# Mutual Information in Variational Autoencoders

**Felipe N. Ducau**
fnd212@nyu.edu

**Sony Trénous**
jgt275@nyu.edu

## Abstract

Motivated by the work of Chen et al. [2], we analyze the role of mutual information in variational autoencoders. We experimentally study the behavior of this model when mutual information between the latent code and the generated data is explicitly enforced as part of its loss function. Furthermore, we make an attempt to formalize the role of MI in the VAE objective. We give an interpretation of a lower bound to the MI as the reconstruction error of a dual VAE.

## 1 Introduction

The infoGAN paper by Chen et al. [2] introduced an information theoretically motivated regularization to GAN models. In their work, they showed how this could lead to disentangled latent representations. They propose in their paper to extend their work to the regenerative models such as variational autoencoders (VAEs). Previous work [10, 1] has shown that there is an intimate relationship between (non-variational) autoencoders and mutual information, actually showing that the lower bound for the reconstruction error is the same as the lower bound for the mutual information between the latent code and the input data. We extend this analysis to variational autoencoders and look particularly at the relationship between the latent code and the output of the model.

The outline of this paper is as follows: Section 2 introduces the notation and theoretical background of the models we use. We report our implementation and experiments in section 3. Following this we discuss the relationship between mutual information and the VAE objective in section 4. The 5 summarizes our findings.

## 2 Theoretical Background

### 2.1 Variational Autoencoders

We follow the definition of a variational autoencoder according to Kingma and Welling [4], in which we have some dataset $\mathbf{X} = \{x^i\}_1^N$ consisting of $N$ i.i.d samples of some continuous or discrete variable $\mathbf{x}$. We assume that the samples are generated by a random process that involves an unobserved continuous random variable $\mathbf{z}$. The generative process has two steps: (1) a value $\mathbf{z}$ is generated from its prior distribution $p_{\theta^*}(\mathbf{z})$; (2) a value $\mathbf{x}^{(i)}$ is generated according to the conditional distribution $p_{\theta^*}(\mathbf{x}|\mathbf{z})$ as shown in the Bayesian Network of Figure 1. It is assumed that the prior $p_{\theta^*}(\mathbf{z})$ and the likelihood $p_{\theta^*}(\mathbf{x}|\mathbf{z})$ come from parametric families of distributions $p_\theta(\mathbf{z})$, $p_\theta(\mathbf{x}|\mathbf{z})$ which have PDFs that are differentiable almost everywhere w.r.t. both $\theta$ and $\mathbf{z}$.

In this setup, no simplifying assumptions about the marginal or posterior probabilities are made. According to this setup, variational autoencoders (VAEs) where introduced as an algorithm to perform stochastic variational inference and learning that scales to large datasets. To achieve this, a recognition model $q_\phi(\mathbf{z}|\mathbf{x})$ is introduced as an approximation to the intractable posterior $p_\theta(\mathbf{x}|\mathbf{z})$. This model is known as the *probabilistic encoder* in this setting, since given a datapoint $\mathbf{x}$, it produces a distribution over the possible values of the *code* $\mathbf{z}$ from which the datapoint $\mathbf{x}$ could have been generated. Following the same line, the distribution $p(\mathbf{x}|\mathbf{z})$ is known as a probabilistic *decoder*, since given a code $\mathbf{z}$ it produces a probability distribution over the possible corresponding values of $\mathbf{x}$.

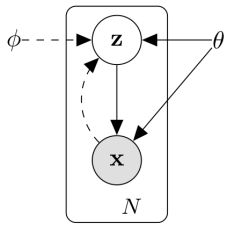


Figure 1: Directed graphical model under consideration. Solid lines show the generative model, while the dashed lines denote the variational approximation $q_\phi(\mathbf{z}|\mathbf{x})$. Figure from [4]

### 2.1.1 Variational Lower Bound

The marginal likelihood of the data if composed of a sum over the marginal likelihoods of individual samples $\log p_\theta(\mathbf{x}^{(1)} \dots, \mathbf{x}^{(N)}) = \sum_{i=1}^{N} \log p_\theta(\mathbf{x}^{(i)})$. Each of the marginal likelihoods can be further written as:

$$\log p_\theta(\mathbf{x}^{(i)}) = \mathbf{D}_{KL}\Big(q_\phi(\mathbf{z}|\mathbf{x}^{(i)}) \;\Big\|\; p_\theta(\mathbf{z}|\mathbf{x}^{(i)})\Big) + \mathcal{L}(\theta, \phi; \mathbf{x}^{(i)}) \tag{1}$$

The first term on the RHS corresponds to the KL divergence between the approximation and the true posterior and the second term is the *variational lower bound* (ELBO) on the marginal likelihood of the datapoint $i$. Since the KL-divergence is non-negative, we can bound the log likelihood of the datapoint $i$ as:

$$\log p_\theta(\mathbf{x}^{(i)}) \geq \mathcal{L}(\theta, \phi; \mathbf{x}^{(i)}) = \mathop{\mathbf{E}}_{q_\phi(\mathbf{z}|\mathbf{x})}\big[-\log q_\phi(\mathbf{z}|\mathbf{x}) + \log p_\theta(\mathbf{x}, \mathbf{z})\big] \tag{2}$$

$$= -\underbrace{\mathbf{D}_{KL}\Big(q_\phi(\mathbf{z}|\mathbf{x}^{(i)}) \;\Big\|\; p_\theta(\mathbf{z})\Big)}_{\text{Regularizer}} + \underbrace{\mathop{\mathbf{E}}_{q_\phi(\mathbf{z}|\mathbf{x}^{(i)})}\Big[\log p_\theta(\mathbf{x}^{(i)}|\mathbf{z})\Big]}_{\text{Reconstruction Term}} \tag{3}$$

. By examining Eq. 2 we see that the first term measures how similar the prior and its variational approximation are, while the second one measures how likely the input of the autoencoder is under the approximation of the posterior. We refer to the former as the *regularizer*, the latter as the *reconstruction term*. If we would not have the KL-divergence term, then we would be in an autoencoder setting where the model is just trying to reconstruct the input as best as possible. The KL-divergence term acts as a regularizer enforcing the prior of the latent space. This allows us to sample new observations after training.

Since we want to maximize the log-likelihood of the data, we are interested in maximizing the variational lower bound, which is equivalent to maximizing Eq. 2. In order to do so, we want to differentiate $\mathcal{L}(\theta, \phi; \mathbf{x}^{(i)})$ w.r.t the variational parameters $\phi$ and the generative parameters $\theta$.

It is often the case that we can compute the KL-divergence in Eq. 2 analytically, so the only term that needs to be estimated is the reconstruction error $\mathop{\mathbf{E}}_{q_\phi(\mathbf{z}|\mathbf{x}^{(i)})}\Big[\log p_\theta(\mathbf{x}^{(i)}|\mathbf{z})\Big]$. Since we have access to both $q_\phi(\mathbf{x}^{(i)}|\mathbf{z})$ and $p_\theta(\mathbf{x}^{(i)}|\mathbf{z})$ we can use them to estimate the expectation by sampling.

To be able to propagate gradients of the loss w.r.t the input, we need to reparametrize the network such that the output is a deterministic function of the input. The *reparametrization trick* as in [4] is used for this purpose. The Monte Carlo estimate of the lower bound is then:

$$\tilde{\mathcal{L}}(\theta, \phi; \mathbf{x}^{(i)}) = -\mathbf{D}_{KL}\Big(q_\phi(\mathbf{z}|\mathbf{x}^{(i)}) \;\Big\|\; p_\theta(\mathbf{z})\Big) + \frac{1}{L}\sum_{l=1}^{L} \log p_\theta(\mathbf{x}^{(i)}|\mathbf{z}^{(i,l)}) \tag{4}$$

where $\mathbf{z}^{(i,l)} = g_\phi(\epsilon^{(i,l)}, X(i))$ and $\epsilon^{(l)} \sim p(\epsilon)$. We define $p(\epsilon)$ to be a normal distribution with zero mean and unit variance.

Given a dataset $\mathbf{X}$ with $N$ datapoints divided into minibatches of size $M$, we can estimate our lower bound as

$$\mathcal{L}(\theta, \phi; \mathbf{x}) \simeq \frac{N}{M} \sum_{i=1}^{M} \tilde{\mathcal{L}}(\theta, \phi; \mathbf{x}^{(i)}) \tag{5}$$

As observed by Kingma and Welling [4], if the minibatches are sufficiently large, the number of samples $L$ per example can be set to one.

### 2.1.2 Mutual Information

Mutual information (MI) is a symmetric quantity related to the change in uncertainty in one random variable when conditioning on another. For variables $\mathbf{x}, \mathbf{z}$, it is defined as

$$\mathbf{I}(\mathbf{x}, \mathbf{z}) = \mathbf{H}(\mathbf{x}) - \mathbf{H}(\mathbf{x}|\mathbf{z}) = \mathbf{H}(\mathbf{z}) - \mathbf{H}(\mathbf{z}|\mathbf{x}) = \mathbf{I}(\mathbf{z}, \mathbf{x})$$

In the discrete case, this quantity is non-negative and bounded above by the minimum of the entropies of the two random variables. For the continuous case, these terms are defined as:

MUTUAL INFORMATION

$$\mathbf{I}(x, z) = \int_x \int_y p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \, \mathrm{d}y \, \mathrm{d}x$$

DIFFERENTIAL ENTROPY

$$\mathbf{H}(x) = -\int_x p(x) \log p(x) \, \mathrm{d}x$$

CONDITIONAL DIFFERENTIAL ENTROPY

$$\mathbf{H}(x|y) = -\int_x \int_y p(x, y) \log p(x|y) \, \mathrm{d}y \, \mathrm{d}x$$

In the continuous case, the differential entropy can attain negative values [1] and thus, the mutual information becomes only bounded by below. Since for this analysis we will consider mostly continuous distributions, we will often refer to the differential entropy and differential conditional entropy just as entropy and conditional entropy for ease of reading. The mutual information for two continuous random variables, thus can be re-written as:

$$\begin{aligned} \mathbf{I}(\mathbf{x}, \mathbf{z}) &= \mathbf{H}(\mathbf{z}) - \mathbf{H}(\mathbf{z}|\mathbf{x}) && (6) \\ &= -\mathop{\mathbf{E}}_{\mathbf{z} \sim p(\mathbf{z})} \left[ \log p(\mathbf{z}) \right] + \mathop{\mathbf{E}}_{\mathbf{x}, \mathbf{z} \sim p(\mathbf{x}, \mathbf{z})} \left[ \log p(\mathbf{z}|\mathbf{x}) \right] && (7) \end{aligned}$$

We observe from equation 7 that computing the MI of latent representations and observations in a generative model requires knowledge of the posterior $p_\theta(z|x)$ and is thus intractable in the settings that we consider. Instead we will optimize the variational lower bound given by Agakov [1] to enforce mutual information between the latent code $\mathbf{z}$ and the output of the VAE $\mathbf{x}$:

[1] It is easy to see for the case of a uniform distribution with support over $[0, 1/2]$

$$
\begin{aligned}
\mathbf{I}(\mathbf{z}, \mathbf{x}) = \mathbf{H}(\mathbf{z}) - \mathbf{H}(\mathbf{z} \mid \mathbf{x}) &= \mathop{\mathbf{E}}_{\mathbf{x}\sim p_\theta(\mathbf{x})} \left[ \mathop{\mathbf{E}}_{\mathbf{z}\sim p_\theta(\mathbf{z}|\mathbf{x})} \left[ \log p_\theta(\mathbf{z} \mid \mathbf{x}) \frac{q_\phi(\mathbf{z} \mid \mathbf{x})}{q_\phi(\mathbf{z} \mid \mathbf{x})} \right] \right] + \mathbf{H}(\mathbf{z}) \\
&= \mathop{\mathbf{E}}_{\mathbf{x}\sim p_\theta(\mathbf{x})} \left[ \underbrace{\mathbf{D}_{KL}\big(p_\theta(\cdot \mid \mathbf{x}) \,\big\|\, q_\phi(\cdot \mid \mathbf{x})\big)}_{\geq 0} + \mathop{\mathbf{E}}_{\mathbf{z}\sim p_\theta(\mathbf{z}|\mathbf{x})} \big[\log q_\phi(\mathbf{z} \mid \mathbf{x})\big] \right] + \mathbf{H}(\mathbf{z}) \\
&\geq \mathop{\mathbf{E}}_{x\sim p_\theta(\mathbf{x})} \left[ \mathop{\mathbf{E}}_{\mathbf{z}\sim p_\theta(\mathbf{z}|\mathbf{x})} \big[\log q_\phi(\mathbf{z} \mid \mathbf{x})\big] \right] + \mathbf{H}(\mathbf{z}) \\
&= \mathop{\mathbf{E}}_{\mathbf{z}\sim p_\theta(\mathbf{z})} \left[ \mathop{\mathbf{E}}_{x\sim p_\theta(\mathbf{x}|\mathbf{z})} \left[ \mathop{\mathbf{E}}_{\mathbf{z}'\sim p_\theta(\mathbf{z}|\mathbf{x})} \big[\log q_\phi(\mathbf{z}' \mid \mathbf{x})\big] \right] \right] + \mathbf{H}(\mathbf{z}) \\
&= \mathop{\mathbf{E}}_{\mathbf{z}\sim p_\theta(\mathbf{z}), \mathbf{x}\sim p_\theta(\mathbf{x}|\mathbf{z})} \big[\log q_\phi(\mathbf{z} \mid \mathbf{x})\big] + \mathbf{H}(\mathbf{z})
\end{aligned}
$$

Where the last equality comes from the following lemma [2]:

**Lemma 1.** *For random variables X, Z and function f(x, z) under suitable regularity conditions:*

$$
\mathop{\mathbf{E}}_{z\sim Z, x\sim X|z} \big[f(x, z)\big] = \mathop{\mathbf{E}}_{z\sim Z, x\sim X|z, z'\sim Z|x} \big[f(x, z')\big]
$$

Note that in contrast to the GAN setting, the VAE already uses a variational distribution $q_\phi$, which can be recycled to compute the variational lower bound.

Following Chen et al. [2], we consider the latent code as a concatenation of two parts $\mathbf{z}, \mathbf{c}$ and enforce the mutual information only over $\mathbf{c}$ by adding the term $-\lambda I(\mathbf{c}, \mathbf{x})$ to the VAE objective function, where $\lambda$ is a regularization parameter. This can be achieved by computing the log likelihood in the inner expectation only for this part of the latent code and then derive a Monte-Carlo estimate based on minibatches of the expectation $\mathop{\mathbf{E}}_{\mathbf{z}\sim p_\theta(\mathbf{z}), \mathbf{x}\sim p_\theta(\mathbf{x}|\mathbf{z})} \big[\log q_\phi(\mathbf{z}|\mathbf{x})\big]$.

For each minibatch, the infoVAE algorithm does the following:

1. Compute $q_\phi(\mathbf{z}, \mathbf{c}|\mathbf{x})$ and the KL Divergence from the prior $p_\theta(\mathbf{z})$.
2. Generate a sample $\mathbf{z}, \mathbf{c}$ from the approximate posterior $q_\phi$.
3. Compute the conditional $p_\theta(\mathbf{x}|\mathbf{z})$ and incur the reconstruction loss.
4. Resample $\tilde{\mathbf{c}}, \tilde{\mathbf{z}} \sim p_\theta(\mathbf{c}, \mathbf{z})$ from the prior.
5. Recompute the conditional $p_\theta(\mathbf{x}|\tilde{\mathbf{z}}, \tilde{\mathbf{c}})$ and generate a sample $\tilde{\mathbf{x}}$.
6. Recompute the approximate posterior $q_\phi(\mathbf{c}|\tilde{\mathbf{x}})$ and incur the loss for the MI lower bound.

## 3 Experiments

### 3.1 Implementation

We implemented the proposed algorithm by extending a stock VAE from Metzen [8]. The implementation [2] is in Tensorflow using identical hidden layer sizes (but transposed) Feed Forward neural nets to parametrize both the variational distribution $q_\phi$ and the conditional $p_\theta$. We use two hidden layers with 500 units each followed by softplus activation functions. The latent space is modeled as isotropic Gaussians with the standard normal prior and the reconstruction space as Bernoulli variables.

One thing to note about the infoVAE implementation is that we are not adding complexity to the model in terms of the number of parameters to tune, even thought it requires a constant larger number of computations. The only hyperparameter introduced is the proportion of the latent code in which

[2] `https://github.com/fducau/infoVAE`

we want to enforce the mutual information and the regularization parameter $\lambda$. In order to gain intuition about the behavior of the infoVAE procedure we trained generative models of images from the MNIST dataset and compared the results when to the stock VAE.

Throughout the experiments we trained the network with Adam [5] optimization algorithm through backpropagation with a learning rate of $10^{-3}$ and batch size of $100$ samples. The presented results were obtained by setting the regularization parameter $\lambda$.

Figure 2 shows the behavior of the VAE and infoVAE algorithms during training with a latent code of $10$ dimensions enforcing mutual information in $5$. The first thing to notice is that when we use VAE algorithm, there is already a certain amount of mutual information. This observation is further explained in Section 4 and it is an expected behavior.

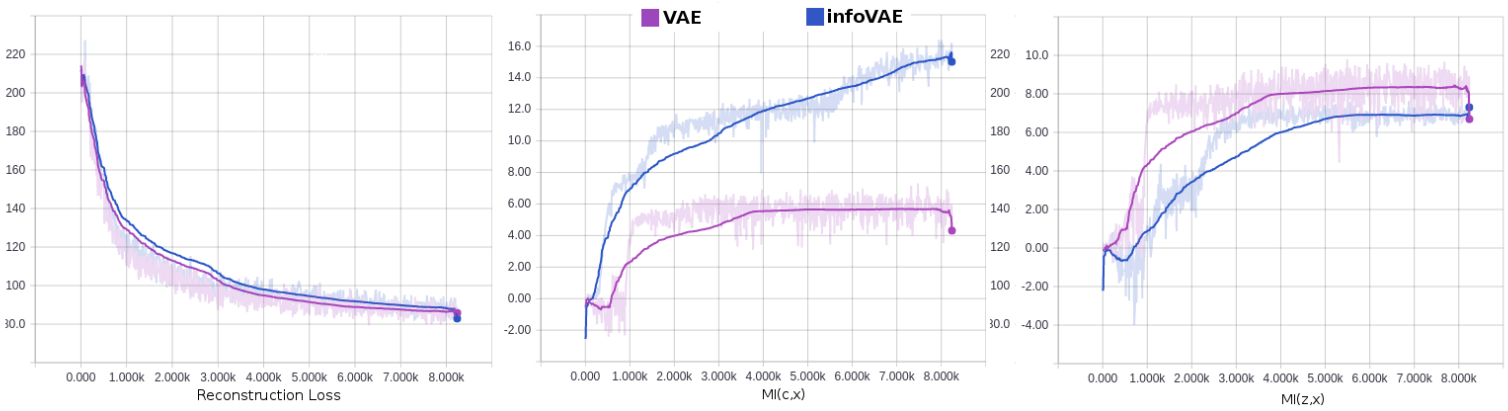


Figure 2: Left: reconstruction loss in training. Center: mutual information between $\mathbf{c}$ and the reconstruction $\mathbf{x}$. Right: mutual information between $\mathbf{z}$ and the reconstruction (10-dimensional latent space, 5 dimensions with MI lower bound enforced) $\mathbf{x}$.

There is also a trade-off between the amount of mutual information encoded in each of the halves of the latent code. As expected we observe an increase in the mutual information $I(\mathbf{c}, \mathbf{x})$ (where is it enforced) while $I(\mathbf{z}, \mathbf{x})$ is reduced. In this case both the reconstruction loss and the KL-divergence term of the ELBO behave similar as in the stock VAE, although in both there is a decrease in the performance. It is worth noticing that the total mutual information encoded in the latent space $[\mathbf{z}, \mathbf{c}]$ is overall higher for the case of infoVAE than when not explicitly enforcing mutual information (VAE).

What the algorithm is doing when the MI is forced just in a part of the latent code is to try to store as as much information as possible in that part of the code. In an attempt to visualize this, we train a network now with $20$ dimensions for the latent code and again impose mutual information in half of the latent representation. From previous experiments we noticed that using $10$ dimensions reaches the best possible performance for this architecture on this dataset, therefore is to expect that infoVAE will try to encode all the information in $\mathbf{c}$. This matches what we observed during training: The MI in $\mathbf{z}$ quickly went to zero, whereas the MI in $\mathbf{c}$ assumed a similar value as the overall MI when training a stock VAE.

Figure 3 shows the output of the model when varying one of the dimensions, $z_i$, in which the MI is not enforced (visualized as a movement from top to bottom) and when varying one of the dimensions $c_i$ in which MI is enforced (from left to right). It is actually the case that all the information is contained in the dimensions of the latent code where the MI is enforced, since changing the values of $z_i$ almost does not change the reconstruction. Even though we are showing just the variation in 2 out of 20 dimensions, the same behavior was observed along the other dimensions. We can see from this experiment the clear link between reconstruction and mutual information – the lower bound of mutual information indicates the salient dimensions of the latent code. This supports our hypothesis that mutual information is already enforced by a VAE.

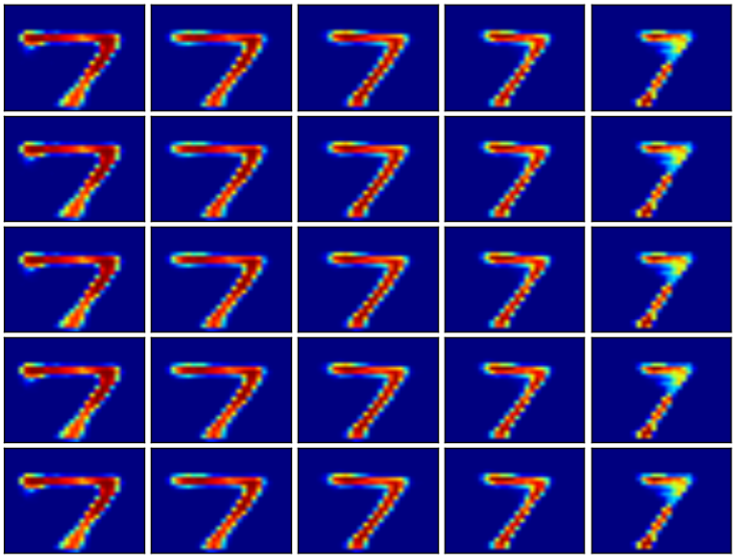

Figure 3: Reconstructions obtained from a model with 20 latent dimensions, where mutual information is enforced on $|\mathbf{c}| = 10$. We vary one dimension of $\mathbf{c}$ (x axis) and one dimension of $\mathbf{z}$ (y axis). Both variations go from $-2.5$ to $2.5$. The model encodes all of the information in the code $\mathbf{c}$ and disregards the code $\mathbf{z}$ (This was observed over all dimensions).

## 4 Discussion

**MI and the VAE objective** If we look at the probability of correctly reconstructing the input of the variational autoencoder, we have

$$
\begin{aligned}
\log \mathop{\mathbf{E}}_{p_\theta(\mathbf{z}|\mathbf{x})} \big[\log p_\theta(\mathbf{x}|\mathbf{z})\big] &= \log \mathop{\mathbf{E}}_{p_\theta(\mathbf{z}|\mathbf{x})} \left[\frac{q_\phi(\mathbf{z}|\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})} p_\theta(\mathbf{x}|\mathbf{z})\right] \\
&= \log \mathop{\mathbf{E}}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[\frac{p_\theta(\mathbf{z}|\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})} p_\theta(\mathbf{x}|\mathbf{z})\right] \\
&\overset{\text{Jensen}}{\geq} \mathop{\mathbf{E}}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[\log \frac{p_\theta(\mathbf{z}|\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})} p_\theta(\mathbf{x}|\mathbf{z})\right] \\
&= \mathop{\mathbf{E}}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[\log \frac{p_\theta(\mathbf{z}|\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})}\right] + \mathop{\mathbf{E}}_{q_\phi(\mathbf{z}|\mathbf{x})} \big[\log p_\theta(\mathbf{x}|\mathbf{z})\big] \\
&= -\mathbf{D}_{KL}\big(q_\phi(\mathbf{z}|\mathbf{x}) \,\big\|\, p_\theta(\mathbf{z}|\mathbf{x})\big) + \mathop{\mathbf{E}}_{q_\phi(\mathbf{z}|\mathbf{x})} \big[\log p_\theta(\mathbf{x}|\mathbf{z})\big]
\end{aligned}
$$

where the last term on the RHS of the equation is the same as the one considered in the variational lower bound in 2.

The KL Divergence in the last line is also the gap between the ELBO and the log likelihood. Maximizing the ELBO is therefore equivalent to maximizing the probability of a correct reconstruction regularized by the prior over the latent space. It has been shown that if the prior distribution is assumed fixed – $\mathbf{H}(\mathbf{z})$ is a constant – maximizing this probability also maximizes the mutual information between latent and output space [10, 1]. We see therefore that the VAE *already enforces* mutual information of the data distribution.

**Relationship between MI lower bound and reconstruction term** The MI lower bound optimized by Chen et al. [2] has an interpretation as the dual of the reconstruction term. It maximizes the

probability of a correct reconstruction of the latent code, while disregarding the observations. If we look at the reconstruction error term of the variational lower bound and the mutual information lower bound, we see that both are intimately related.

$$\text{Reconstruction Term} = \mathop{\mathbf{E}}_{\mathbf{x}\sim p_{\text{data}}(x)}\left[\mathop{\mathbf{E}}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right]\right]$$

$$\text{MI lower bound} = \mathop{\mathbf{E}}_{\mathbf{z}\sim p_\theta(z)}\left[\mathop{\mathbf{E}}_{\mathbf{x}\sim p_\theta(\mathbf{x}|\mathbf{z})}\left[\log q_\phi(\mathbf{z}|\mathbf{x})\right]\right]$$

The reconstruction term is the MI lower bound for a dual variational autencoder where the roles of $\mathbf{x}, \mathbf{z}$ are flipped, and the MI lower bound would be the reconstruction term for it.

**Comparison Infogan and VAE** The InfoGAN model thus combines a dual autoencoder for (part of) its latent representation with a generative-adversarial net. Without this autoencoder, there is no reason that the mapping from code to image should be partially invertible, as the adversarial loss only looks at the likelihood of the generated images. On the other hand, the prior distribution in a GAN is fixed as the samples are generated from it. This is the reason the variational posterior $q_\phi$ does not need to be regularized with respect to the prior. As the VAE does never sample from the prior, omitting the KL divergence term would lead to a posterior that approaches a delta distribution, minimizing uncertainty in the reconstruction.

## 5 Conclusion

We have studied the relationship between mutual information and variational autoencoders. Even though the relationship with plain autoencoders was already formulated, we extended this analysis to probabilistic generative models. We showed that the VAE objective is intimately related to MI maximization. When enforcing MI between some part of the latent code of a variational autoencoder and its output, the VAE tries to encode as much information as possible in that part of the code, to the point of encoding all the information in it. MI can thus be used to identify the most salient parts of the latent space. We also found that we can interpret this mutual information as the reconstruction loss of a *dual* VAE and showed how this can be interpreted in the setting of generative adversarial networks with mutual information.

## References


[1] David Barber Felix Agakov. "The IM algorithm: a variational approach to information maximization". In: *Advances in Neural Information Processing Systems 16: Proceedings of the 2003 Conference*. Vol. 16. MIT Press. 2004, p. 201.

[2] Xi Chen et al. "InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets". In: *CoRR* abs/1606.03657 (2016). URL: `http://arxiv.org/abs/1606.03657`.

[3] I. J. Goodfellow et al. "Generative Adversarial Networks". In: *ArXiv e-prints* (June 2014). arXiv: `1406.2661 [stat.ML]`.

[4] D. P Kingma and M. Welling. "Auto-Encoding Variational Bayes". In: *ArXiv e-prints* (Dec. 2013). arXiv: `1312.6114 [stat.ML]`.

[5] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *CoRR* abs/1412.6980 (2014). URL: `http://arxiv.org/abs/1412.6980`.

[6] T. D. Kulkarni et al. "Deep Convolutional Inverse Graphics Network". In: *ArXiv e-prints* (Mar. 2015). arXiv: `1503.03167 [cs.CV]`.

[7] Alireza Makhzani et al. "Adversarial Autoencoders". In: *CoRR* abs/1511.05644 (2015). URL: `http://arxiv.org/abs/1511.05644`.

[8] Jan Hendrik Metzen. *Variational Autoencoder in TensorFlow*. https://jmetzen.github.io/2015-11-27/vae.html. Blog. 2015.

[9] Jason Tyler Rolfe. "Discrete Variational Autoencoders". In: *arXiv preprint arXiv:1609.02200* (2016).

[10] P. Vincent et al. *Extracting and composing robust features with denoising autoencoders*. Tech. rep. Université de Montréal, dept. IRO, 2008.