the dropout rate was set uniformly to alleviate over-fitting on the test set across all models.

4.4 Representation learning comparison across models

In the following experiments, DIM(G) refers to DIM with a global-only objective ($\alpha = 1, \beta = 0, \gamma = 1$) and DIM(L) refers to DIM with a local-only objective ($\alpha = 0, \beta = 1, \gamma = 0.1$), the latter chosen from the results of an ablation study presented below. For the prior, we chose a compact uniform distribution on $[0, 1]^{64}$, which worked better in practice than other priors, such as Gaussian, unit ball, or unit sphere.

Classification comparisons Our classification results can be found in Tables 1 and 2. In general, DIM with the local objective, DIM(L), outperformed all models presented here by a large margin on all datasets except STL-10, regardless of which layer the representation was drawn from. DIIM(L) performs nearly as well as or outperforms a fully-supervised classifier without fine-tuning, which indicates that the representations are nearly as good or better than the raw pixels given the model constraints.

Table 2: Classification accuracy (top 1) results on Tiny ImageNet and STL-10. For Tiny ImageNet, DIM with the local objective outperforms all other models presented by a large margin, and approaches accuracy of a fully-supervised classifier with similar with the Alexnet architecture used here.

	Tiny ImageNet			STL-10 (random crop pretraining)			
	conv	fc (4096)	$Y(64)$	conv	fc (4096)	$Y(64)$	SS
Fully supervised	36.60			68.7			
VAE	18.63	16.88	11.93	58.27	56.72	46.47	68.65
AAE	18.04	17.27	11.49	59.54	54.47	43.89	64.15
BiGAN	24.38	20.21	13.06	71.53	67.18	58.48	74.77
NAT	13.70	11.62	1.20	64.32	61.43	48.84	70.75
DIM(G)	11.32	6.34	4.95	42.03	30.82	28.09	51.36
DIM(L)	33.8	34.5	30.7	71.82	67.22	61.61	75.62

Table 3: Extended comparisons on CIFAR10. Linear classification results using SVM are over five runs. MS-SSIM is estimated by training a separate decoder using the fixed representation as input and minimizing the L_2 loss with the original input. Mutual information estimates were done using MINE and the neural dependence measure (NDM) were trained using a discriminator between unshuffled and shuffled representations. NDM measures for DIM are the measures with the sigmoid function applied at estimation and without in parentheses.

Model	Proxies				Neural Estimators	
	SVM (conv)	SVM (fc)	SVM (Y)	MS-SSIM	$\hat{I}_\rho(X, Y)$	NDM
VAE	53.83 ± 0.62	42.14 ± 3.69	39.59 ± 0.01	0.72	93.02	1.62
AAE	55.22 ± 0.06	43.34 ± 1.10	37.76 ± 0.18	0.67	87.48	0.03
BiGAN	56.40 ± 1.12	38.42 ± 6.86	44.90 ± 0.13	0.46	37.69	24.49
NAT	48.62 ± 0.02	42.63 ± 3.69	39.59 ± 0.01	0.29	6.04	0.02
DIM(G)	46.8 ± 2.29	28.79 ± 7.29	29.08 ± 0.24	0.49	49.63	0.35(9.96)
DIM(L+G)	57.55 ± 1.442	45.56 ± 4.18	18.63 ± 4.79	0.53	101.65	0.5(22.89)
DIM(L)	63.25 ± 0.86	54.06 ± 3.6	49.62 ± 0.3	0.37	45.09	0.18(9.18)

On STL-10, we found that all models performed better without random cropping (using re-scaling instead) during classification evaluation, though we found data augmentation was important for pretraining. Because of this we used the same decaying learning rate schedule to avoid over-fitting with the same model selection as previous experiments. DIM(L) outperformed all models on all tasks by a large margin, except with BiGAN (which on some tasks it only outperforms by an insignificant amount). This supports our hypothesis that our local DIM objective is suitable for extracting class information.

Extended comparisons Tables 3 shows results on linear separability, reconstruction (MS-SSIM), mutual information, and independence (NDM) with the CIFAR10 dataset. For linear classifier results (SVC), we trained five support vector machines with a simple hinge loss for each model, averaging the test accuracy. For MINE, we used a decaying learning rate schedule, which helped reduce variance in estimates and provided faster convergence.

The MS-SSIM measure of reconstruction correlated well with the mutual information estimate provided by MINE, indicating that these models encoded pixel-wise information well. As our prior matching was done using a sigmoid function on the representation, Y , we measured NDM with and without (in parentheses) this nonlinearity. Overall, all models showed much lower dependence than BiGAN, indicating the marginal of the encoder output is not matching to the generator’s spherical Gaussian input prior. For mutual information, reconstruction-based models like VAE and AAE have high scores, and we found that combining local and global objectives had very high scores ($\alpha = 0.5$, $\beta = 0.1$ is presented here as DIM(L+G), see ablation study below for details). NAT showed the lowest mutual information and reconstruction measures, indicating that information about the pixels is not readily encoded by NAT. We note here as well that classification performance usually anti-correlates with reconstruction performance in this setting, with some exceptions.

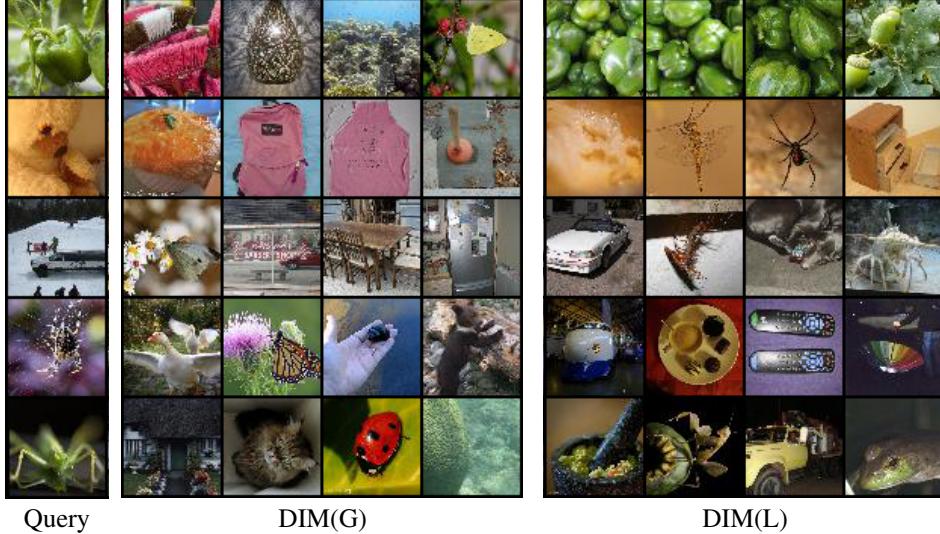


Figure 5: Nearest-neighbor using the L_1 distance on the encoded Tiny ImageNet images, with $\text{DIM}(\text{G})$ and $\text{DIM}(\text{L})$. The images on the far left are randomly-selected reference images (query) from the training set and the four images their nearest-neighbor from the test set as measured in the representation, sorted by proximity. The nearest neighbors from $\text{DIM}(\text{L})$ are much more interpretable than those with the purely global objective.

4.5 Nearest-neighbor analysis

In order to better understand the metric structure of DIM’s representations, we did a nearest-neighbor analysis, randomly choosing a sample from each class in the test set, ordering the test set in terms of L_1 distance in the representation space (to reflect the uniform prior), then selecting the four with the lowest distance. Our results in Figure 5 show that DIM with a local-only objective, $\text{DIM}(\text{L})$, learns a representation with a much more interpretable structure across the image. However, our result potentially highlights an issue with using only consistent information across patches, as many of the nearest neighbors share patterns (colors, shapes, texture) but not class.

4.6 Ablation studies

To better understand the effects of hyperparameters α , β , and γ on the representational characteristics of the encoder, we performed several ablation studies. These illuminate the relative importance of global versus local mutual information objectives as well as the role of the prior.

Local versus global mutual information maximization The results of our ablation study for DIM on CIFAR10 are presented in Figure 6. In general, good classification performance is highly dependent on the local term, β , while good reconstruction is highly dependent on the global term, α . However, a small amount of α helps in classification accuracy and a small about of β improves reconstruction. For mutual information, we found that having a combination of α and β yielded higher MINE estimates.

The effect of the prior We found including the prior term, γ , was absolutely necessary for ensuring low dependence between components of the high-level representation, Y , as measured by NDM. In addition, a small amount of the prior term helps improve classification results when used with the local term, β . This may be because the additional constraints imposed on the representation help to encourage the local term to focus on consistent, rather than trivial, information.

5 Conclusion

In this work, we introduced Deep INFOMAX (DIM), a new method for learning unsupervised representations by maximizing mutual information. DIM allows for representations that contain

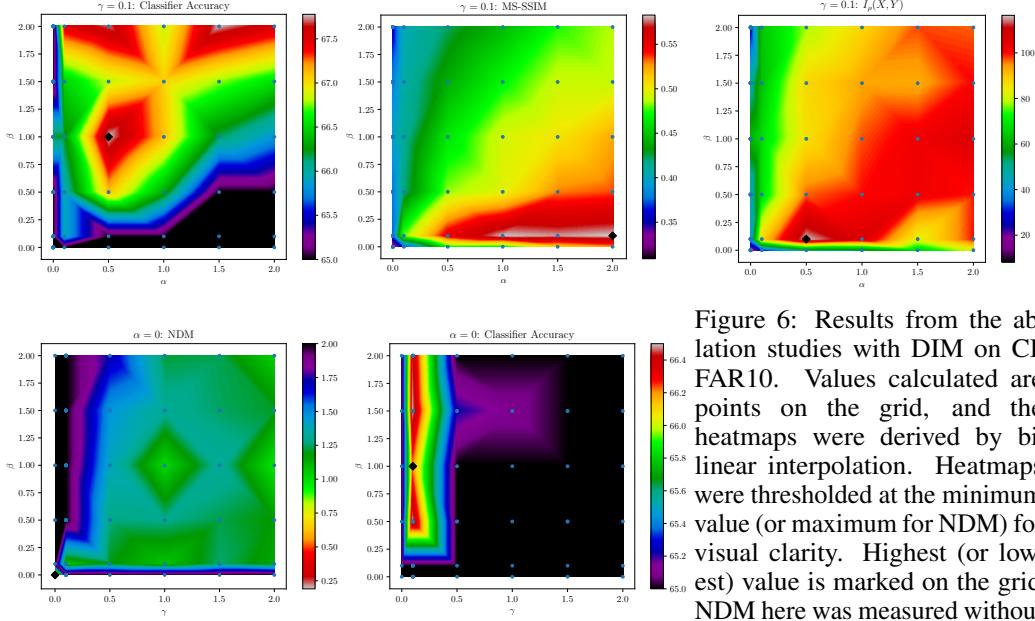


Figure 6: Results from the ablation studies with DIM on CIFAR10. Values calculated are points on the grid, and the heatmaps were derived by bilinear interpolation. Heatmaps were thresholded at the minimum value (or maximum for NDM) for visual clarity. Highest (or lowest) value is marked on the grid. NDM here was measured without the sigmoid function.

locally-consistent information across structural “locations” (e.g., patches in an image). This provides a straightforward and flexible way to learn representations that perform well on a variety of tasks. We believe that this is an important direction in learning “good” and more principled representations which will provide benefit to future research into artificial intelligence.

6 Acknowledgements

RDH received partial support from IVADO, NIH grants 2R01EB005846, P20GM103472, P30GM122734, and R01EB020407, and NSF grant 1539067. AF received partial support from NIH grant R01EB020407. We would also like to thank Philip Bachman (MSR), Geoff Gordon (MSR), Ishmael Belghazi (MILA), Marc Bellemare (Google Brain), Simon Sebbagh, and Aaron Courville (MILA) for their useful input at various points through the course of this research.

References

- Allen, Elena A, Erhardt, Erik B, Wei, Yonghua, Eichele, Tom, and Calhoun, Vince D. Capturing inter-subject variability with group independent component analysis of fMRI data: a simulation study. *Neuroimage*, 59(4):4141–4159, 2012.
- Almeida, Luís B. Linear and nonlinear ica based on mutual information. *The Journal of Machine Learning Research*, 4:1297–1318, 2003.
- Arjovsky, Martin and Bottou, Léon. Towards principled methods for training generative adversarial networks. In *International Conference on Learning Representations*, 2017.
- Becker, Suzanna. *An information-theoretic unsupervised learning algorithm for neural networks*. University of Toronto, 1992.
- Belghazi, Ishmael, Rajeswar, Sai, Baratin, Aristide, Hjelm, R Devon, and Courville, Aaron. Mine: Mutual information neural estimation. *ICML*, 2018.
- Bell, Anthony J and Sejnowski, Terrence J. An information-maximization approach to blind separation and blind deconvolution. *Neural computation*, 7(6):1129–1159, 1995.
- Bengio, Yoshua. Deep generative models for speech and images, 2017. CBMM Workshop on SPEECH REPRESENTATION, PERCEPTION AND RECOGNITION, MIT.

- Bengio, Yoshua, Courville, Aaron, and Vincent, Pascal. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 35(8):1798–1828, 2013.
- Bojanowski, Piotr and Joulin, Armand. Unsupervised learning by predicting noise. *arXiv preprint arXiv:1704.05310*, 2017.
- Bottou, Leon, Arjovsky, Martin, Lopez-Paz, David, and Oquab, Maxime. Geometrical insights for implicit generative modeling. *arXiv preprint arXiv:1712.07822*, 2017.
- Brakel, Philemon and Bengio, Yoshua. Learning independent features with adversarial nets for non-linear ica. *arXiv preprint arXiv:1710.05050*, 2017.
- Bubic, Andreja, von Cramon, D. Yves, and Schubotz, Ricarda I. Prediction, cognition and the brain. *Frontiers in Human Neuroscience*, 4(25), 2010.
- Calhoun, Vince D., Kiehl, Kent A., and Pearson, Godfrey D. Modulation of temporally coherent brain networks estimated using ICA at rest and during cognitive tasks. *Human Brain Mapping*, 29 (7):828–838, 2008.
- Chang, Jianlong, Wang, Lingfeng, Meng, Gaofeng, Xiang, Shiming, and Pan, Chunhong. Deep adaptive image clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5879–5887, 2017.
- Chen, Tian Qi, Li, Xuechen, Grosse, Roger, and Duvenaud, David. Isolating sources of disentanglement in variational autoencoders. *arXiv preprint arXiv:1802.04942*, 2018.
- Chen, Xi, Duan, Yan, Houthooft, Rein, Schulman, John, Sutskever, Ilya, and Abbeel, Pieter. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pp. 2172–2180, 2016.
- Coates, Adam, Ng, Andrew, and Lee, Honglak. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 215–223, 2011.
- Damaraju, Eswar, Allen, Elena A, Belger, Aysenil, Ford, JM, McEwen, S, Mathalon, DH, Mueller, BA, Pearson, GD, Potkin, SG, Preda, A, et al. Dynamic functional connectivity analysis reveals transient states of dysconnectivity in schizophrenia. *NeuroImage: Clinical*, 5:298–308, 2014.
- Dinh, Laurent, Krueger, David, and Bengio, Yoshua. Nice: non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.
- Dinh, Laurent, Sohl-Dickstein, Jascha, and Bengio, Samy. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.
- Donahue, Jeff, Krähenbühl, Philipp, and Darrell, Trevor. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.
- Donsker, M.D and Varadhan, S.R.S. Asymptotic evaluation of certain markov process expectations for large time, iv. *Communications on Pure and Applied Mathematics*, 36(2):183?212, 1983.
- Dumoulin, Vincent, Belghazi, Ishmael, Poole, Ben, Lamb, Alex, Arjovsky, Martin, Mastropietro, Olivier, and Courville, Aaron. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016.
- Gonzalez-Garcia, Abel, van de Weijer, Joost, and Bengio, Yoshua. Image-to-image translation for cross-domain disentanglement. *arXiv preprint arXiv:1805.09730*, 2018.
- Goodfellow, Ian, Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville, Aaron, and Bengio, Yoshua. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pp. 2672–2680, 2014.
- Gretton, Arthur, Borgwardt, Karsten M, Rasch, Malte J, Schölkopf, Bernhard, and Smola, Alexander. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773, 2012.

- Gulrajani, Ishaan, Ahmed, Faruk, Arjovsky, Martin, Dumoulin, Vincent, and Courville, Aaron. Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028*, 2017.
- He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
- Hinton, Geoffrey E. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.
- Hjelm, R Devon, Jacob, Athul Paul, Che, Tong, Trischler, Adam, Cho, Kyunghyun, and Bengio, Yoshua. Boundary-seeking generative adversarial networks. In *International Conference on Learning Representations*, 2018.
- Hyvärinen, Aapo and Oja, Erkki. Independent component analysis: algorithms and applications. *Neural networks*, 13(4):411–430, 2000.
- Hyvärinen, Aapo and Pajunen, Petteri. Nonlinear independent component analysis: Existence and uniqueness results. *Neural Networks*, 12(3):429–439, 1999.
- Ioffe, Sergey and Szegedy, Christian. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.
- Ji, Xu, Henriques, João F, and Vedaldi, Andrea. Invariant information distillation for unsupervised image segmentation and clustering. *arXiv preprint arXiv:1807.06653*, 2018.
- Kingma, Diederik and Welling, Max. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- Kingma, Diederik P, Mohamed, Shakir, Rezende, Danilo Jimenez, and Welling, Max. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pp. 3581–3589, 2014.
- Kohonen, Teuvo. The self-organizing map. *Neurocomputing*, 21(1-3):1–6, 1998.
- Krizhevsky, Alex and Hinton, Geoffrey. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.
- Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.
- Kumar, Abhishek, Sattigeri, Prasanna, and Fletcher, Tom. Semi-supervised learning with gans: Manifold invariance with improved inference. In *Advances in Neural Information Processing Systems*, pp. 5540–5550, 2017.
- Linsker, Ralph. Self-organization in a perceptual network. *IEEE Computer*, 21(3):105–117, 1988. doi: 10.1109/2.36. URL <https://doi.org/10.1109/2.36>.
- Liu, Ziwei, Luo, Ping, Wang, Xiaogang, and Tang, Xiaoou. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.
- Makhzani, Alireza, Shlens, Jonathon, Jaitly, Navdeep, Goodfellow, Ian, and Frey, Brendan. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.
- Mescheder, Lars, Geiger, Andreas, and Nowozin, Sebastian. Which training methods for gans do actually converge? In *International Conference on Machine Learning*, pp. 3478–3487, 2018.
- Nowozin, Sebastian, Cseke, Botond, and Tomioka, Ryota. f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in Neural Information Processing Systems*, pp. 271–279, 2016.
- Oord, Aaron van den, Li, Yazhe, and Vinyals, Oriol. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

- Radford, Alec, Metz, Luke, and Chintala, Soumith. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.
- Rezende, Danilo Jimenez, Mohamed, Shakir, Danihelka, Ivo, Gregor, Karol, and Wierstra, Daan. One-shot generalization in deep generative models. *arXiv preprint arXiv:1603.05106*, 2016.
- Rifai, Salah, Vincent, Pascal, Muller, Xavier, Glorot, Xavier, and Bengio, Yoshua. Contractive auto-encoders: Explicit invariance during feature extraction. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pp. 833–840. Omnipress, 2011.
- Rifai, Salah, Bengio, Yoshua, Courville, Aaron, Vincent, Pascal, and Mirza, Mehdi. Disentangling factors of variation for facial expression recognition. In *European Conference on Computer Vision*, pp. 808–822. Springer, 2012.
- Salimans, Tim, Goodfellow, Ian, Zaremba, Wojciech, Cheung, Vicki, Radford, Alec, and Chen, Xi. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pp. 2234–2242, 2016.
- Schmidhuber, Jürgen. Learning factorial codes by predictability minimization. *Neural Computation*, 4(6):863–879, 1992.
- Thomas, Valentin, Pondard, Jules, Bengio, Emmanuel, Sarfati, Marc, Beaudoin, Philippe, Meurs, Marie-Jean, Pineau, Joelle, Precup, Doina, and Bengio, Yoshua. Independently controllable features. *arXiv preprint arXiv:1708.01289*, 2017.
- Vincent, Pascal, Larochelle, Hugo, Bengio, Yoshua, and Manzagol, Pierre-Antoine. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pp. 1096–1103. ACM, 2008.
- Vincent, Pascal, Larochelle, Hugo, Lajoie, Isabelle, Bengio, Yoshua, and Manzagol, Pierre-Antoine. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(Dec):3371–3408, 2010.
- Wang, Zhou, Simoncelli, Eero P, and Bovik, Alan C. Multiscale structural similarity for image quality assessment. In *Signals, Systems and Computers, 2004. Conference Record of the Thirty-Seventh Asilomar Conference on*, volume 2, pp. 1398–1402. Ieee, 2003.
- Wiskott, Laurenz and Sejnowski, Terrence J. Slow feature analysis: Unsupervised learning of invariances. *Neural computation*, 14(4):715–770, 2002.
- Xie, Junyuan, Girshick, Ross, and Farhadi, Ali. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pp. 478–487, 2016.
- Yu, Fisher, Zhang, Yinda, Song, Shuran, Seff, Ari, and Xiao, Jianxiong. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

A Supplementary material

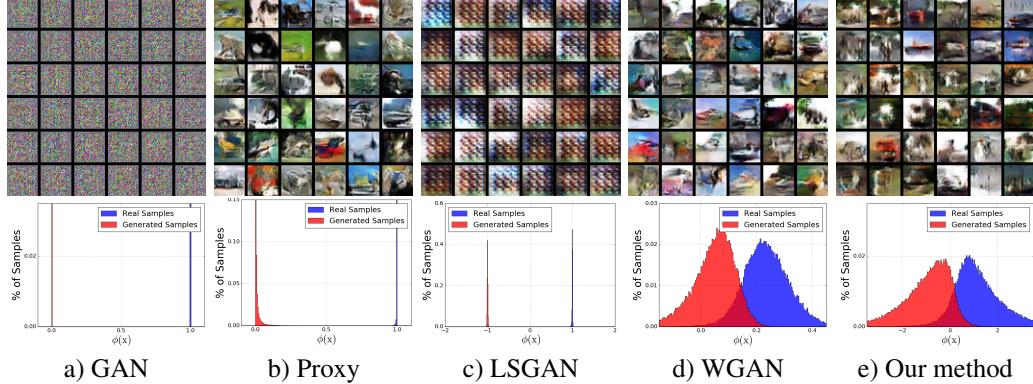


Figure 7: Histograms depicting the discriminator’s unnormalized output distribution for the standard GAN, GAN with $-\log D$ loss, Least Squares GAN, Wasserstein GAN and our proposed method when trained with a 50:1 training ratio.

A.1 Training a generator by matching to a prior implicitly

We show here and in our experiments below that we can use prior objective in DIM (Equation 9) to train a high-quality generator of images by training $\mathbb{U}_{\psi, \mathbb{P}}$ to map to a one-dimensional mixture of two Gaussians implicitly. One component of this mixture will be a target for the push-forward distribution of \mathbb{P} through the encoder while the other will be a target for the push-forward distribution of the generator, \mathbb{Q}_θ , through the same encoder.

Let $G_\theta : \mathcal{Z} \rightarrow \mathcal{X}$ be a generator function, where the input $z \in \mathcal{Z}$ is drawn from a simple prior, $p(z)$ (such as a spherical Gaussian). Let \mathbb{Q}_θ be the generated distribution and \mathbb{P} be the empirical distribution of the training set. Like in GANs, we will pass the samples of the generator or the training data through another function, E_ψ , in order to get gradients to find the parameters, θ . However, unlike GANs, we will not play the minimax game between the generator and this function. Rather E_ψ will be trained to *generate* a mixture of Gaussians conditioned on whether the input sample came from \mathbb{P} or \mathbb{Q}_θ :

$$\mathbb{V}_\mathbb{P} = \mathcal{N}(\mu_P, 1), \quad \mathbb{V}_\mathbb{Q} = \mathcal{N}(\mu_Q, 1), \quad \mathbb{U}_{\psi, \mathbb{P}} = \mathbb{P} \# E_\psi, \quad \mathbb{U}_{\psi, \mathbb{Q}} = \mathbb{Q}_\theta \# E_\psi, \quad (11)$$

where $\mathcal{N}(\mu_P, 1)$ and $\mathcal{N}(\mu_Q, 1)$ are normal distributions with unit variances and means μ_P and μ_Q respectively. In order to find the parameters ψ , we introduce two discriminators, $T_{\phi_P}^P, T_{\phi_Q}^Q : \mathcal{Y} \rightarrow \mathbb{R}$, and use the lower bounds following defined by the JSD f-GAN:

$$(\hat{\psi}, \hat{\phi}_P, \hat{\phi}_Q) = \arg \min_{\psi} \arg \max_{\phi_P, \phi_Q} \mathcal{L}_d(\mathbb{V}_\mathbb{P}, \mathbb{U}_{\psi, \mathbb{P}}, T_{\phi_P}^P) + \mathcal{L}_d(\mathbb{V}_\mathbb{Q}, \mathbb{U}_{\psi, \mathbb{Q}}, T_{\phi_Q}^Q). \quad (12)$$

The generator is trained to move the first-order moment of $\mathbb{E}_{\mathbb{U}_{\psi, \mathbb{Q}}}[y] = \mathbb{E}_{p(z)}[E_\psi(G_\theta(z))]$ to μ_P :

$$\hat{\theta} = \arg \min(\mathbb{E}_{p(z)}[E_\psi(G_\theta(z))] - \mu_P)^2. \quad (13)$$

Some intuition might help understand why this might work. As discussed in Arjovsky & Bottou (2017), if \mathbb{P} and \mathbb{Q}_θ have support on a low-dimensional manifolds on \mathcal{X} , unless they are perfectly aligned, there exists a discriminator that will be able to perfectly distinguish between samples coming from \mathbb{P} and \mathbb{Q}_θ , which means that $\mathbb{U}_{\psi, \mathbb{P}}$ and $\mathbb{U}_{\psi, \mathbb{Q}}$ must also be disjoint.

However, to train the generator, $\mathbb{U}_{\psi, \mathbb{P}}$ and $\mathbb{U}_{\psi, \mathbb{Q}}$ need to share support on \mathcal{Y} in order to ensure stable and non-zero gradients for the generator. Our own experiments by overtraining the discriminator (Figure 7) confirm that lack of overlap between the two modes of the discriminator is symptomatic of poor training.

Suppose we start with the assumption that the encoder targets, $\mathbb{V}_\mathbb{P}$ and $\mathbb{V}_\mathbb{Q}$, should overlap. Unless \mathbb{P} and \mathbb{Q}_θ are perfectly aligned (which according to Arjovsky & Bottou (2017) is almost guaranteed not to happen with natural images), then the discriminator can always accomplish this task by discarding information about \mathbb{P} or \mathbb{Q}_θ . This means that, by choosing the overlap, we fix the strength of the encoder.

Table 4: Generation scores on the Tiny Imagenet dataset for non-saturating GAN with contractive penalty (NS-GAN-CP), Wasserstein GAN with gradient penalty (WGAN-GP) and our method, using an implicit encoder. Our encoder was penalized using CP.

Model	Inception score	FID
Real data	$31.21 \pm .68$	4.03
IE (ours)	$7.41 \pm .10$	55.15
NS-GAN-CP	$8.65 \pm .08$	40.17
WGAN-GP	8.38 ± 0.18	42.30

A.2 Generation experiments and results

For the generator and encoder, we use a ResNet architecture (He et al., 2016) identical to the one found in Gulrajani et al. (2017). We used the contractive penalty (found in Mescheder et al. (2018) but first introduced in contractive autoencoders (Rifai et al., 2011)) on the encoder, gradient clipping on the discriminators, and no regularization on the generator. Batch norm (Ioffe & Szegedy, 2015) was used on the generator, but not on the discriminator. We trained on 64×64 dimensional LSUN (Yu et al., 2015), CelebA (Liu et al., 2015), and Tiny Imagenet dataset.

A.3 Images Generation



Figure 8: Samples of generated results used to get scores in Table 4. For every methods, the sample are generated after 100 epochs and the models are the same. Qualitative results from these three methods show no qualitative difference.

Here, we train a generator mapping to two Gaussian implicitly as described in Section A.1. Our results (Figure 8) show highly realistic images qualitatively competitive to other methods (Gulrajani et al., 2017; Hjelm et al., 2018). In order to quantitatively compare our method to GANs, we trained a non-saturating GAN with contractive penalty (NS-GAN-CP) and WGAN-GP (Gulrajani et al., 2017) with identical architectures and training procedures. Our results (Table 4) show that, while our method did not surpass NS-GAN-CP or WGAN-GP in our experiments, they came reasonably close.