InfoNCE is a variational autoencoder

Anonymous authors
Paper under double-blind review

Abstract

Unsupervised learning typically involves learning a full generative model of the inputs, and goes back at least to the Boltzmann (Ackley et al. 1985) and Helmholtz machines (Dayan et al. 1995). More recently, it was noted that we can get good representations of unlabelled datapoints without the need to learn a full generative model, and this gave birth to the modern field of self-supervised learning (SSL). We reconcile these critically important families of machine learning method, by showing that modern SSL methods including InfoNCE which maximize mutual information are equivalent to a particular unsupervised learning method, the variational autoencoder. Additionally, our approach resolves mysteries purely in SSL. In particular, recent work (Tschannen et al., 2019) has argued that mutual information objectives can give arbitrarily entangled representations. Instead, they argue that the excellent performance of InfoNCE arises from the use of a simplified, linear mutual information estimator. How can we understand the success of InfoNCE if better mutual information estimators lead to worse representations? Remarkably, under one choice of prior, the VAE objective (i.e. the ELBO) is exactly equal to the mutual information (up to constants). Under an alternative choice of prior, the SSVAE objective is exactly equal to the simplified parametric mutual information estimator used in InfoNCE (up to constants). As such, the SSVAE framework naturally provides a principled justification for using simplified mututal information estimators, because they are equivalent to structured priors.

1 Introduction

A common challenge occuring across machine learning is to extract useful, structured representations from unlabelled data (such as images). This problem is now known as self-supervised learning, and there are two broad approaches: generative and contrastive learning (Liu et al., 2021).

Generative self-supervised learning (also known as unsupervised learning) can be traced back at least to the Boltzmann machine (Ackley et al., 1985) and the Helmholtz machine (Dayan et al., 1995; Hinton et al., 1995). This classical work emphasises two key characteristics of most generative models; first, they should in some sense model the probability density of the data and second they should use latent variables that are ideally interpretable. Modern generative models are exemplified by variational autoencoders (VAEs) (Kingma & Welling, 2013; Rezende et al., 2014). VAEs (like the Helmholtz machine) learn a probabilitistic encoder which maps from the data to a latent representation, and learn a decoder which maps from the latent representation back to the data domain. This highlights perhaps the key issue with VAEs: the need to reconstruct the data, which may be highly complex (e.g. images) (Dorta et al., 2018) which may force the latent space to encode details of the image that are irrelevant for forming a good high-level representation (Chen et al., 2016b).

Contrastive self-supervised learning is an alternative class of methods that learn good representations without needing to reconstruct the data. One common approach is to define a "pretext" classification task (Dosovitskiy et al., 2015; Noroozi & Favaro, 2016; Doersch et al., 2015; Gidaris et al., 2018). For instance, we might take a number of images, rotate them, and then ask the model to determine the rotation applied (Gidaris et al., 2018). The rotation can be identified by looking at the objects in the image (e.g. grass is typically on the bottom of the image, while birds are nearer the top), and thus a representation useful for determining the orientation may also extract useful information for other high-level tasks. We are interested

in an alternative class of objectives known as InfoNCE (NCE standing for noise contrastive estimation) (Oord et al., 2018). These methods take two inputs (e.g. two different patches from the same underlying image), encode them to form two latent representations, and maximize the mutual information between them. As the shared information should concern high-level properties such as objects, but not low-level details of each patch, this should again extract a useful representation.

InfoNCE was thought to learn good representations by (approximately) maximizing mutual information (Oord et al., 2018). However, recent work has argued that principle behind the success of InfoNCE cannot be maximizing mutual information, because the mutual information is invariant under arbitrary invertible transformations, so maximizing mutual information could give arbitrarily entangled representations (Tschannen et al., 2019). Instead, they argue that InfoNCE learns good representations because it uses a highly approximate, linear mutual information estimator (Oord et al., 2018). This is highly problematic: better mutual information estimators give worse representations (Tschannen et al., 2019), so InfoNCE's success with a highly approximate estimator cannot be due to maximizing mutual information. So what is InfoNCE doing? And how can the success of its simplified mutual information estimator be understood?

Here, we develop a new family of self-supervised variational autoencoders (SSVAEs). Critically, the SSVAE objective (the ELBO) is exactly equal to the mutual information under one choice of prior, and exactly equal to the parametric mutual information estimator used in InfoNCE under an alternative prior (up to constants). This unifies contrastive and generative SSL, and provides a principled justification for the use of simplified parametric mutual information estimators in InfoNCE methods: they are priors, and of course we are free to choose priors with e.g. linear structure in the latent space, if we believe those describe the true latent dynamics. This suggests the success of InfoNCE is because of a choice of prior that induces simple latent representations. In contrast, if the success of InfoNCE arose from mutual information maximization, then InfoNCE would improve as we used more complex and accurate mutual information estimators. But of course, the performance of InfoNCE actually gets worse with more accurate mutual information estimators (Tschannen et al., 2019).

2 Background

2.1 Variational Autoencoders

Usually in a variational autoencoder (Kingma & Welling, 2013; Rezende et al., 2014), we have observed data, x, and latents, z, and we specify a prior, $\mathrm{P}(z)$, a likelihood, $\mathrm{P}(x|z)$, and an approximate posterior, $\mathrm{Q}(z|x)$. We then jointly optimize parameters of the prior, likelihood, and approximate posterior using the ELBO,

$$\log \mathrm{P}(x) \geq \mathcal{L} = \mathrm{E}_{\mathrm{Q}(z|x)}\left[\log \frac{\mathrm{P}(x|z)\,\mathrm{P}(z)}{\mathrm{Q}(z|x)}\right], \tag{1}$$

which bounds the model evidence or marginal likelihood, $\mathrm{P}(x)$ (as can be shown using Jensen's inequality). The approximate posterior, $\mathrm{Q}(z|x)$, is often known as the encoder as it maps from data to latents, while the likelihood, $\mathrm{P}(x|z)$, is often known as the decoder, as it maps from latents back to the data domain.

2.2 InfoNCE

In InfoNCE (Oord et al., 2018), there are two data items, x and x'. Oord et al. (2018) initially describes a time-series setting where x was a context giving the recent history of past data and x' was data for the next time step. But Oord et al. (2018) also consider other contexts where x and x' are different augmentations or patches of the same underlying image. We then form latent representations, z and z' by passing x and x' through neural network encoders, $z = g(x)$ and $z' = g'(x')$. Note that while we often have $g = g'$, there is no requirement that this be the case. The InfoNCE objective was originally motivated as maximizing the mutual information between latent representations,

$$\mathrm{I}(z; z') = \mathrm{E}_{\mathrm{Q}(z,z')}\left[\log \frac{\mathrm{Q}(z'|z)}{\mathrm{Q}(z')}\right]. \tag{2}$$

Note, we are using Q rather than P for consistency with VAE derivations in the methods. As such, $\mathrm{Q}(z, z')$ denotes the distribution induced by taking true data, (x, x'), and encoding them with neural networks, g and g'. As the mutual information is difficult to estimate, they use a bound based on a classifer that distinguishes the positive sample (i.e. the z' paired with the corresponding z) from negative samples (i.e. z'_j drawn from the marginal distribution and unrelated to z or to the underlying data; see Poole et al., 2019 for further details).

$$\mathrm{I}(z; z') \geq \mathcal{L}_{\mathrm{InfoNCE};N} = \mathrm{E}\left[\log \frac{f(z, z')}{f(z, z') + \sum_{j=1}^{N} f(z, z'_j)}\right] + \log N \tag{3}$$

where the expectation is taken over $\mathrm{Q}(z, z') \prod_j \mathrm{Q}\left(z'_j\right)$, and we use this objective to optimize the parameters of f, g and g'. Taking the limit as N goes to infinity, and assuming an arbitrarily flexible classifier, the bound becomes tight (Oord et al., 2018), and can be written as (Wang & Isola, 2020; Li et al., 2021),

$$\mathrm{I}(z; z') = \mathcal{L}_{\mathrm{InfoNCE}} = \mathrm{E}_{\mathrm{Q}(z,z')}\left[\log f(z, z')\right] - \mathrm{E}_{\mathrm{Q}(z)}\left[\log \mathrm{E}_{\mathrm{Q}(z')}\left[f(z, z')\right]\right]. \tag{4}$$

Of course, in reality we do not have an arbitrarily flexible classifier. The original InfoNCE paper does not even use a neural network; instead, they use a linear classifier,

$$f(z, z') = \exp\left(z^T W z'\right). \tag{5}$$

Here, W is a weight matrix, and Tschannen et al. (2019) argue that this simple classifier is critical to the success of InfoNCE.

3 Methods

We begin by looking at the unstructured SSVAEs with a single latent and observed variable. This gives useful intuition but does not recover InfoNCE. We then go on to look at the structured SSVAE with two latent and two observed variables which recovers InfoNCE.

3.1 Unstructured SSVAEs

In a standard variational autoencoder, we specify parametric forms (e.g. using neural networks) for the prior, $\mathrm{P}(z)$, the likelihood, $\mathrm{P}(x|z)$ and the approximate posterior, $\mathrm{Q}(z|x)$. However, in an SSVAE, we specify only the prior, $\mathrm{P}(z)$, and the approximate posterior, $\mathrm{Q}(z|x)$. The likelihood, $\mathrm{P}(x|z)$, is given implicitly. In a simple model with one latent variable, z, and one observation, x, the likelihood is given by Bayes theorem,

$$\mathrm{P}(x|z) = \frac{\mathrm{Q}(z|x)\,\mathrm{P}_{\mathrm{true}}(x)}{\mathrm{Q}(z)}. \tag{6}$$

Here, $\mathrm{P}_{\mathrm{true}}(x)$ is the true distribution over data, which is fixed, independent of parameters, and in general different from the model's distribution, $\mathrm{P}(x)$. We cannot evaluate the probability density $\mathrm{P}_{\mathrm{true}}(x)$, and hence we cannot evaluate the probability density $\mathrm{P}(x|z)$ (it will turn out that we do not need to). Next, $\mathrm{Q}(z|x)$ is the variational approximate posterior. Specifying the likelihood, $\mathrm{P}(x|z)$, in terms of the approximate posterior, $\mathrm{Q}(z|x)$, is highly unusual in the variational framework — indeed, we believe we are the first to propose it. Nonetheless, it is perfectly valid, as it is simply equivalent to sharing parameters between $\mathrm{P}(x|z)$ and $\mathrm{Q}(z|x)$, which is not ruled out by the standard variational formulation. Indeed, sharing parameters between the prior/likelihood and the approximate posterior is frequently used in practice (Ustyuzhaninov et al., 2020; Ober & Aitchison, 2021b; Aitchison et al., 2021; Ober & Aitchison, 2021a). Finally, we define marginal approximate posterior, $\mathrm{Q}(z)$, as,

$$\mathrm{Q}(z) = \int dx\, \mathrm{Q}(z|x)\,\mathrm{P}_{\mathrm{true}}(x). \tag{7}$$

Next, $\mathrm{P}(x|z)$ defined in Eq. (6) is a valid distribution over x' (albeit one whose probability density cannot be computed) because it is non-negative and integrates to 1. In particular, integrating, and substituting

Eq. (7) into Eq. (6),

$$\int dx \, \mathrm{P}\left(x|z\right) = \frac{\int dx \, \mathrm{Q}\left(z|x\right) \mathrm{P}_{\mathrm{true}}\left(x\right)}{\int dx' \, \mathrm{Q}\left(z|x'\right) \mathrm{P}_{\mathrm{true}}\left(x'\right)} = 1. \tag{8}$$

The model's joint distribution over x and z is thus,

$$\mathrm{P}\left(x, z\right) = \mathrm{P}\left(x|z\right) \mathrm{P}\left(z\right) = \mathrm{Q}\left(z|x\right) \mathrm{P}_{\mathrm{true}}\left(x\right) \frac{\mathrm{P}\left(z\right)}{\mathrm{Q}\left(z\right)}. \tag{9}$$

where, remember, $\mathrm{Q}\left(z|x\right)$ is our neural network encoder, $\mathrm{P}_{\mathrm{true}}\left(x\right)$ is the true data distribution, $\mathrm{P}\left(z\right)$ is our choice of prior, and $\mathrm{Q}\left(z\right)$ is given by Eq. (7). Substituting the likelihood (Eq. 6) into the ELBO (Eq. 1), we get,

$$\mathcal{L}(x) = \log \mathrm{P}_{\mathrm{true}}\left(x\right) + \mathrm{E}_{\mathrm{Q}(z|x)} \left[\log \frac{\mathrm{P}\left(z\right)}{\mathrm{Q}\left(z\right)}\right] \tag{10}$$

where $\mathrm{P}\left(z\right)$ is our parametric form for the prior and $\mathrm{Q}\left(z\right)$ is given by Eq. 7.

Remember that $\log \mathrm{P}_{\mathrm{true}}\left(x\right)$ is constant with respect to the parameters as $\mathrm{P}_{\mathrm{true}}$ is the true, fixed data distribution. This term can thus be treated as a constant for the purposes of optimizing the parameters of $\mathrm{P}\left(z\right)$ and $\mathrm{Q}\left(z|x\right)$. Thus, to optimize $\mathcal{L}(x)$, we need to focus on the density ratio, $\mathrm{P}\left(z\right) / \mathrm{Q}\left(z\right)$. However, this density ratio cannot be evaluated directly as we cannot evaluate $\mathrm{Q}\left(z\right)$ (Eq. 7). Instead, we could be inspired by InfoNCE and NCE in general to estimate this ratio using a classifier that distinguishes samples of $\mathrm{P}\left(z\right)$ from those of $\mathrm{Q}\left(z\right)$.

However, it turns out that this approach is unlikely to be useful for forming latent representations. In particular, consider taking the expectation of the ELBO (Eq. 10) over the true data distribution $\mathrm{P}_{\mathrm{true}}\left(x\right)$,

$$\begin{aligned} \mathrm{E}_{\mathrm{P}_{\mathrm{true}}(x)} \left[\mathcal{L}(x)\right] &= \mathrm{E}_{\mathrm{P}_{\mathrm{true}}(x)} \left[\log \mathrm{P}_{\mathrm{true}}\left(x\right)\right] + \mathrm{E}_{\mathrm{Q}(z)} \left[\log \frac{\mathrm{P}\left(z\right)}{\mathrm{Q}\left(z\right)}\right] \\ &= c - \mathrm{D}_{\mathrm{KL}} \left(\mathrm{Q}\left(z\right) \| \mathrm{P}\left(z\right)\right). \end{aligned} \tag{11}$$

Optimizing the ELBO thus matches the marginal distributions in latent space between $\mathrm{Q}\left(z\right)$ (Eq. 7) and our parametric prior, $\mathrm{P}\left(z\right)$. In essence all we are doing is to find an encoder, $\mathrm{Q}\left(z|x\right)$, from x to z such that, averaging over x from the data, the resulting z's have a distribution close to $\mathrm{P}\left(z\right)$. However, it is not at all clear that this will give us a good representation. For instance, if $\mathrm{P}\left(z\right)$ is Gaussian, and if noise in the data, x, is Gaussian, then it may be easier to get Gaussian z's by extracting noise, rather than (as we would like), extracting high-level structure. That said, it may still be possible to do something useful by applying identifiability results inspired by ICA (e.g. Khemakhem et al., 2020).

3.2 Structured SSVAEs

The previous section argued that an SSVAE with just one latent and observed variable is unlikely to give useful representations. Instead, consider a generative model with two observed variables, x and x', and two latent variables, z and z'. The approximate posterior is given in terms of neural network encoders for x and x' separately,

$$\mathrm{Q}\left(z, z'|x, x'\right) = \mathrm{Q}\left(z|x\right) \mathrm{Q}\left(z'|x'\right). \tag{12}$$

The generative model has structure $x \leftarrow z - z' \rightarrow x'$,

$$\mathrm{P}\left(x, x', z', z\right) = \mathrm{P}\left(x|z\right) \mathrm{P}\left(x'|z'\right) \mathrm{P}\left(z, z'\right). \tag{13}$$

where $\mathrm{P}\left(z, z'\right)$ may be a specific, parametric form (e.g. a Gaussian), and the decoders, $\mathrm{P}\left(x|z\right)$ and $\mathrm{P}\left(x'|z'\right)$ are given implicitly in terms of the encoders, $\mathrm{Q}\left(z|x\right)$ and $\mathrm{Q}\left(z'|x'\right)$ and the true marginal distributions of the

data, $\mathrm{P}_{\text{true}}(x)$ and $\mathrm{P}_{\text{true}}(x')$,

$$\mathrm{P}(x|z) = \frac{\mathrm{Q}(z|x)\,\mathrm{P}_{\text{true}}(x)}{\mathrm{Q}(z)} \tag{14a}$$

$$\mathrm{P}(x'|z') = \frac{\mathrm{Q}(z'|x')\,\mathrm{P}_{\text{true}}(x')}{\mathrm{Q}(z')} \tag{14b}$$

where,

$$\mathrm{Q}(z) = \int dx\, \mathrm{Q}(z|x)\,\mathrm{P}_{\text{true}}(x) \tag{15a}$$

$$\mathrm{Q}(z') = \int dx'\, \mathrm{Q}(z'|x')\,\mathrm{P}_{\text{true}}(x')\,. \tag{15b}$$

Now, we compute the model evidence (note we delay applying Jensen's inequality to get the ELBO),

$$\begin{aligned}\log \mathrm{P}(x, x') &= \log \int dz\; dz'\;\; \mathrm{Q}(z, z'|x, x')\, \frac{\mathrm{P}(x, x', z, z')}{\mathrm{Q}(z, z'|x, x')} \\ &= \log \mathrm{E}_{\mathrm{Q}(z,z'|x,x')}\left[\frac{\mathrm{P}(x, x', z, z')}{\mathrm{Q}(z, z'|x, x')}\right].\end{aligned} \tag{16}$$

Substituting for the approximate posterior (Eq. 12) and prior (Eq. 13),

$$\log \mathrm{P}(x, x') = \log \mathrm{E}_{\mathrm{Q}(z,z'|x,x')}\left[\frac{\mathrm{P}(x|z)}{\mathrm{Q}(z|x)}\frac{\mathrm{P}(x'|z')}{\mathrm{Q}(z'|x')}\,\mathrm{P}(z, z')\right]. \tag{17}$$

Substituting Eq. (14) and remembering that $\log \mathrm{P}_{\text{true}}(x)$ and $\log \mathrm{P}_{\text{true}}(x')$ are parameter-independent constants

$$\log \mathrm{P}(x, x') = \log \mathrm{E}_{\mathrm{Q}(z,z'|x,x')}\left[\frac{\mathrm{P}(z, z')}{\mathrm{Q}(z)\,\mathrm{Q}(z')}\right] + c \tag{18}$$

Finally, applying Jensen's inequality we get the ELBO,

$$\log \mathrm{P}(x, x') \geq \mathcal{L}(x', x) \tag{19}$$

$$\mathcal{L}(x', x) = \mathrm{E}_{\mathrm{Q}(z,z'|x,x')}\left[\log \frac{\mathrm{P}(z, z')}{\mathrm{Q}(z)\,\mathrm{Q}(z')}\right] + c \tag{20}$$

3.3 Deterministic encoders

Consider a deterministic encoder,

$$\mathrm{Q}(z|x) = \delta(z - g(x)) \tag{21a}$$

$$\mathrm{Q}(z'|x') = \delta(z' - g'(x')) \tag{21b}$$

where δ is the Kronecker delta. In this case, we have $z = g(x)$ and $z' = g'(x')$ and the ELBO bound is tight,

$$\log \mathrm{P}(x, x') = \mathcal{L}(x, x') = \log \frac{\mathrm{P}(z, z')}{\mathrm{Q}(z)\,\mathrm{Q}(z')} + c, \tag{22}$$

so, the ELBO is *equal* to the model evidence (up to constant factors).

3.4 Understanding the SSVAE objective

To understand how this objective behaves, consider its expectation under the data distribution, $\mathrm{P}_{\text{true}}(x, x')$,

$$\mathcal{L} = \mathrm{E}_{\mathrm{P}_{\text{true}}(x,x')}\left[\mathcal{L}(x, x')\right] = \mathrm{E}_{\mathrm{Q}(z,z')}\left[\log \frac{\mathrm{P}(z, z')}{\mathrm{Q}(z)\,\mathrm{Q}(z')}\right] + c \tag{23}$$

where,

$$\mathrm{Q}\left(z, z'\right) = \int dx\, dx'\ \mathrm{Q}\left(z|x\right) \mathrm{Q}\left(z'|x'\right) \mathrm{P}_{\mathrm{true}}\left(x, x'\right). \tag{24}$$

Adding and subtracting $\mathrm{E}_{\mathrm{Q}(z,z')}\left[\log \mathrm{Q}\left(z, z'\right)\right]$,

$$\mathcal{L} = \mathrm{E}_{\mathrm{Q}(z,z')}\left[\log \frac{\mathrm{Q}\left(z, z'\right)}{\mathrm{Q}\left(z\right)\mathrm{Q}\left(z'\right)}\right] + \mathrm{E}_{\mathrm{Q}(z,z')}\left[\log \frac{\mathrm{P}\left(z, z'\right)}{\mathrm{Q}\left(z, z'\right)}\right] + c, \tag{25}$$

allows us to identify two KL-divergence terms,

$$\mathcal{L} = \mathrm{D}_{\mathrm{KL}}\left(\mathrm{Q}\left(z, z'\right) \| \mathrm{Q}\left(z\right)\mathrm{Q}\left(z'\right)\right) - \mathrm{D}_{\mathrm{KL}}\left(\mathrm{Q}\left(z, z'\right) \| \mathrm{P}\left(z, z'\right)\right) + c \tag{26}$$

The first term is a mutual information. The objective therefore maximizes the mutual information between z and z' under $\mathrm{Q}\left(z, z'\right)$ (Eq. 24). At the same time, the second term is the KL-divergence between the parametric prior, $\mathrm{P}\left(z, z'\right)$, and $\mathrm{Q}\left(z, z'\right)$. Thus, the objective also encourages $\mathrm{Q}\left(z, z'\right)$ (Eq. 24) and the parametric prior, $\mathrm{P}\left(z, z'\right)$, to become more similar.

3.5 Recovering a maximum mutual-information objective

We can recover a mutual-information objective by giving an implicit definition of the prior over latent variables,

$$\mathrm{P}_{\mathrm{MI}}\left(z, z'\right) = \mathrm{Q}\left(z, z'\right), \tag{27}$$

in which case the KL-divergence between $\mathrm{P}\left(z, z'\right)$ and $\mathrm{Q}\left(z, z'\right)$ is zero, and we are left with just the mutual information (Eq. 2),

$$\mathcal{L}_{\mathrm{MI}} = \mathrm{E}_{\mathrm{Q}(z,z')}\left[\log \frac{\mathrm{Q}\left(z'|z\right)}{\mathrm{Q}\left(z'\right)}\right] + c, \tag{28}$$

3.6 Recovering the exact form for the InfoNCE estimator

Recent work has argued that the good representation arising from InfoNCE cannot be from maximizing mutual information alone, because the mutual information is invariant under arbitrary invertible transformations (Tschannen et al., 2019; Li et al., 2021). Instead, the good properties must arise somehow out of the simple mutual information estimator in Eq. (5). Remarkably, this estimator (at least in the infinite N limit) can be recovered by making a specific choice of prior in the latent space. We choose the prior on z implicitly, as $\mathrm{Q}\left(z\right)$, and we choose the distribution over z' conditioned on z to be given by an energy based model that depends on $\mathrm{Q}\left(z'\right)$ and an arbitrary coupling function, $f(z, z')$, which could be given by Eq. (5) as in the original InfoNCE, or could be more general,

$$\mathrm{P}_{\mathrm{InfoNCE}}\left(z\right) = \mathrm{Q}\left(z\right) \tag{29}$$

$$\mathrm{P}_{\mathrm{InfoNCE}}\left(z'|z\right) = \frac{1}{Z}\,\mathrm{Q}\left(z'\right) f(z, z'). \tag{30}$$

The normalizing constant, Z, is

$$Z = \int dz'\, \mathrm{Q}\left(z'\right) f(z, z') = \mathrm{E}_{\mathrm{Q}(z')}\left[f(z, z')\right]. \tag{31}$$

To obtain the ELBO objective in Eq. (19), we need the ratio,

$$\log \frac{\mathrm{P}_{\mathrm{InfoNCE}}\left(z, z'\right)}{\mathrm{Q}\left(z\right)\mathrm{Q}\left(z'\right)} = \log \frac{\mathrm{Q}\left(z\right)\frac{1}{Z}\,\mathrm{Q}\left(z'\right) f(z, z')}{\mathrm{Q}\left(z\right)\mathrm{Q}\left(z'\right)} = \log f(z, z') - \log \mathrm{E}_{\mathrm{Q}(z')}\left[f(z, z')\right] \tag{32}$$

Remarkably, this is exactly equal to the infinite limit of the InfoNCE objective in Eq. (4), so the full expected ELBO becomes,

$$\mathcal{L}_{\mathrm{InfoNCE}} = \mathrm{E}_{\mathrm{Q}(z,z')}\left[\log f(z, z')\right] - \mathrm{E}_{\mathrm{Q}(z)}\left[\log \mathrm{E}_{\mathrm{Q}(z')}\left[f(z, z')\right]\right] \tag{33}$$

as derived in (Wang & Isola, 2020) and used in (Li et al., 2021). This can be bounded by the usual finite-sample estimator, as described in (Oord et al., 2018).

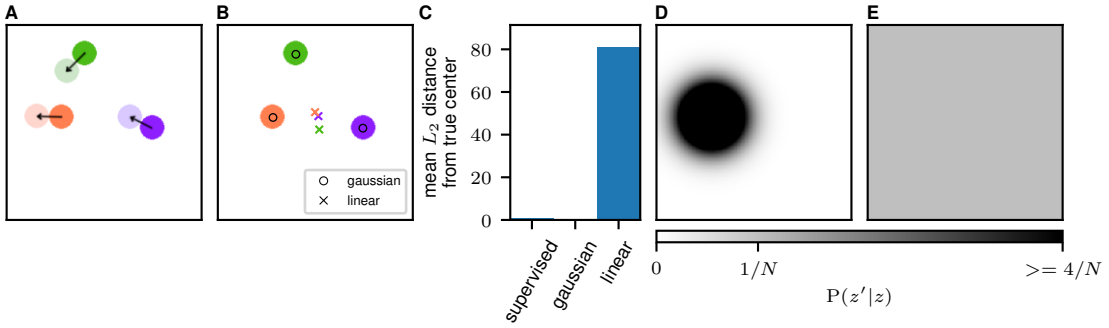

Figure 1: Results of the moving balls experiment. **A**) Example of the motion between consecutive frames. The balls move by a full diameter in a semi-random direction. **B**) Locations of the extracted ball centres, after supervised linear decoding. The standard InfoNCE setup fails to extract correct locations. **C**) The mean distance from the extracted and true centres of the balls for a supervised method, InfoNCE with a Gaussian discriminator after supervised decoding and InfoNCE with a linear discriminator after supervised decoding. **D**) Probability distribution for the next location of the coral ball in **A** according to an encoder trained with a Gaussian discriminator. **E**) Probability distribution for the next location of the same ball according to an encoder trained with a linear discriminator.

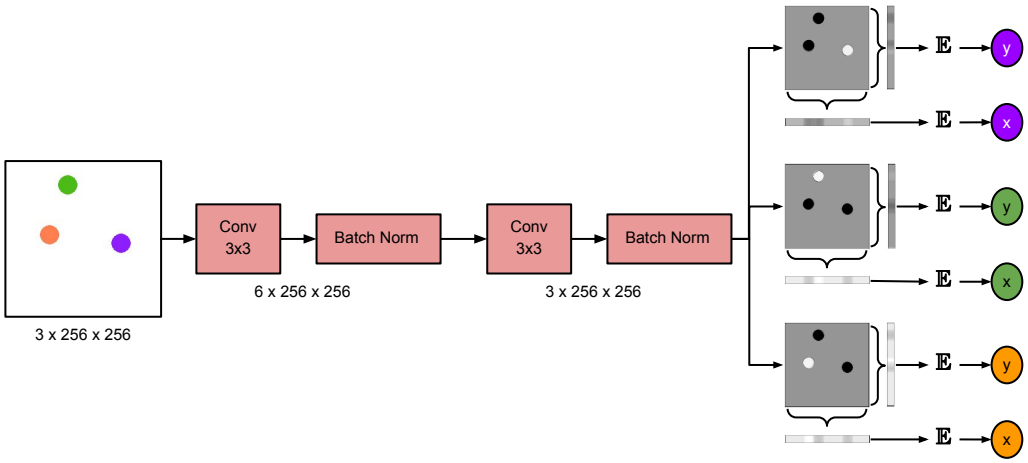

Figure 2: Architecture of the encoder neural network. The first of the two 3x3 convolutional layers outputs 6 feature maps and uses a ReLU activation. The second convolution outputs 3 feature maps and applies a sigmoid activation. For each of these 3 maps, we extract their centre of mass. This is done by summing each dimension and normalising it to 1. This is then used to perform a weighted average over the axis locations and get the final coordinates.

4 Results

Remember that the success of InfoNCE cannot have arisen due to mutual information maximization, because more complex and accurate mutual information estimators give worse representations (Tschannen et al., 2019). Thus, our work's key value is in giving a normative explanation for the success of InfoNCE, by showing that the choice of simplified mutual information estimator is equivalent to a choice of prior in an SSVAE, and of course we are free to choose a simplified prior which induces good representations if that captures our true beliefs about the latent space. Indeed, the connection we draw is so strong — showing that the InfoNCE is exactly equivalent to our SSVAE — that experimental comparisons become meaningless. Arguably, through that equivalence, we could potentially count all previous successes of InfoNCE as successes

also for SSVAEs. Indeed, note that if our theoretical results had been weaker, we would have been able to do a much stronger experiments section: if we had not shown exact equivalence, we could at least have made a meaningful comparison.

Of course, our approach does offer some advantages over the InfoNCE framework, but these advantages are conceptual. In particular, we cannot use mututal information maximization as a principle to choose an estimator that gives good representations, because better mututal information estimators give worse representations (Tschannen et al., 2019). But we show that the choice of simplified mututal information estimator is equivalent to the choice of prior in an SSVAE, and we can choose a prior that gives good high-level representations, assuming that this accords with our beliefs about the underlying system.

As the primary importance of our work is theoretical, we give only a simple proof-of-principle, of how choosing mutual information estimators based on priors is beneficial. In particular, consider a self-supervised learner that extracts the locations of three moving balls, based on videos of these balls bouncing around in a square (Fig. 1A).

We generated 900 images in a single continuous video with a resolution of 256×256 pixels. The three balls had a diameter of 32 pixels. Between consecutive frames the balls moved by a full diameter in a random direction, as illustrated in Fig. 1A. The movement trajectory was picked by taking the previous trajectory and adding a uniform noise of $-2°$ to $+2°$. If the picked movement resulted in a collision, we sampled a new trajectory by doubling the noise range until a valid trajectory is found.

We trained the model in a classic self-supervised manner. We encoded one "base" frame, one "target" frame (the next frame in a video sequence), along with a number of random frames. As usual, the network was trained to distinguish between the target frame (adjacent to the base frame) and random frames. We then trained a linear decoder in a supervised manner to return the (x, y) locations of the balls.

The encoder itself is a simple convolutional neural network, as shown in Fig. 2. It consists of 2 batch normalised convolutional layers with a kernel size of 3. The first layer uses ReLU as the activation function, while the second layer uses a sigmoid. At the output of the convolutional layers, we have 3 feature maps, which we interpret as the locations of the 3 different balls. We finally extract these locations by computing the centre of mass of the feature maps, giving a vector of six numbers as output (the x and y locations of the centres of mass of each feature map). The training itself was performed by using stochastic gradient descent with a learning rate of 0.005 over the course of 30 epochs. The batches were made of 30 random pairs of consecutive frames. For any pair, we use the second frame as the positive example and we use the second frame of the other pairs in the batch, as the random negative examples, against which we contrast.

When we naively applied the usual InfoNCE setup (using Eq. 5), self-supervised learning failed (linear in Fig. 1BC), because we did not correctly encode prior information about the structure of the problem. Critically, our prior is that for the adjacent frames, the locations extracted by the network will be close, while for random frames, the locations extracted by the network will be far apart. The linear estimator in Eq. (5) is not suitable for extracting the proximity of the ball locations, so it fails (linear in Fig. 1 BC). In particular, it corresponds to a non-sensical prior over z' given z,

$$\mathrm{P}_{\mathrm{InfoNCE}}\left(z'|z\right) = \frac{1}{Z}\,\mathrm{Q}\left(z'\right) f(z, z') \propto \exp\left(z^T W z'\right) \tag{34}$$

(where we have taken $\mathrm{Q}\left(z'\right)$ to be uniform). Instead, we would like a prior that encodes our knowledge that z' is likely to be close to z, such as,

$$\mathrm{P}\left(z'|z\right) = \mathcal{N}\left(z'; z, L^2\right). \tag{35}$$

This corresponds to a Gaussian RBF form for f,

$$f(z, z') = \exp\left(-\tfrac{1}{2L^2}(z - z')^2\right). \tag{36}$$

where L is a learned lengthscale. Critically, this sensible, effective prior was chosen entirely based on the prior viewpoint. It is very difficult to see how one would have arrived at this choice if we were using the mutual information estimator viewpoint. As such, the Bayesian SSVAE framework naturally allows us to incorporate our knowledge in the form of structured latent variables, with rich prior dependencies.

5 Related work

Perhaps the closest prior work is Zimmermann et al. (2021), which also identifies an interpretation of InfoNCE as inference in a principled generative model. Unlike this work, we identify a connection between the InfoNCE objective and the ELBO or model evidence. Moreover, their proof requires complex geometric properties, whereas our's merely involves straightforward manipulations of probability distributions. In addition, their approach requires four restrictive assumptions. First, they assume deterministic encoder, e.g $\mathrm{Q}\left(z|x\right) = \delta\left(z - g(x)\right)$. In contrast, all our theory applies to stochastic encoders. While we do explicitly consider deterministic encoders in Sec. 3.3, this is only to show that with deterministic encoders, the ELBO bound is tight — all the derivations outside of this very small section (which includes all our key derivations) use fully general encoders, $\mathrm{Q}\left(z|x\right)$ and $\mathrm{Q}\left(z'|x'\right)$. Second, they assume that $z(x)$ is invertible, i.e. that there exists a deterministic decoder $x(z')$, which is not necessary in our framework. Third, they assume that the latent space is unit hypersphere, while in our framework there is no constraint on the latent space. Fourth, they assume the ground truth marginal of the latents of the generative process is uniform, whereas our framework accepts any choice of ground-truth marginal. As such, our framework has considerably more flexibility to include rich priors on complex, structured latent spaces.

Other work looked at the specific case of isolating content from style (von Kügelgen et al., 2021). This work used a similar derivation to that in Zimmermann et al. (2021) with slightly different assumptions. While they still required deterministic, invertible encoders, they relax e.g. uniformity in the latent space. But because they are working in the specific case of style and content variables, they make a number of additional assumptions on those variables. Importantly, they again do not connect the InfoNCE objective with the ELBO or model evidence.

Very different methods use noise-contrastive methods to update a VAE prior (Aneja et al., 2020). Importantly, they still use an explicit decoder.

There is a large class of work that seeks to use VAEs to extract useful, disentangled representations (e.g. Burgess et al., 2018; Chen et al., 2018; Kim & Mnih, 2018; Mathieu et al., 2019; Joy et al., 2020). Again, this work differs from our work in that it uses explicit decoders and thus does not identify an explicit link to self-supervised learning.

Likewise, there is work on using GANs to learn interpretable latent spaces (e.g. Chen et al., 2016a). Importantly, GANs learn a decoder (mapping from the random latent space to the data domain). Moreover, GANs use a classifier to estimate a density ratio. However, GANs estimate this density ratio for the data, x and x', whereas InfoNCE, like the methods described here, uses a classifier to estimate a density ratio on the latent space, z and z'.

There is work on reinterpreting classifiers as energy-based probabilistic generative models (e.g. Grathwohl et al., 2019), which is related if we view SSL methods as being analogous to a classifier. Our work is very different, if for no other reason than because it is not possible to sample data from an SSVAE (even using a method like MCMC), because the decoder is written in terms of the unknown true data distribution.

6 Conclusions

In conclusion, we have seen that the ELBO in an SSVAE is equal to the mutual information with one choice of prior, and equal to the InfoNCE parametric mutual information estimator with an alternative prior (up to constants). As such, we unify contrastive semi-supervised learning with generative self-supervised learning (or unsupervised learning). In addition, we provide a principled framework for using simple parametric models in the latent space to enforce disentangled representations, and our framework allows us to use Bayesian intuition to form richer priors on the latent space. Finally, our framework provides the basis for future work to combine self-supervised learning with Bayesian (generative) approaches.

References

David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski. A learning algorithm for boltzmann machines. *Cognitive science*, 9(1):147–169, 1985.

Laurence Aitchison, Adam Yang, and Sebastian W Ober. Deep kernel processes. In *International Conference on Machine Learning*, pp. 130–140. PMLR, 2021.

Jyoti Aneja, Alexander Schwing, Jan Kautz, and Arash Vahdat. Ncp-vae: Variational autoencoders with noise contrastive priors. *arXiv preprint arXiv:2010.02917*, 2020.

Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-vae. *arXiv preprint arXiv:1804.03599*, 2018.

Ricky TQ Chen, Xuechen Li, Roger Grosse, and David Duvenaud. Isolating sources of disentanglement in variational autoencoders. *arXiv preprint arXiv:1802.04942*, 2018.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *arXiv preprint arXiv:1606.03657*, 2016a.

Xi Chen, Diederik P Kingma, Tim Salimans, Yan Duan, Prafulla Dhariwal, John Schulman, Ilya Sutskever, and Pieter Abbeel. Variational lossy autoencoder. *arXiv preprint arXiv:1611.02731*, 2016b.

Peter Dayan, Geoffrey E Hinton, Radford M Neal, and Richard S Zemel. The helmholtz machine. *Neural computation*, 7(5):889–904, 1995.

Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pp. 1422–1430, 2015.

Garoe Dorta, Sara Vicente, Lourdes Agapito, Neill DF Campbell, and Ivor Simpson. Structured uncertainty prediction networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5477–5485, 2018.

Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(9):1734–1747, 2015.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

Will Grathwohl, Kuan-Chieh Wang, Jörn-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. *arXiv preprint arXiv:1912.03263*, 2019.

Geoffrey E Hinton, Peter Dayan, Brendan J Frey, and Radford M Neal. The" wake-sleep" algorithm for unsupervised neural networks. *Science*, 268(5214):1158–1161, 1995.

Tom Joy, Sebastian Schmon, Philip Torr, N Siddharth, and Tom Rainforth. Capturing label characteristics in vaes. In *International Conference on Learning Representations*, 2020.

Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvarinen. Variational autoencoders and nonlinear ica: A unifying framework. In *International Conference on Artificial Intelligence and Statistics*, pp. 2207–2217. PMLR, 2020.

Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, pp. 2649–2658. PMLR, 2018.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Yazhe Li, Roman Pogodin, Danica J Sutherland, and Arthur Gretton. Self-supervised learning with kernel dependence maximization. *arXiv preprint arXiv:2106.08320*, 2021.

Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

Emile Mathieu, Tom Rainforth, Nana Siddharth, and Yee Whye Teh. Disentangling disentanglement in variational autoencoders. In *International Conference on Machine Learning*, pp. 4402–4412. PMLR, 2019.

Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pp. 69–84. Springer, 2016.

Sebastian Ober and Laurence Aitchison. A variational approximate posterior for the deep Wishart process. *Advances in Neural Information Processing Systems*, 34, 2021a.

Sebastian W Ober and Laurence Aitchison. Global inducing point variational posteriors for Bayesian neural networks and deep Gaussian processes. In *International Conference on Machine Learning*, pp. 8248–8259. PMLR, 2021b.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *International Conference on Machine Learning*, pp. 5171–5180. PMLR, 2019.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pp. 1278–1286. PMLR, 2014.

Michael Tschannen, Josip Djolonga, Paul K Rubenstein, Sylvain Gelly, and Mario Lucic. On mutual information maximization for representation learning. *arXiv preprint arXiv:1907.13625*, 2019.

Ivan Ustyuzhaninov, Ieva Kazlauskaite, Markus Kaiser, Erik Bodin, Neill Campbell, and Carl Henrik Ek. Compositional uncertainty in deep gaussian processes. In *Conference on Uncertainty in Artificial Intelligence*, pp. 480–489. PMLR, 2020.

Julius von Kügelgen, Yash Sharma, Luigi Gresele, Wieland Brendel, Bernhard Schölkopf, Michel Besserve, and Francesco Locatello. Self-supervised learning with data augmentations provably isolates content from style. *arXiv preprint arXiv:2106.04619*, 2021.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pp. 9929–9939. PMLR, 2020.

Roland S Zimmermann, Yash Sharma, Steffen Schneider, Matthias Bethge, and Wieland Brendel. Contrastive learning inverts the data generating process. *arXiv preprint arXiv:2102.08850*, 2021.