InfoNCE is a variational autoencoder

Anonymous authors
Paper under double-blind review

Abstract

There are two main approaches to self-supervised learning (SSL): generative SSL, where we learn a full probabilistic model of all the inputs, or contrastive SSL where we train on a supervised learning task that has been carefully designed to encourage good representations. We reconcile these generative and contrastive SSL, by showing that modern contrastive SSL methods including InfoNCE which maximize mutual information (MI) implicitly learn a full probabilistic model of the inputs, parameterised as a variational autoencoder (VAE). Under a particular choice of prior we show that the MI is equal to the VAE objective (the ELBO) which is in turn equal to the Bayesian model evidence for a deterministic encoder. This establishes a profound connection between Bayesian inference and information theory. However, practical SSL methods such as InfoNCE do not optimize the true MI, both because it is intractable and because it can give arbitrarily entangled representations, which are unlikely to be useful for representation learning (Tschannen et al., 2019). Instead, InfoNCE learns good representations by using a loose bound on the MI. Tschannen et al. (2019) thus raise a key question: does it really make sense to motivate an objective that works (i.e. the InfoNCE objective) as a loose bound on an objective that does not work (i.e. the true MI which gives arbitrarily entangled representations). We offer an alternative motivation for the InfoNCE objective by showing that in the infinite sample limit it is equal to the log-Bayesian model evidence but only bounds the MI.

1 Introduction

A common challenge occuring across machine learning is to extract useful, structured representations from unlabelled data (such as images). This problem is known as self-supervised learning, and there are two broad approaches: generative and contrastive (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 maximizing the true mutual information could lead to arbitrarily entangled representations, as the mutual information is invariant under arbitrary invertible transformations (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) which forms only a loose bound on the true MI. This is highly problematic: Tschannen et al. (2019) argue tha better MI estimators give worse representations (Tschannen et al., 2019), so InfoNCE's success with a highly approximate estimator cannot be due to maximizing mutual information, but instead appears to be due to an ad-hoc choice of simplified mutual information estimator. 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 contrastive self-supervised variational autoencoders (CSSVAEs). The CSSVAE objective is the ELBO, which we show is equivalent to the log-Bayesian model evidence in the usual case where the encoders are deterministic (Sec. 3.3). We show that the log-Bayesian model evidence is equal to the MI under one choice of prior (Sec. 3.5), and equal to the infinite-sample limit of the InfoNCE objectie under a different choice of prior(Sec. 3.6). At the same time, the InfoNCE objective forms only a bound on the true MI, even in the infinite sample setting (Sec. 2.2). This would argue that the InfoNCE objective is better motivated in terms of the log-Bayesian model evidence (as they are equal), as opposed to the mutual information (as the InfoNCE objective only forms a bound). This unifies contrastive and generative SSL. Finally, we highlight that the generative model viewpoint is useful for designing new InfoNCE-like learning methods in terms of priors in the latent space. We give an example in a toy system in which the usual choice of InfoNCE objective fails completely, but a modified system that exploits our prior knowledge about dynamics in the latent space succeeds (Sec. 4).

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.

We can rewrite the ELBO as an expected log-likelihood plus a KL-divergence,

$$\mathcal{L} = \mathrm{E}_{\mathrm{Q}(z|x)} \left[\mathrm{P}(x|z) \right] - \mathrm{D}_{\mathrm{KL}} \left(\mathrm{Q}(z|x) \| \mathrm{P}(z) \right). \tag{2}$$

That KL-divergence is very closely related (with a particular choice of $\mathrm{P}(z)$) to the MI (Alemi et al., 2018; Chen et al., 2016b), so the KL-divergence can be understood intuitively as reducing the MI between data, x and latent, z. However this connection between the ELBO and MI is entirely irrelevant for our work: at no point do we compute the MI between data and latents. Our results instead relate to a very different quantity: the MI between different latent variables. Of course, these two MIs are entirely different quantities, so we cannot expect them to behave similarly in any way.

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. We consider stochastic encoders, $\mathrm{Q}(x|z)$ and $\mathrm{Q}(x'|z')$, which are usually, but do not have to be, chosen to be deterministic,

$$\mathrm{Q}(z|x) = \delta(z - g(x)) \tag{3a}$$

$$\mathrm{Q}(z'|x') = \delta(z' - g'(x')). \tag{3b}$$

Again, we can but do not have to choose $g = g'$. The InfoNCE objective was originally motivated as maximizing the mutual information between latent representations,

$$\mathrm{I}(z; z') = \mathrm{E}_{\mathrm{Q}(z,z')}\left[\log \frac{\mathrm{Q}(z'|z)}{\mathrm{Q}(z')}\right]. \tag{4}$$

Here, we are using Q rather than P for consistency with VAE derivations in the methods. The distributions, $\mathrm{Q}(z'|z)$ and $\mathrm{Q}(z')$ are a conditional and marginal of the joint distribution, $\mathrm{Q}(z, z')$. This joint distribution is formed by taking datapoints, (x, x') drawn from the true data distribution, $\mathrm{P}_{\mathrm{true}}(x, x')$, and encoding them with the encoders $\mathrm{Q}(x|z)$ and $\mathrm{Q}(x'|z')$,

$$\mathrm{Q}(z, z') = \int dx\, dx'\;\; \mathrm{Q}(z|x)\,\mathrm{Q}(z'|x')\,\mathrm{P}_{\mathrm{true}}(x, x'). \tag{5}$$

Of course, x and x' exhibit dependencies under the true distribution, $\mathrm{P}_{\mathrm{true}}(x, x')$, so z and z' exhibit dependencies under $\mathrm{Q}(z, z')$, otherwise the whole exercise would not make much sense. As the mutual information is difficult to estimate, InfoNCE uses a bound, $\mathcal{I}_N(f)$, based on a classifier that uses f to distinguish 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),

$$\mathrm{I}(z; z') \geq \mathcal{I}_N(f) = \mathrm{E}\left[\log \frac{f(z, z')}{f(z, z') + \sum_{j=1}^{N} f(z, z'_j)}\right] + \log N. \tag{6}$$

Here, the expectation is taken over $\mathrm{Q}(z, z') \prod_j \mathrm{Q}(z'_j)$, and we use this objective to optimize the parameters of f, and the encoders. There are two source of slack in this bound, arising from finite N and a restrictive choice of f. To start, we can reduce but not eliminate slack by taking the limit as N goes to infinity, (Oord et al., 2018),

$$\mathrm{I}(z; z') \geq \mathcal{I}_\infty(f) \geq \mathcal{I}_N(f). \tag{7}$$

However, the bound only becomes tight if we additionally optimize an arbitrarily flexible f (Oord et al., 2018),

$$\mathrm{I}(z; z') = \max_f \mathcal{I}_\infty(f). \tag{8}$$

If we restrict f to a class $\mathcal{F}$ which does not include an optimal f, then the bound does not become tight, even as $N \to \infty$,

$$\mathrm{I}(z; z') > \max_{f \in \mathcal{F}} \mathcal{I}_\infty(f). \tag{9}$$

In reality InfoNCE does indeed use a highly restrictive class of function for f, which can be expected to give a loose bound on the MI (Oord et al., 2018),

$$f(z, z') = \exp\left(z^T W z'\right). \tag{10}$$

This raises the question of why we do not use a more flexible f if our goal really is to maximize the MI. The answer that our goal is not ultimately to maximize the MI. Our goal is ultimately to learn a good representation, and MI is merely a means to that end. Further, Tschannen et al. (2019) argue that optimizing the true MI is likely to lead give poor repesentations, as the MI is invariant to arbitrary invertible transformations that can entangle the representation. They go on to argue that it is precisely the restrictive family of functions, $f \in \mathcal{F}$, in Eq. (10), corresponding to a loose bound on the MI, that encourages good representations. Tschannen et al. (2019) thus raise an important question: does it really make sense to motivate an objective that works (the InfoNCE objective) as a loose bound on an objective that does not work (the mutual information). We offer an alternative motivation by showing that the InfoNCE objective is equal to the log-Bayesian model evidence under a particular choice of prior and with a deterministic encoder.

3 Theoretical results

We begin by looking at the unstructured CSSVAEs 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 CSSVAE with two latent and two observed variables which does recover InfoNCE.

3.1 Unstructured CSSVAEs

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 CSSVAE, 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,

$$\mathrm{P}(x|z) = \frac{\mathrm{Q}(z|x)\,\mathrm{P}_{\text{true}}(x)}{\mathrm{Q}(z)}. \tag{11}$$

Here, $\mathrm{P}_{\text{true}}(x)$ 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 $\mathrm{P}_{\text{true}}(x)$, 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, parameterised e.g. as a neural network. Specifying the likelihood, $\mathrm{P}(x|z)$, in terms of the approximate posterior, $\mathrm{Q}(z|x)$, is highly unusual in the variational framework (thought see Walker et al., 2022). Nonetheless, it is perfectly valid; any likelihood that can be written as Eq. (11) could also be written out more explicitly in terms of parameters that are shared between $\mathrm{P}(x|z)$ and $\mathrm{Q}(z|x)$. This sharing is allowed in the standard variational formulation, and indeed is common in theory and practice (Zhao et al., 2018; Ustyuzhaninov et al., 2020; Ober & Aitchison, 2021b; Aitchison et al., 2021; Ober & Aitchison, 2021a). Finally, we define marginal approximate posterior, $\mathrm{Q}(z)$, as,

$$\mathrm{Q}(z) = \int dx\, \mathrm{Q}(z|x)\,\mathrm{P}_{\text{true}}(x)\,. \tag{12}$$

Next, $\mathrm{P}(x|z)$ defined in Eq. (11) is a valid distribution over x' (albeit one whose probability density is intractable) because it is non-negative and integrates to 1. In particular, integrating, and substituting Eq. (12) into Eq. (11),

$$\int dx\, \mathrm{P}(x|z) = \frac{\int dx\, \mathrm{Q}(z|x)\,\mathrm{P}_{\text{true}}(x)}{\int dx'\, \mathrm{Q}(z|x')\,\mathrm{P}_{\text{true}}(x')} = 1. \tag{13}$$

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

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

where, remember, $\mathrm{Q}(z|x)$ is our neural network encoder, $\mathrm{P}_{\text{true}}(x)$ is the true data distribution, $\mathrm{P}(z)$ is our choice of prior, and $\mathrm{Q}(z)$ is given by Eq. (12). Substituting the likelihood (Eq. 11) into the ELBO (Eq. 1),

we get,

$$\mathcal{L}(x) = \log \mathrm{P}_{\mathrm{true}}(x) + \mathrm{E}_{\mathrm{Q}(z|x)}\left[\log \frac{\mathrm{P}(z)}{\mathrm{Q}(z)}\right] \tag{15}$$

where $\mathrm{P}(z)$ is our parametric form for the prior and $\mathrm{Q}(z)$ is given by Eq. 12.

Remember that $\log \mathrm{P}_{\mathrm{true}}(x)$ 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 $\mathrm{P}(z)$ and $\mathrm{Q}(z|x)$. Thus, to optimize $\mathcal{L}(x)$, we need to focus on the density ratio, $\mathrm{P}(z) / \mathrm{Q}(z)$. However, this density ratio cannot be evaluated directly as we cannot evaluate $\mathrm{Q}(z)$ (Eq. 12). Instead, we could be inspired by InfoNCE and NCE in general to estimate this ratio using a classifier that distinguishes samples of $\mathrm{P}(z)$ from those of $\mathrm{Q}(z)$.

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. 15) over the true data distribution $\mathrm{P}_{\mathrm{true}}(x)$,

$$\begin{aligned}\mathrm{E}_{\mathrm{P}_{\mathrm{true}}(x)}\left[\mathcal{L}(x)\right] &= \mathrm{E}_{\mathrm{P}_{\mathrm{true}}(x)}\left[\log \mathrm{P}_{\mathrm{true}}(x)\right] + \mathrm{E}_{\mathrm{Q}(z)}\left[\log \frac{\mathrm{P}(z)}{\mathrm{Q}(z)}\right] \\ &= c - \mathrm{D}_{\mathrm{KL}}\left(\mathrm{Q}(z) \| \mathrm{P}(z)\right).\end{aligned} \tag{16}$$

Optimizing the ELBO thus matches the marginal distributions in latent space between $\mathrm{Q}(z)$ (Eq. 12) and our parametric prior, $\mathrm{P}(z)$. In essence all we are doing is to find an encoder, $\mathrm{Q}(z|x)$, from x to z such that, averaging over x from the data, the resulting z's have a distribution close to $\mathrm{P}(z)$. It is not at all clear that this will give us a good representation. For instance, if $\mathrm{P}(z)$ 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 CSSVAEs

The previous section argued that an CSSVAE 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,

$$\mathrm{Q}(z, z'|x, x') = \mathrm{Q}(z|x)\,\mathrm{Q}(z'|x'). \tag{17}$$

To model this structure, we use a generative model of the form, $x \leftarrow z - z' \rightarrow x'$,

$$\mathrm{P}(x, x', z', z) = \mathrm{P}(x|z)\,\mathrm{P}(x'|z')\,\mathrm{P}(z, z'). \tag{18}$$

which implies dependencies between x and x' and between z and z'. Here, $\mathrm{P}(z, z')$ may be a specific, parametric form (e.g. a Gaussian). The decoders, $\mathrm{P}(x|z)$ and $\mathrm{P}(x'|z')$ are given implicitly in terms of the encoders, $\mathrm{Q}(z|x)$ and $\mathrm{Q}(z'|x')$ and the true marginal distributions of the data, $\mathrm{P}_{\mathrm{true}}(x)$ and $\mathrm{P}_{\mathrm{true}}(x')$,

$$\mathrm{P}(x|z) = \frac{\mathrm{Q}(z|x)\,\mathrm{P}_{\mathrm{true}}(x)}{\mathrm{Q}(z)} \tag{19a}$$

$$\mathrm{P}(x'|z') = \frac{\mathrm{Q}(z'|x')\,\mathrm{P}_{\mathrm{true}}(x')}{\mathrm{Q}(z')} \tag{19b}$$

where $\mathrm{Q}\,z$ and $\mathrm{Q}\,z'$ can be understood as marginals of $\mathrm{Q}(z, z')$ in Eq. (5), but are most easily written,

$$\mathrm{Q}(z) = \int dx\, \mathrm{Q}(z|x)\,\mathrm{P}_{\mathrm{true}}(x) \tag{20a}$$

$$\mathrm{Q}(z') = \int dx'\, \mathrm{Q}(z'|x')\,\mathrm{P}_{\mathrm{true}}(x'). \tag{20b}$$

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

$$\begin{aligned}\log \mathrm{P}\left(x, x'\right) &= \log \int dz\, dz'\;\; \mathrm{Q}\left(z, z' | x, x'\right) \frac{\mathrm{P}\left(x, x', z, z'\right)}{\mathrm{Q}\left(z, z' | x, x'\right)} \\ &= \log \mathrm{E}_{\mathrm{Q}(z, z' | x, x')} \left[\frac{\mathrm{P}\left(x, x', z, z'\right)}{\mathrm{Q}\left(z, z' | x, x'\right)}\right]. \end{aligned} \tag{21}$$

Substituting for the approximate posterior (Eq. 17) and prior (Eq. 18),

$$\log \mathrm{P}\left(x, x'\right) = \log \mathrm{E}_{\mathrm{Q}(z, z' | x, x')} \left[\frac{\mathrm{P}\left(x | z\right)}{\mathrm{Q}\left(z | x\right)} \frac{\mathrm{P}\left(x' | z'\right)}{\mathrm{Q}\left(z' | x'\right)} \mathrm{P}\left(z, z'\right)\right]. \tag{22}$$

Substituting Eq. (19) and remembering that $\log \mathrm{P}_{\mathrm{true}}\left(x\right)$ and $\log \mathrm{P}_{\mathrm{true}}\left(x'\right)$ are parameter-independent constants

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

Finally, applying Jensen's inequality we get the ELBO for a datapoint,

$$\log \mathrm{P}\left(x, x'\right) \geq \mathcal{L}(x', x) \tag{24}$$

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

And averaging over datapoints,

$$\mathcal{L} = \mathrm{E}_{\mathrm{P}_{\mathrm{true}}(x, x')} \left[\mathcal{L}(x, x')\right] = \mathrm{E}_{\mathrm{Q}(z, z')} \left[\log \frac{\mathrm{P}\left(z, z'\right)}{\mathrm{Q}\left(z\right) \mathrm{Q}\left(z'\right)}\right] + c. \tag{26}$$

3.3 Under a deterministic encoder, the CSSVAE ELBO is equal to the log-Bayesian model evidence

For a deterministic encoder (Eq. 3), where $z = g(x)$ and $z' = g'(x')$, it can be shown that the ELBO is equal to the model evidence. In particular, as the encoder is deterministic, the expectations for the log-marginal likelihood (Eq. 23) and the ELBO (Eq. 25) can be evaluated straightforwardly, and are equal,

$$\log \mathrm{P}\left(x, x'\right) = \log \frac{\mathrm{P}\left(z, z'\right)}{\mathrm{Q}\left(z\right) \mathrm{Q}\left(z'\right)} + c = \mathcal{L}(x', x). \tag{27}$$

so, the ELBO becomes *equal* to the log-Bayesian model evidence. This is expected if we remember that the variational bound in Eq. (25) arose from applying Jensen's inequality to Eq. (23). Importantly, looseness in the Jensen bound arises from variability in $\mathrm{P}\left(z, z'\right) / \mathrm{Q}\left(z\right) \mathrm{Q}\left(z'\right)$, so if this quantity is fixed because we have a deterministic encoder, then Jensen's bound is tight.

3.4 The CSSVAE ELBO can be written as the mutual information plus a KL divergence

To get an intuitive understanding of the ELBO we take Eq. (26) and add and subtract $\mathrm{E}_{\mathrm{Q}(z, z')} \left[\log \mathrm{Q}\left(z, z'\right)\right]$,

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

The first term is the mutual information between z and z' under $\mathrm{Q}\left(z, z'\right)$ (Eq. 5), and the second term is a KL-divergence,

$$\mathcal{L} = \mathrm{I}\left(z; z'\right) - \mathrm{D}_{\mathrm{KL}}\left(\mathrm{Q}\left(z, z'\right) \| \mathrm{P}\left(z, z'\right)\right) + c. \tag{29}$$

This objective therefore encourages large mutual information (Eq. 5), while encouraging $\mathrm{Q}\left(z, z'\right)$ to lie close to the prior, $\mathrm{P}\left(z, z'\right)$.

3.5 Under one prior, the CSSVAE ELBO is equal to the mutual-information (up to a constant)

The ELBO can be reduced to just the mutual-information by defining the prior implicitly as,

$$\mathrm{P}_{\mathrm{MI}}\left(z, z'\right) = \mathrm{Q}\left(z, z'\right). \tag{30}$$

Substituting this choice into Eq. (29), the KL-divergence is zero, so we are left only with the mutual information between z and z' (and a constant),

$$\mathcal{L}_{\mathrm{MI}} = \mathrm{I}\left(z; z'\right) + c. \tag{31}$$

3.6 Under a different prior, the CSSVAE ELBO is equal to the infinite-sample InfoNCE objective (up to a constant)

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 fact that the InfoNCE objective forms only a loose bound on the true MI, even in the infinite sample limit Eq. (9). Remarkably, the infinite-sample InfoNCE objective is equal to the ELBO (or log-Bayesian model evidence for deterministic encoders) for a specific choice of prior. In particular, 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 unrestricted coupling function, $f(z, z')$ (we could of course use Eq. 10),

$$\mathrm{P}_{\mathrm{InfoNCE}}\left(z\right) = \mathrm{Q}\left(z\right) \tag{32}$$

$$\mathrm{P}_{\mathrm{InfoNCE}}\left(z'|z\right) = \frac{1}{Z(z)}\,\mathrm{Q}\left(z'\right) f(z, z'). \tag{33}$$

The normalizing constant, Z, is

$$Z(z) = \int dz'\,\mathrm{Q}\left(z'\right) f(z, z') = \mathrm{E}_{\mathrm{Q}(z')}\left[f(z, z')\right]. \tag{34}$$

Substituting these choices into Eq. (26), the average ELBO or log-Bayesian model evidence becomes,

$$\mathcal{L}_{\mathrm{InfoNCE}}(f) = \mathrm{E}_{\mathrm{Q}(z,z')}\left[\log \frac{\mathrm{Q}\left(z\right) \frac{1}{Z(z)}\,\mathrm{Q}\left(z'\right) f(z, z')}{\mathrm{Q}\left(z\right)\mathrm{Q}\left(z'\right)}\right] + c. \tag{35}$$

Cancelling $\mathrm{Q}\left(z\right)\mathrm{Q}\left(z'\right)$,

$$\mathcal{L}_{\mathrm{InfoNCE}}(f) = \mathrm{E}_{\mathrm{Q}(z,z')}\left[\log f(z, z') - \log Z(z)\right] + c, \tag{36}$$

and substituting for $Z(z)$ gives,

$$\mathcal{L}_{\mathrm{InfoNCE}}(f) = \mathrm{E}_{\mathrm{Q}(z,z')}\left[\log f(z, z')\right] - \mathrm{E}_{\mathrm{Q}(z)}\left[\log \mathrm{E}_{\mathrm{Q}(z')}\left[f(z, z')\right]\right] + c. \tag{37}$$

Following Wang & Isola (2020) and Li et al. (2021) the right hand side can be identified as the infinite sample InfoNCE objective that we introduced in Sec. 2.2,

$$\mathcal{L}_{\mathrm{InfoNCE}}(f) = \mathcal{I}_{\infty}(f) + c. \tag{38}$$

Thus in this choice of model, the log-Bayesian model evidence (for a deterministic encoder), $\mathcal{L}_{\mathrm{InfoNCE}}(f)$, is exactly equal to the infinite-sample InfoNCE objective, $\mathcal{I}_{\infty}(f)$. This would argue that the InfoNCE objective has a closer link to the log-Bayesian model evidence than it does to the MI, as the infinite-sample InfoNCE is exactly equal to the log-Bayesian model evidence, but forms only a bound on the MI,

$$\mathrm{I}\left(z; z'\right) \geq \mathcal{I}_{\infty}(f) = \mathcal{L}_{\mathrm{InfoNCE}}(f) + c. \tag{39}$$

Of course, in practice, InfoNCE uses finite samples, because the infinite limit in Eq. (37) is intractable. Likewise, if we were to practically use a CSSVAE with this prior, we would hit exactly the same issue, that we are not able to compute Eq. (37). The solution in both cases would be exactly the same, to use the bound given by the usual finite-sample estimator, as originally described in the InfoNCE framework (Oord et al., 2018).

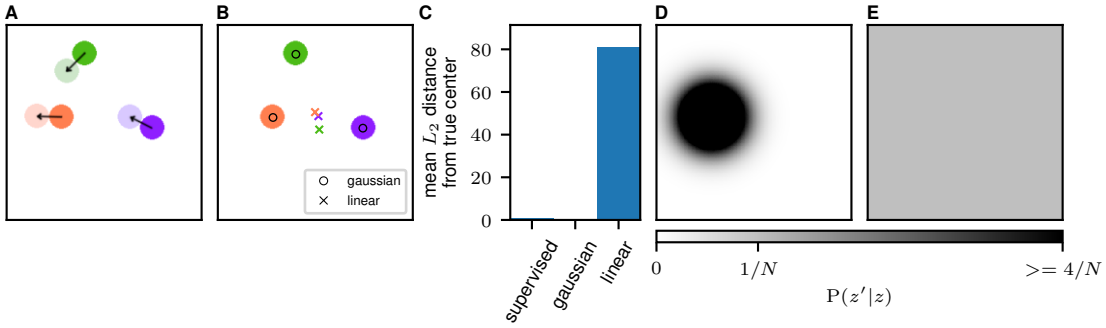

Figure 1: Results of the moving balls experiment. **A**) Example of the motion between consecutive frames. The balls move by a full diameter in a semi-random direction. **B**) Locations of the extracted ball centres, after supervised linear decoding. The standard InfoNCE setup fails to extract correct 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.

4 Experimental results

Our primary results are theoretical: in connecting the log-Bayesian model evidence and mutual information, and in showing that the InfoNCE objective with a restricted choice of f makes more sense as a bound on the log-Bayesian model evidence than on the MI. At the same time, our approach encourages a different way of thinking about how to set up contrastive SSL methods, in terms of Bayesian priors. As an example, we considered a task in which the goal was to extract the locations of three moving balls, based on videos of these balls bouncing around in a square (Fig. 1A; Appendix A).

Naively applying the usual InfoNCE setup (using Eq. 10), 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. (10) 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,

$$\mathrm{P}_{\mathrm{InfoNCE}}\left(z'|z\right) = \frac{1}{Z}\,\mathrm{Q}\left(z'\right) f(z,z') \propto \exp\left(z^T W z'\right) \tag{40}$$

(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,

$$\mathrm{P}\left(z'|z\right) = \mathcal{N}\left(z'; z, L^2\right). \tag{41}$$

This corresponds to a Gaussian RBF form for f,

$$f(z,z') = \exp\left(-\tfrac{1}{2L^2}(z-z')^2\right). \tag{42}$$

where L is a learned lengthscale. Critically, this choice of f is natural and obvious if we take a probabilistic generative view of the problem. In contrast, if our goal was to maximize information, then the most appropriate choice would be an arbitrarily flexible f.

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. In addition, their approach requires four restrictive assumptions. First, they assume deterministic encoder, e.g $\mathrm{Q}(z|x) = \delta(z - g(x))$. 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, $\mathrm{Q}(z|x)$ and $\mathrm{Q}(z'|x')$. 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. This is a particularly problematic assumption as practical encoders commonly used in contrastive SSL are not invertible. 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 CSSVAE (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 developed a new family of contrastive VAE, CSSVAEs. The CSSVAE ELBO is equal to the log-Bayesian model evidence for a deterministic encoder. For one choice of prior, the CSSVAE ELBO is equal to the mutual information, and with a different choice of prior, the CSSVAE ELBO is equal to the infinite-sample InfoNCE objective (up to constants). In contrast, the infinite-sample InfoNCE forms only a loose bound on the true MI, which would argue that the InfoNCE objective might be better motivated as the CSSVAE ELBO. As such, we unify contrastive semi-supervised learning with generative self-supervised learning (or unsupervised learning). Finally, 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.

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.

Alexander Alemi, Ben Poole, Ian Fischer, Joshua Dillon, Rif A Saurous, and Kevin Murphy. Fixing a broken elbo. In *International Conference on Machine Learning*, pp. 159–168. PMLR, 2018.

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.

William I Walker, Hugo Soulat, Changmin Yu, and Maneesh Sahani. Unsupervised representational learning with recognition-parametrised probabilistic models. *arXiv preprint arXiv:2209.05661*, 2022.

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.

Shengjia Zhao, Jiaming Song, and Stefano Ermon. The information autoencoding family: A lagrangian perspective on latent variable generative models. *arXiv preprint arXiv:1806.06514*, 2018.

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.

A Experimental details

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.

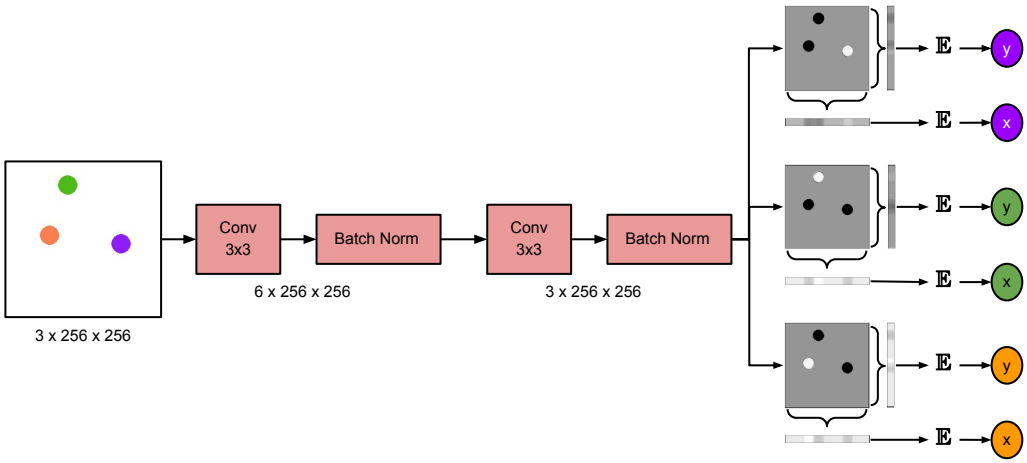

Figure 2: Architecture of the encoder neural network. The first of the two 3x3 convolutional layers outputs 6 feature maps and uses a ReLU activation. The second convolution outputs 3 feature maps and applies a sigmoid activation. For each of these 3 maps, we extract their centre of mass. This is done 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.

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.