

A Probabilistic Model to explain Self-Supervised Representation Learning

Anonymous Authors¹

Abstract

Self-supervised learning (SSL) learns representations by leveraging an auxiliary unsupervised task, such as classifying semantically related samples, e.g. different data augmentations or modalities. Of the many approaches to SSL, contrastive methods, e.g. SimCLR, CLIP and VicREG, have gained attention for learning representations that achieve downstream performance close to that of supervised learning. However, a theoretical understanding of the mechanism behind these methods eludes. We propose a generative latent variable model for the data and show that several families of discriminative self-supervised algorithms, including contrastive methods, approximately induce its latent structure over representations, providing a unifying theoretical framework. We also justify links to mutual information and the use of a projection head. Fitting our model generatively, as *SimVAE*, improves performance over previous VAE methods on common benchmarks (e.g. FashionMNIST, CIFAR10, CelebA), narrows the gap to discriminative methods on *content* classification and, as our analysis predicts, outperforms them where *style* information is required, taking a step toward task-agnostic representations.

1. Introduction

In self-supervised learning (SSL), a model is trained on an auxiliary task without class labels, and learns representations of the data in the process. Of the many approaches to SSL (Ericsson et al., 2022; Balestrieri et al., 2023), recent contrastive methods, such as InfoNCE (Oord et al., 2018), SimCLR (Chen et al., 2020), DINO (Caron et al., 2021) and CLIP (Radford et al., 2021), have gained attention for their remarkable performance on downstream tasks, approaching that of supervised learning. These methods exploit seman-

tically related observations, such as different parts (Oord et al., 2018; Mikolov et al., 2013), augmentations (Chen et al., 2020; Misra & Maaten, 2020), or modalities/views (Baevski et al., 2020; Radford et al., 2021; Arandjelovic & Zisserman, 2017) of the data. There is significant interest in understanding self-supervised methods (e.g. Wang & Isola, 2020; Zimmermann et al., 2021; Tian, 2022; Ben-Shaul et al., 2023), but a general mathematical mechanism justifying their performance remains unclear. At the same time, theoretically principled representations can be learned *generatively* by fitting a latent variable model by variational inference (e.g. Kingma & Welling, 2014), but they typically under-perform recent (*discriminative*) SSL methods.

To address this, we draw an equivalence between discriminative and generative representation learning that treats representations as latent variables and an encoder $f : \mathcal{X} \rightarrow \mathcal{Z}$ (mapping data x to representations $z = f(x)$) as a special case of the posterior $p(z|x)$. We then propose a generative model for self-supervised learning (Fig. 1) and by considering its evidence lower bound (ELBO), show that families of *discriminative* SSL methods, including the InfoNCE objective adopted by several algorithms (e.g. Chen et al., 2020; Radford et al., 2021), induce similar latent structure. Notably, the common perception that self-supervised learning objectives “pull together” representations of semantically related data and “push apart” others (e.g. Wang & Isola, 2020) is explained from first principles by this model: the prior pulls and the reconstruction term pushes.

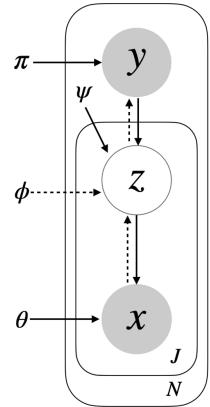


Figure 1: Generative model for SSL (J semantically related samples, parameters explained in Eq. 4)

Under the proposed model, representations (z) of semantically related data (x) form clusters conditioned on shared latent *content* (y), with variation in $z|y$ considered *style*. Our analysis predicts that these clusters form under previous *discriminative* SSL methods, but “collapse”, losing *style* information that distinguishes related samples, e.g. colour, orientation. Any information loss may affect some downstream tasks more than others, limiting the generality of

¹Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

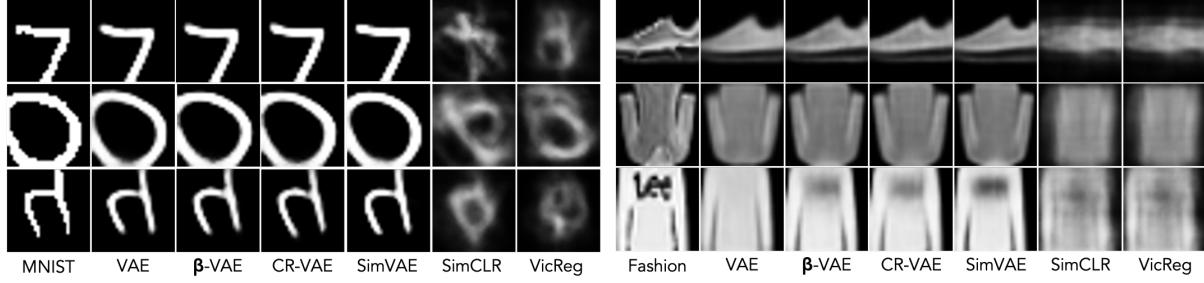


Figure 2: Assessing the information in representations: original images (left cols) and reconstructions from representations learned by generative unsupervised learning (VAE, β -VAE, CR-VAE), generative SSL (our SimVAE) and discriminative SSL (SimCLR, VicREG) on MNIST (*l*), Fashion MNIST (*r*). Discriminative methods lose style information (e.g. orientation).

representations. To verify this, we fit the latent variable model *generatively* by maximising the ELBO, termed *SimVAE*,¹ to explicitly induce the proposed latent structure.

While training a generative model is more challenging than training a discriminative one in general, our results show that SimVAE is competitive with or outperforms popular discriminative methods at downstream classification on common benchmark datasets (MNIST, FashionMNIST and CelebA). On more complex data (CIFAR10), SimVAE is less competitive for *content* classification but, as predicted by our analysis above, consistently outperforms discriminative methods on tasks requiring *style* information. Notably, SimVAE significantly outperforms other VAE-based generative models on all tasks. These results provide empirical support for the generative model as a mathematical basis for SSL; and suggest that generative representation learning is promising if distributions can be well modelled, particularly given its added benefits of uncertainty estimation from the posterior, a means to generate synthetic samples and to qualitatively assess the information captured by inspecting reconstructions and naturally model groups (Fig. 2).

To summarise our main contributions:

- we propose a latent variable model that underpins and unifies popular families of self-supervised learning algorithms, including contrastive methods (Fig. 1);
- we show that this model underpins the notion that SSL “pulls together”/“pushes apart” representations, rationalises the link to mutual information and justifies the common use of a projection head (§3);
- we fit the latent variable model generatively, termed SimVAE, and show clear improvement (>15% on CIFAR10) on benchmark (*content*) classification tasks over previous generative (VAE-based) methods, including one tailored to SSL (Sinha & Dieng, 2021) (§5); and
- we show that SimVAE captures more *style* information than discriminative methods as our analysis predicts (§5).

¹So-named as it encodes the latent structure of SimCLR (Chen et al., 2020) in the prior of a VAE (Kingma & Welling, 2014).

2. Background and Related Work

Representation Learning aims to learn an encoder $f: \mathcal{X} \rightarrow \mathcal{Z}$ that maps data $x \in \mathcal{X}$ to (often lower-dimensional) representations $z = f(x) \in \mathcal{Z}$ that perform well on downstream tasks. Representation learning is not “well defined” in the sense that downstream tasks can be arbitrary and a representation that performs well on one task may perform poorly on another (Zhang et al., 2022). For instance, unsupervised image representations are commonly evaluated by predicting semantic class labels, but the downstream task could be to detect lighting, position or orientation, which a representation useful for class prediction may not capture. This suggests that *general-purpose* unsupervised representations should capture as much information about the data as possible. Recent works support this by evaluating on a variety of downstream tasks (e.g. Balažević et al., 2023).

Self-Supervised Learning (SSL) learns representations by leveraging an auxiliary task. The many approaches to SSL can be categorised in various ways (e.g. Balestrieri et al., 2023). We define those we focus on as follows:

Instance Discrimination (Dosovitskiy et al., 2014; Wu et al., 2018) treats each sample x_i , along with any augmentations, as a distinct class i . A softmax classifier is trained to predict the “class” and encoder outputs are used as representations.

Latent Clustering performs clustering on representations. Song et al. (2013); Xie et al. (2016); Yang et al. (2017) apply K-means or similar to the hidden layer of a standard auto-encoder. DeepCluster (Caron et al., 2020) iteratively clusters ResNet representations by K-means, and predicts the cluster assignments as “pseudo-labels”. DINO (Caron et al., 2021), a transformer-based model, can be interpreted as clustering in the latent space (Balestrieri et al., 2023).

Contrastive Learning encourages representations of semantically related data (positive samples) to be “close” in contrast to those sampled at random (negative samples). Early SSL approaches include energy-based models (Chopra et al., 2005; Hadsell et al., 2006); and word2vec (Mikolov et al., 2013) that predicts co-occurring words, thereby capturing their pointwise mutual information (PMI) in its embeddings (Levy & Goldberg, 2014; Allen & Hospedales, 2019).

InfoNCE (Oord et al., 2018; Sohn, 2016) extends word2vec to other data domains. Its widely used objective, for a *positive* pair of semantically related samples (x, x^+) and randomly selected *negative* samples $X^- = \{x_k^-\}_{k=1}^K$, is defined

$$\mathcal{L}_{\text{INCE}}(x, x^+, X^-) = \log \frac{e^{\text{sim}(z, z^+)}}{\sum_{x' \in \{x^+\} \cup X^-} e^{\text{sim}(z, z')}} , \quad (1)$$

where $\text{sim}(\cdot, \cdot)$ is a similarity function, e.g. dot product. Eq. 1 is minimised if $\text{sim}(z, z') = \text{PMI}(x, x') + c$, for constant c (Oord et al., 2018). Many works build on InfoNCE, e.g. SimCLR (Chen et al., 2020) uses synthetic augmentations and CLIP (Radford et al., 2021) uses different modalities as positive samples; DIM (Hjelm et al., 2019) takes other encoder parameters as representations; and MoCo (He et al., 2020), BYOL (Grill et al., 2020) and VicREG (Bardes et al., 2022) find alternative strategies to negative sampling to prevent representations from collapsing.

We do not address all SSL algorithms in this work, in particular those with regression-based auxiliary tasks such as reconstructing data from perturbed versions (e.g. He et al., 2022; Xie et al., 2022); or predicting perturbations, such as rotation angle (e.g. Gidaris et al., 2018). For clarity, we refer to the methods we address as **Predictive SSL**.

Variational Auto-Encoder (VAE): For a generative model $\mathbf{z} \rightarrow \mathbf{x}$, parameters θ of $p_\theta(x) = \int_z p_\theta(x|z)p_\theta(z)$ can be learned by maximising the *evidence lower bound* (ELBO)

$$\mathbb{E}_x [\log p_\theta(x)] \geq \mathbb{E}_x \left[\int_z q_\phi(z|x) \log \frac{p_\theta(x|z)p(z)}{q_\phi(z|x)} \right] , \quad (2)$$

where $q_\phi(z|x)$ learns to approximate the posterior $p_\theta(z|x)$. Latent variables z can be used as representations (see §3.1). A VAE (Kingma & Welling, 2014) maximises the ELBO, with p_θ, q_ϕ modelled as Gaussians parameterised by neural networks. A β -VAE (Higgins et al., 2017) weights ELBO terms to increase disentanglement of latent factors. A CR-VAE (Sinha & Dieng, 2021) considers semantically related samples through an additional regularisation term. Further relevant VAE variants are summarised in Appendix A.1.

Variational Classification (VC): Dhuliawala et al. (2023) define a latent variable model for classification of labels y , $p(y|x) = \int_z q(z|x) \frac{p(z|y)p(y)}{p(z)}$, that generalises softmax neural network classifiers, interpreting the encoder as parameterising $q(z|x)$; and the softmax layer as encoding $p(y|z)$ by Bayes' rule (**VC-A**). For data in continuous domains \mathcal{X} , e.g. images, $q(z|x)$ of a softmax classifier can overfit to a single delta-distribution for all samples of a class, meaning representations of a class collapse together (termed neural collapse by Popyan et al., 2020).² This loses semantic and probabilistic information distinguishing class samples and harms properties such as calibration and robustness (**VC-B**).

²In practice, constraints such as l_2 regularisation and early stopping arbitrarily restrict the described optimum being attained.

Prior theoretical analysis: There has been considerable interest in understanding the mathematical mechanism behind self-supervised learning (Arora et al., 2019; Tsai et al., 2020; Wang & Isola, 2020; Zimmermann et al., 2021; Lee et al., 2021; Von Kügelgen et al., 2021; HaoChen et al., 2021; Wang et al., 2021; Saunshi et al., 2022; Tian, 2022; Sansone & Manhaeve, 2022; Nakamura et al., 2023; Shwartz-Ziv et al., 2023; Ben-Shaul et al., 2023), as summarised by HaoChen et al. (2021) and Saunshi et al. (2022). A thread of works (Arora et al., 2019; Tosh et al., 2021; Lee et al., 2021; HaoChen et al., 2021; Saunshi et al., 2022) aims to prove that auxiliary task performance translates to downstream classification accuracy, but Saunshi et al. (2022) show this can not apply to typical datasets, and model architecture must be considered. Several works propose an information theoretic basis for SSL (Hjelm et al., 2019; Bachman et al., 2019; Tsai et al., 2020; Shwartz-Ziv et al., 2023), e.g. maximising mutual information between representations, but Tschannen et al. (2020); McAllester & Stratos (2020); Tosh et al. (2021) raise doubts with this. We show that the relationship to mutual information is justified more fundamentally by our latent variable model (Fig. 1).

3. Self-Supervised Representation Learning

We consider datasets $X = \bigcup_{i=1}^N \mathbf{x}_i$, where each subset $\mathbf{x}_i = \{x_i^j\}_j$ contains semantically related data $x_i^j \in \mathcal{X}$, considered to share *semantic content* and vary in *style*.³ For example, \mathbf{x}_i may contain different augmentations, modalities or snippets of a given observation (e.g. an image of a triangle and its mirror image, where the shape is content and orientation is style). We assume a generative process where $y_i \sim p(y)$ is sampled, determining semantic content, then $\mathbf{z}_i = \{z_i^j\}_j$ are sampled conditionally independently $z_i^j \sim p(z|y_i)$, reflecting the same content but varying in style. Finally data points $x_i^j \in \mathbf{x}_i$ are sampled $x_i^j \sim p(x|z_i^j)$. Fig. 1 shows the generative model for SSL, under which

$$p(\mathbf{x}_i|y_i) = \int_{\mathbf{z}_i} \left(\prod_j p(x_i^j|z_i^j) \right) \left(\prod_j p(z_i^j|y_i) \right) . \quad (3)$$

Note that y_i is not semantically meaningful, only an observe indicator reflecting that $x_i^j \in \mathbf{x}_i$ are related. If distributions in Eq. 3 are modelled parametrically, their parameters can be learned by maximising **ELBO_{SSL}**, a lower bound on $\log p(\mathbf{x}|y)$ for J semantically related data $\mathbf{x} = \{x^j\}_{j=1}^J \subseteq \mathbf{x}_i$,

$$\mathbb{E}_{\mathbf{x}, y} \left[\sum_{j=1}^J \int_{z^j} q_\phi(z^j|x^j) (\log p_\theta(x^j|z^j) - \log q_\phi(z^j|x^j)) + \int_{\mathbf{z}} q_\phi(\mathbf{z}|\mathbf{x}) \log p_\psi(\mathbf{z}|y) \right] , \quad (4)$$

where the approximate posterior is assumed to factorise as $q(\mathbf{z}|\mathbf{x}) \approx \prod q(z^j|x^j)$.⁴ A derivation of Eq. 4 is given in

³ $|\mathbf{x}_i|$ may vary and domains $\mathcal{X}^j \ni x_i^j$ can differ with modality.

⁴Expected to be reasonable for z^j that carry high information

165 Appendix A.2. As in the generic ELBO (Eq. 2), ELBO_{SSL}
166 comprises: a **reconstruction term**, the **approximate posterior**
167 **entropy** and the (now conditional) **log prior** over all $z^j \in \mathbf{z}$.

168 Note that if all $p(z|y_i)$ are concentrated relative to $p(z)$, i.e.
169 $\text{Var}[z|y_i] \ll \text{Var}[z]$, then latent variables of semantically re-
170 lated data $z \in \mathbf{z}_i$ are clustered, or closer on average than
171 those of random samples – as SSL representations are often
172 described. Indeed, our claim is that the generative model in
173 Fig. 1, fitted by maximising Eq. 4, underpins discriminative
174 self-supervised learning algorithms, including instance dis-
175 crimination, deep clustering and contrastive methods (pre-
176 dictive SSL, §2). We first clearly define how discriminative
177 and generative methods relate and, from that, show that the
178 loss functions of discriminative methods reflect Eq. 4.

3.1. Discriminative vs Generative Learning

180 Representation learning aims to train an encoder $f: \mathcal{X} \rightarrow \mathcal{Z}$
181 so that representations $z = f(x)$ are arranged usefully for
182 downstream tasks, e.g. clustered so that a class of interest is
183 easily identified. The predictive self-supervised approaches
184 we consider (§2) train f under a loss function without a gen-
185 erative model, so are *discriminative*. Representations can
186 also be learned *generatively*: if latent variables z , under a
187 generative model $z \rightarrow x$, determine underlying properties of
188 the data x , then the posterior $p(z|x)$ is expected to estimate
189 semantic properties of x , so approximating it (by $q(z|x)$
190 parameterised by f) gives semantically meaningful rep-
191 resentations. These approaches are actually closely related:

192 **Gen-Discrim Equivalence:** A deterministic encoder f
193 can be viewed as a concentrated posterior distribution
194 $p_f(z|x) = \delta_{z-f(x)}$. Together with $p(x)$, this defines the joint
195 $p_f(x, z) = p_f(z|x)p(x)$ and marginal $p_f(z) = \int_x p_f(x, z)$.
196 Thus, an encoder can be trained to map the data distribu-
197 tion to a “useful” target distribution (or distribution family)
198 $p^*(z)$:

- *generatively* by optimising the ELBO with prior $p^*(z)$; or
- *discriminatively* if there exists a loss function that when optimised induces $p_f(z) \approx p^*(z)$.

200 The takeaway is that both approaches have an equivalent
201 aim. In particular, our claim can now be restated as: pre-
202 dictive SSL methods (§2) induce a distribution $p_f(z)$ that
203 approximates $p(z)$ under the graphical model in Fig. 1.

3.2. Inducing $p^*(z)$: discriminatively or generatively?

204 If an encoder can be trained to give a useful distribution
205 $p^*(z)$ over representations discriminatively or generatively
206 then, all else equal, the former may seem preferable to avoid
207 modelling $p(x|z)$. However, while the ELBO gives a prin-
208 cipled generative objective to learn representations that fit
209 $p^*(z)$, a principled discriminative loss that minimises with

210 w.r.t. x^j , such that observing related x^k or its representation z^k pro-
211 vides negligible extra information, i.e. $p(z^j|x^j, x^k) \approx p(z^j|x^j)$.

212 $p_f(z) = p^*(z)$ is less clear. We therefore consider the effect
213 each ELBO term has on the optimal posterior/encoder to
214 understand what a discriminative, or “ $p(x|z)$ -free”, objec-
215 tive needs to achieve to induce a given distribution over
216 representations.

217 **Entropy:** discriminative methods use a deterministic map-
218 ping $q(z|x) = \delta_{z-f(x)}$. We treat this as a posterior with very
219 small but non-zero *fixed* variance, hence $\mathcal{H}[q]$ is a constant
220 and dropped from the SSL objective.

221 **Prior:** this term is optimal w.r.t. q iff related samples $x \in \mathbf{x}_i$
222 each map to a mode of $p(z|y_i)$. For uni-modal $p(z|y_i)$ this
223 means all representations $z \in \mathbf{z}_i$ “collapse” to a point, losing
224 information distinguishing $x \in \mathbf{x}_i$.⁵

225 **Reconstruction:** this term is maximised w.r.t. q iff each x
226 maps to a distinct representation z that $p(x|z)$ maps back to
227 x . This requires all z to be distinct and all information to be
228 retained, countering the prior to **avoid collapse**.

229 In summary, we consider a discriminative objective to emu-
230 late the ELBO if it combines the log prior with a $p(x|z)$ -free
231 substitute for the reconstruction term (denoted $\text{Rec}(\cdot)$) that
232 should, *inter alia*, avoid representation collapse both be-
233 tween and within subsets \mathbf{z}_i , i.e.

$$\int_z q(z|x) (\log p(z|y) + \text{Rec}(z, x)). \quad (5)$$

234 From the analysis above, maximising Eq. 5 can be seen to
235 fit the notion that SSL objectives “pull together” representa-
236 tions of related data and “push apart” others (e.g. Wang &
237 Isola, 2020), the prior pulls (representations are attracted to
238 modes) and the reconstruction pushes or “avoids collapse”
239 (representations are made distinct). Our claim is that Eq. 5,
240 and by extension ELBO_{SSL} , is the rationale for that notion,
241 i.e. SSL objectives of the form of Eq. 5 emulate ELBO_{SSL}
242 and so approximately induce the latent structure $p(z)$ in
243 Fig. 1. The *true* reconstruction term not only avoids col-
244 lapsed but is also integral to modelling the data distribution
245 and posterior to give meaningful representations. As we
246 shall see, approximation by $\text{Rec}(\cdot)$ can impact representa-
247 tions and their performance in downstream tasks.

3.3. Discriminative Self-Supervised Learning

248 We now consider examples of instance discrimination, la-
249 tent clustering and contrastive SSL methods (§2) and their
250 relationship to the latent variable model in Fig. 1 via Eq. 5.

251 **Instance Discrimination (ID)** (Dosovitskiy et al., 2014;
252 Wu et al., 2018) trains a softmax classifier on sample-index
253 pairs $\{(x_i^j, y_i = i)\}_{i,j}$, which VC-A (Dhuliawala et al., 2023,
254 §2) interprets, under Fig. 1, as maximising the RHS of:

$$\log p(y|x) \geq \int_z q_\phi(z|x) \log p(y|z) \quad (6)$$

255 ⁵Assuming classes \mathbf{x}_i are distinct, as is the case for empirical
256 self-supervised learning datasets of interest.

$$= \int_z q_\phi(z|x) (\log p(z|y) - \log p(z)) + c. \quad (7)$$

Eq. 7 matches Eq. 5 with $J=1$ and $\text{Rec}(\cdot) = \mathcal{H}[p(z)]$, the entropy of $p(z)$. Intuitively, maximising entropy helps avoid collapse, but under a softmax loss, while representations of distinct classes do spread apart, those of the same class $z \in \mathbf{z}_i$ collapse together (VC-B, Dhuliawala et al., 2023).

Deep Clustering (DC) (Caron et al., 2018) iteratively assigns temporary labels y_i to data \mathbf{x}_i by (K-means) clustering representations of a ResNet encoder and updating the encoder to predict those labels with a softmax head. While semantically related subsets \mathbf{x}_i are now defined by the ResNet’s “inductive bias”, the same loss is used as in ID (Eq. 7) and representation clusters $z \in \mathbf{z}_i$ collapse together.

Contrastive Learning stems from the InfoNCE objective (Eq 1). A variational lower bound under Fig. 1 (comparable to Eq. 6) is optimised when $\text{sim}(z, z') = \text{PMI}(z, z') + c$ (§2). Combining this, we see that InfoNCE maximises:

$$\begin{aligned} \mathcal{L}_{\text{NCE}}(x_i, X') &= \{x_i^+, x_{i_1}^-, \dots, x_{i_k}^-\} \\ &= \mathbb{E}_X \left[\int_z q_\phi(Z|X) \log \frac{\text{sim}(z_i, z_i^+)}{\sum_{z' \in Z'} \text{sim}(z_i, z')} \right] \\ &\stackrel{\text{opt}}{\rightarrow} \mathbb{E}_X \left[\int_Z q_\phi(Z|X) \log \frac{p(z_i^+|z_i)/p(z_i^+)}{\sum_{z' \in Z'} p(z'|z_i)/p(z')} \right] \\ &\leq \mathbb{E}_\mathbf{x} \left[\int_z q_\phi(\mathbf{z}|\mathbf{x}) \log \frac{p(z_i, z_i^+)}{p(z_i)p(z_i^+)} \right] + \log \frac{1}{k-1} \\ &\stackrel{k \rightarrow \infty}{\rightarrow} \mathbb{E}_\mathbf{x} \left[\int_z q_\phi(\mathbf{z}|\mathbf{x}) (\log p(\mathbf{z}|y_i) - \sum_j \log p(z_i^j)) \right] \quad (8) \end{aligned}$$

(A full derivation is given in Appendix A.3.) Thus the InfoNCE objective again approximates ELBO_{SSL} using entropy $\text{Rec}(\cdot) = \mathcal{H}[p(z)]$, as in Eq. 7 but with $J=2$.

Role of J : ID and DC consider $J=1$ sample x_i^j at a time and compute $p(z_i|y_i; \psi_i)$ from saved parameters ψ_i (which could be memory intensive). For $J \geq 2$ samples, one can estimate $p(\mathbf{z}_i|y_i) = \int_{\psi_i} p(\psi_i) \prod_j p(z_i^j|y_i; \psi_i) = s(\mathbf{z}_i)$, where ψ_i integrates out (see Appendix A.4 for an example). Since y_i now defines no parameters, joint distributions over $\{z_{i_r}\}_r$ depend only on whether $\{x_{i_r}\}_r$ are semantically related:

$$p(z_{i_1}, \dots, z_{i_k}) = \begin{cases} s(\mathbf{z}_i) & \text{if } i_r = i \ \forall r \ (\text{i.e. } x_{i_r} \in \mathbf{x}_i) \\ \prod_{r=1}^k p(z_{i_r}) & \text{if } i_r \neq i_s \ \forall r, s \end{cases}$$

InfoNCE implicitly applies a similar “trick” with $\text{sim}(\cdot, \cdot)$.

Mutual Information (MI): InfoNCE is known to optimise a lower bound on MI, $I(x, x') = \mathbb{E}[\log \frac{p(x, x')}{p(x)p(x')}]$ (Oord et al., 2018) and Eq. 8 shows it also lower bounds $I(z, z')$. As a result, maximising MI has at times been argued to underpin contrastive learning (Hjelm et al., 2019; Ozsoy et al., 2022). However, our analysis suggests that MI is not the fundamental explanation for contrastive learning,

rather MI arises from substituting the reconstruction term in ELBO_{SSL} . This is supported by Tschanen et al. (2020); McAllester & Stratos (2020); Poole et al. (2019) who show that better MI estimators do not give better representations, MI approximation is noisy and the InfoNCE estimator is arbitrarily upper-bounded. Further, for disjoint \mathbf{x}_i , pointwise mutual information (PMI) values span an *unbounded* range, $[-\infty, k]$, $k > 1$, yet are commonly modelled by the *bounded* cosine similarity $\text{sim}(z, z') = \frac{z^\top z'}{\|z\|\|z'\|} \in [-1, 1]$ (e.g. Chen et al., 2020), making MI estimation worse (see §A.5). In Appendix A.5, we show that this common constraint leads to representations comparable to those learned by softmax cross entropy (*cf* ID), but without class parameters ψ_i .⁶ Notably, representations of related data again collapse.

Summary: We have considered discriminative SSL methods from instance discrimination, latent clustering and contrastive learning, and shown that, despite considerable differences, each approximates ELBO_{SSL} and so approximately induces the prior of our model for SSL (Fig. 1). The number of SSL methods, even within the predictive subset we focus on, makes an exhaustive analysis infeasible. However, many methods adopt or approximate aspects analysed above. For example, SimCLR (Chen et al., 2020) and CLIP (Radford et al., 2021) use the InfoNCE objective. MoCo (He et al., 2020), BYOL (Grill et al., 2020) and VicREG (Bardes et al., 2022) replace negative sampling with a momentum encoder, stop gradients or (co-)variance terms, but nonetheless “pull together” representations of semantically related data modelling the prior; and “push apart” others, or “avoid collapse”, by those mechanisms that mimic the effect of reconstruction. DINO (Caron et al., 2021) assigns representations of semantically related data pairs to a common cluster as in DC (Balestrieri et al., 2023), but with $J=2$.

The predictive SSL methods analysed above each replace the reconstruction term of ELBO_{SSL} with entropy $\mathcal{H}[p(z)]$. As such, representations of semantically related data form distinct clusters, but those clusters collapse, **losing style information** since representations $z \in \mathbf{z}_i$ become indistinguishable (e.g. to a downstream classifier). Style information is important in many real-world tasks, e.g. detecting facial expression or voice sentiment; and is the focus of representation learning elsewhere (e.g. Higgins et al., 2017; Karras et al., 2019). Thus, counter to *general* representation learning, **discriminative SSL may over-fit to content-based tasks** (see Appendix A.6 for further discussion.)

Projection Head: Our analysis suggests a plausible explanation for the practice of adding layers, or a projection head, to an encoder and using its input as representations rather than its output z used in the loss function. Since the ELBO prevents both *inter-* and *intra-cluster* collapse, any approxi-

⁶We note that Wang & Isola (2020) reach a similar conclusion but less rigorously, e.g. the known PMI minimiser is not addressed.

mation should also. Near-final encoder layers are found to exhibit similar clustering to z , but with higher intra-cluster variance (Gupta et al., 2022), as also seen in supervised softmax classification (Wang et al., 2022). We conjecture that representations from near-final layers are preferable to z because their intra-cluster variance gives an overall distribution closer to that under Fig. 1. Conversely, the simple fact that *any* other representations outperform z suggests that discriminative objectives do not learn ideal representations, supporting our claim that they are an approximation.

3.4. Generative Self-Supervised Learning (SimVAE)

The proposed model for SSL (Fig. 1) has been shown to justify: (i) the training objectives of predictive SSL methods; (ii) the notion that SSL “pulls together”/“pushes apart” representations; (iii) the connection to mutual information; and (iv) the use of a projection head. We now aim to add empirical support by validating the following predictions: [H1] maximising ELBO_{SSL} achieves self-supervised learning, where distributions can be well modelled; [H2] maximising ELBO_{SSL} retains more style information than discriminative objectives; [H3] the generative model (Fig. 1) gives better representations than a range of VAE alternatives.

To test these hypotheses, we maximise ELBO_{SSL} (Eq. 4), effectively taking a generative approach to self-supervised learning, termed *SimVAE*. SimVAE can be considered a VAE with a mixture prior $p(\mathbf{z}) = \sum_y p(\mathbf{z}|y)p(y)$, where representations $z \in \mathbf{z}$ of semantically related samples are conditioned on the same y . We assume that $p(x|z)$ and $q(z|x)$ of Eq. 4 are Gaussians parameterised by neural networks, as in a standard VAE; that conditionals $p(\mathbf{z}|y = i; \psi_i) = \mathcal{N}(\mathbf{z}; \psi_i, \sigma^2)$ are Gaussian with small fixed variance σ^2 ; and $p(\psi)$ is uniform (over a suitable space). Integrating out ψ (as in Appendix A.4) gives:

$$p(\mathbf{z}|y) \propto \exp\left\{-\frac{1}{2\sigma^2} \sum_{z \in \mathbf{z}} (z - \bar{z})^2\right\}, \quad (9)$$

a Gaussian centred around the mean representation $\bar{z} = \frac{1}{J} \sum_j z_j$. Maximising this, as a component of ELBO_{SSL} , “pinches together” representations of semantically related samples. Whereas contrastive methods typically compare pairs of related representations (i.e. $J = 2$), Eq. 4 allows any number J to be processed. In practice a balance is struck between better approximating $p(\mathbf{z}|y)$ and preserving diversity in a mini-batch. Algorithm 1 in Appendix A.7.1 details the steps to optimise Eq. 4 under these assumptions.

4. Experimental Setup

Datasets and Evaluation Metrics We test our hypotheses by evaluating SimVAE representations on four benchmark datasets including two with natural images: MNIST (LeCun, 1998), FashionMNIST (Xiao et al., 2017), CelebA (Liu et al.,

2015) and CIFAR10 (Krizhevsky et al., 2009). We augment images following the SimCLR protocol (Chen et al., 2020) which includes cropping and flipping, and colour jitter for natural images. Frozen pre-trained representations are evaluated by (unsupervised) clustering under a gaussian mixture model and (supervised) training a non-linear MLP probe, a k-nearest neighbors (kNN) estimator (Cover & Hart, 1967) and a linear probe on classification tasks (Chen et al., 2020; Caron et al., 2020). Downstream performance is measured in terms of classification accuracy (Acc). Generative quality is evaluated by FID score (Heusel et al., 2017) and reconstruction error (see Appendix A.8). For further experimental details see Appendices A.7.2, A.7.3, A.7.6 and A.8.

Baselines methods We compare SimVAE to other VAE-based models including the vanilla VAE (Kingma & Welling, 2014), β -VAE (Higgins et al., 2017) and CR-VAE (Sinha & Dieng, 2021), as well as to state-of-the-art self-supervised discriminative methods including SimCLR (Chen et al., 2020), VicREG (Bardes et al., 2022), and MoCo (He et al., 2020). As a lower bound, we also provide results for a randomly initialized encoder. For fair comparison, the augmentation strategy, representation dimensionality, batch size, and encoder-decoder architectures are invariant across methods. To enable a qualitative comparison of representations, decoder networks were trained for each discriminative baseline on top of frozen representations using the reconstruction error. See Appendices A.7.4 and A.7.5 for further details on training baselines and decoder models.

Implementation Details We use MLP and Resnet18 (He et al., 2016) network architectures for simple and natural image datasets respectively. The dimension of representations z is set to 10 for MNIST, FashionMNIST, and 64 for CelebA and CIFAR10 datasets. For all generative approaches, we adopt Gaussian posteriors, $q(z|x)$, priors, $p(z)$, and likelihoods, $p(x|z)$, with diagonal covariance matrices (Kingma & Welling, 2014). For SimVAE, we adopt Gaussian $p(z|y)$ as described in §3.4. SimVAE allows J related observations to be simultaneously incorporated. Based on early experiments, we fix the number of augmentations to $J = 10$ (see Fig. 7 for an ablation). Ablations were performed for all sensitive hyperparameters for each method and parameter values were selected based on the best average MLP Acc across datasets. Further details regarding hyperparameters and computational resources can be found in Appendices A.7.4 and A.7.5.

5. Results

Content classification: Table 1 reports the downstream classification across datasets using benchmark class labels (i.e., content). Tables 1 and 5 show that SimVAE is comparable to or outperforms *generative* baselines on supervised and unsupervised learning metrics on simple datasets. On nat-

		Acc-MP	Acc-KNN			Acc-MP	Acc-KNN
MNIST	Random	38.1 \pm 3.8	46.1 \pm 2.5	CelebA	Random	83.5 \pm 1.0	80.0 \pm 0.9
	SimCLR	97.2 \pm 0.0	97.2 \pm 0.1		SimCLR	93.7 \pm 0.4	91.6 \pm 0.3
	MoCo	94.6 \pm 0.4	94.6 \pm 0.3		MoCo	89.7 \pm 1.0	88.6 \pm 1.0
	VicREG	96.7 \pm 0.0	97.0 \pm 0.0		VicREG	94.7 \pm 0.1	92.7 \pm 0.4
	VAE	97.8 \pm 0.1	98.0 \pm 0.1		VAE	89.0 \pm 0.5	86.9 \pm 0.7
	β -VAE	98.0 \pm 0.0	98.3 \pm 0.0		β -VAE	93.4 \pm 0.4	92.0 \pm 0.1
	CR-VAE	97.7 \pm 0.0	98.0 \pm 0.0		CR-VAE	93.1 \pm 0.4	91.6 \pm 0.6
	SimVAE	98.4 \pm 0.0	98.5 \pm 0.0		SimVAE	95.6 \pm 0.4	93.2 \pm 0.1
	Random	49.8 \pm 0.8	66.5 \pm 0.4		Random	16.3 \pm 0.4	13.1 \pm 0.6
	SimCLR	74.9 \pm 0.2	76.0 \pm 0.1		SimCLR	67.4 \pm 0.1	64.0 \pm 0.0
FashionMNIST	MoCo	71.2 \pm 0.1	76.9 \pm 0.2		MoCo	56.4 \pm 1.6	54.0 \pm 2.0
	VicREG	73.2 \pm 0.1	76.0 \pm 0.1		VicREG	69.7 \pm 0.0	68.3 \pm 0.0
	VAE	80.2 \pm 0.3	83.7 \pm 0.2		VAE	30.3 \pm 0.4	25.6 \pm 0.5
	β -VAE	82.2 \pm 0.1	86.1 \pm 0.0		β -VAE	36.6 \pm 0.1	28.5 \pm 0.1
	CR-VAE	82.6 \pm 0.0	86.4 \pm 0.0		CR-VAE	36.8 \pm 0.0	28.1 \pm 0.1
	SimVAE	82.1 \pm 0.0	86.5 \pm 0.0		SimVAE	51.8 \pm 0.0	47.1 \pm 0.0
	Random	49.8 \pm 0.8	66.5 \pm 0.4		Random	16.3 \pm 0.4	13.1 \pm 0.6
	SimCLR	74.9 \pm 0.2	76.0 \pm 0.1		SimCLR	67.4 \pm 0.1	64.0 \pm 0.0
	MoCo	71.2 \pm 0.1	76.9 \pm 0.2		MoCo	56.4 \pm 1.6	54.0 \pm 2.0
	VicREG	73.2 \pm 0.1	76.0 \pm 0.1		VicREG	69.7 \pm 0.0	68.3 \pm 0.0

Table 1: Top-1% self-supervised Acc (\uparrow) for MNIST, FashionMNIST, CIFAR10, and CelebA (gender classification) using a MLP probe (MP) and k-Nearest Neighbors (KNN) classification methods; We report mean and standard errors over three runs; Bold indicate best scores in each method class: generative (teal), discriminative methods (red).

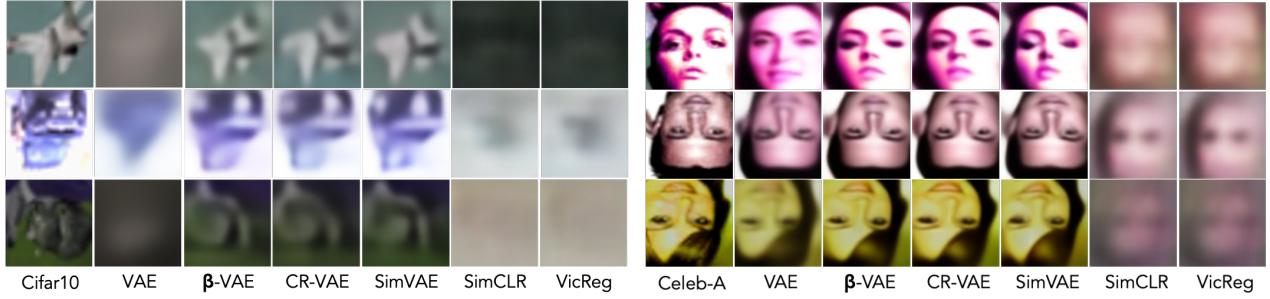


Figure 3: Assessing the information in representations: original images (left cols) and reconstructions from representations learned by generative unsupervised learning (VAE, β -VAE, CR-VAE), generative SSL (our SimVAE) and discriminative SSL (SimCLR, VicREG) on Cifar10 (*l*), CelebA (*r*). Discriminative methods lose style information (e.g. orientation/colour).

ural image datasets, we observe a significant improvement in performance over all VAE methods including the self-supervised approach, CR-VAE (+2.5% for CelebA, +15% for CIFAR10), supporting **H3**.

Table 1 also shows that representations learned by SimVAE materially reduces the performance gap (Δ) with respect to representations learned by popular *discriminative* methods, by approximately half for CIFAR10 ($\Delta = 32.8\% \rightarrow \Delta = 17.6\%$). SimVAE outperforms all baselines on the CelebA dataset. The results in Table 1 thus empirically support **H1** and demonstrate that SimVAE achieves self-supervised, particularly where distributions can be well-modelled.

Style classification: We further investigate the downstream performance on style-related features (e.g., colour, position, and orientation) using the CelebA multi-attribute dataset. Figure 4 (*left*) shows that SimVAE outperforms both generative and discriminative baselines on predicting an attribute dependant on style (hair colour). We also explore image reconstructions to gain qualitative insights into the information captured by the representation. Figures 2 and 3

shows a loss of orientation and colour information in representations learned by discriminative methods across four datasets. These findings support our hypothesis (**H2**) that discriminative methods lose *style* information, which may be important in some downstream tasks, and confirms the ability of generative methods to retain both content and style information (Figure 4, *middle*). Importantly, Figure 4 (*right*) shows that SimVAE learns more task-agnostic representations, surpassing all generative and discriminative baselines on average across the prediction of all (20) CelebA attributes. An overview of downstream performances for individual CelebA features are reported in Appendix A.8.

Image Generation: While generative quality is not relevant to our main hypotheses, out of interest and perhaps as a future benchmark, we show randomly generated SimVAE images and generative quality metrics in Appendix A.8. We observe small but significant FID score and reconstruction error improvements relative to previous VAE methods on MNIST, FashionMNIST, CelebA and CIFAR10 datasets.

CelebA	Acc-MP
Random	56.2 ± 0.8
SimCLR	52.0 ± 0.2
VicREG	52.8 ± 0.4
VAE	64.4 ± 0.3
β -VAE	66.4 ± 0.4
CR-VAE	66.2 ± 0.4
SimVAE	67.5 ± 0.3

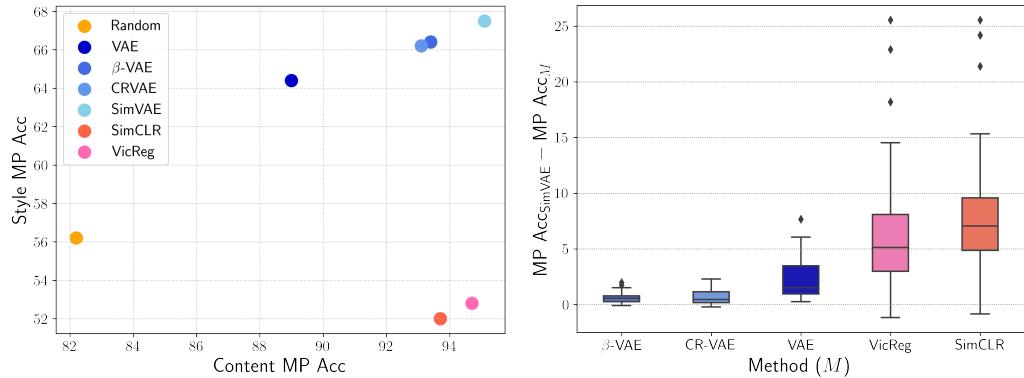


Figure 4: Style prediction on CelebA (Acc-MP \uparrow): (left) hair colour prediction (mean and standard error over three runs, best overall results indicated in bold); (middle) content vs. style prediction (gender vs hair colour), best performance in top-right; (right) Performance increase of SimVAE relative to baselines across all 20 CelebA attributes.

6. Discussion & Conclusion

Representations learned by SSL achieve impressive performance on downstream tasks, nearing that of fully supervised learning (e.g. Chen et al., 2020; Caron et al., 2021; Bardes et al., 2022), and can transfer well between datasets (Ericsson et al., 2022). This performance has spurred considerable interest in understanding the theoretical mechanism behind SSL (§2). Several works aim to explain how auxiliary task performance transfers to a downstream task but have been disproved for typical datasets (Saunshi et al., 2022). It has also been argued that SSL relies on maximising mutual information between views/augmentations (Hjelm et al., 2019) or pulling representations of semantically similar data closer than those of random samples (Wang & Isola, 2020).

We propose a hierarchical latent variable model for predictive SSL (Fig. 1) in which representations (z) of semantically related data (x) are conditioned on latent variable (y) determining semantic *content*, and $z \sim p(z|y)$ governs *style*. We show that our model justifies loss functions of predictive SSL methods, including contrastive learning, by reference to its ELBO; and underpins the common notion that SSL ‘pulls together’ representations of semantically related data and ‘pushes apart’ others. Predictive SSL methods are seen to maximise entropy $\mathcal{H}[p(z)]$ in place of the ELBO’s reconstruction term, thereby maximising mutual information between representations. This seems intuitive appealing, but in fact causes representations of semantically related data to collapse together and lose information about the data, reducing the generality of representations. Our analysis also justifies the common use of a projection head (§3.3), which itself suggests caution in (over-)analysing representations that are ultimately discarded.

We show that fitting the model generatively, as SimVAE, learns representations that perform comparably, or demonstrably reduce the performance gap, to discriminative meth-

ods at *content* prediction, while outperforming where *style* information is required, taking a step towards task-agnostic representations. SimVAE outperforms previous generative VAE-based approaches, including CR-VAE tailored to SSL.

Regarding limitations, a full explanation of SSL should connect learning an auxiliary task to downstream task performance. This requires understanding: (i) what the auxiliary task requires of the encoder; (ii) how the encoder achieves it; and (iii) how (i-ii) relate to the downstream task. Our work tackles (i). It is unclear if we currently have sufficient understanding of neural networks to resolve (ii), required for (iii) (Saunshi et al., 2022). However, continuity, smoothness and inductive bias of the encoder seem key: as representations of semantically related data come together under the auxiliary task, representations of *other related samples*, ‘close’ in \mathcal{X} , are brought together in \mathcal{Z} . By addressing (i), we hope to facilitate future work in (ii) and (iii).

Learning representations generatively is currently challenging, but we believe the connection drawn to popular SSL methods in our analysis and supported empirically justifies further research. Particularly given the prospect of uncertainty estimation from the posterior, of generating novel samples and of learning task-agnostic representations that *disentangle* style information rather than lose it (Higgins et al., 2017). Preliminary analysis (c.f., Figure 6 in Appendix A.8) also suggests that SimVAE is less sensitive to hyperparameters, notably in the augmentation strategy. Even while generative representation learning remains a challenge, understanding the mechanism behind SSL offers a guide for designing and interpreting future discriminative methods. Our latent variable model also recasts SSL in line with other latent variable paradigms, such as unsupervised VAEs (Kingma & Welling, 2014) and supervised Variational Classification (Dhuliawala et al., 2023), which may facilitate a principled unified learning regime.

440 7. Broader Impact

441 This paper presents work whose goal is to advance the field
 442 of Machine Learning. There are many potential societal
 443 consequences of our work, none which we feel must be
 444 specifically highlighted here. Specifically, we focus on a
 445 theoretical understanding of existing self-supervised learning
 446 methods with the potential impact of improving their
 447 interpretability and future algorithm design.

450 References

451 Allen, C. and Hospedales, T. Analogies explained: Towards
 452 understanding word embeddings. In ICML, 2019.

453 Arandjelovic, R. and Zisserman, A. Look, listen and learn.
 454 In ICCV, 2017.

455 Arora, S., Khandeparkar, H., Khodak, M., Plevrakis, O.,
 456 and Saunshi, N. A theoretical analysis of contrastive
 457 unsupervised representation learning. In ICML, 2019.

458 Bachman, P., Hjelm, R. D., and Buchwalter, W. Learning
 459 representations by maximizing mutual information across
 460 views. In NeurIPS, 2019.

461 Baevski, A., Zhou, Y., Mohamed, A., and Auli, M. wav2vec
 462 2.0: A framework for self-supervised learning of speech
 463 representations. In NeurIPS, 2020.

464 Balažević, I., Steiner, D., Parthasarathy, N., Arandjelović,
 465 R., and Hénaff, O. J. Towards in-context scene under-
 466 standing. In NeurIPS, 2023.

467 Balestrieri, R., Ibrahim, M., Sobal, V., Morcos, A., Shekhar,
 468 S., Goldstein, T., Bordes, F., Bardes, A., Mialon, G., Tian,
 469 Y., et al. A cookbook of self-supervised learning. arXiv
 470 preprint arXiv:2304.12210, 2023.

471 Bardes, A., Ponce, J., and LeCun, Y. Vicreg: Variance-
 472 invariance-covariance regularization for self-supervised
 473 learning. In ICLR, 2022.

474 Ben-Shaul, I., Shwartz-Ziv, R., Galanti, T., Dekel, S., and
 475 LeCun, Y. Reverse engineering self-supervised learning.
 476 In NeurIPS, 2023.

477 Caron, M., Bojanowski, P., Joulin, A., and Douze, M. Deep
 478 clustering for unsupervised learning of visual features. In
 479 ECCV, 2018.

480 Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P.,
 481 and Joulin, A. Unsupervised learning of visual features
 482 by contrasting cluster assignments. In NeurIPS, 2020.

483 Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J.,
 484 Bojanowski, P., and Joulin, A. Emerging properties in
 485 self-supervised vision transformers. In ICCV, 2021.

486 Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A
 487 simple framework for contrastive learning of visual rep-
 488 resentations. In ICML, 2020.

489 Chopra, S., Hadsell, R., and LeCun, Y. Learning a sim-
 490 ilarity metric discriminatively, with application to face
 491 verification. In CVPR, 2005.

492 Cover, T. and Hart, P. Nearest neighbor pattern classification.
 493 In IEEE Transactions on Information Theory, 1967.

494 Dhuliawala, S., Sachan, M., and Allen, C. Variational Clas-
 495 sification. In TMLR, 2023.

496 Dosovitskiy, A., Springenberg, J. T., Riedmiller, M., and
 497 Brox, T. Discriminative unsupervised feature learning
 498 with convolutional neural networks. In NeurIPS, 2014.

499 Edwards, H. and Storkey, A. Towards a neural statistician.
 500 In ICLR, 2016.

501 Ericsson, L., Gouk, H., Loy, C. C., and Hospedales,
 502 T. M. Self-supervised representation learning: Introduc-
 503 tion, advances, and challenges. IEEE Signal Processing
 504 Magazine, 39(3):42–62, 2022.

505 Gidaris, S., Singh, P., and Komodakis, N. Unsupervised rep-
 506 resentation learning by predicting image rotations. arXiv
 507 preprint arXiv:1803.07728, 2018.

508 Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P.,
 509 Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z.,
 510 Gheshlaghi Azar, M., et al. Bootstrap your own latent-a
 511 new approach to self-supervised learning. In NeurIPS,
 512 2020.

513 Gupta, K., Ajanthan, T., Hengel, A. v. d., and Gould,
 514 S. Understanding and improving the role of projec-
 515 tion head in self-supervised learning. arXiv preprint
 516 arXiv:2212.11491, 2022.

517 Hadsell, R., Chopra, S., and LeCun, Y. Dimensionality
 518 reduction by learning an invariant mapping. In CVPR,
 519 2006.

520 HaoChen, J. Z., Wei, C., Gaidon, A., and Ma, T. Provable
 521 guarantees for self-supervised deep learning with spectral
 522 contrastive loss. In NeurIPS, 2021.

523 He, J., Gong, Y., Marino, J., Mori, G., and Lehrmann, A.
 524 Variational autoencoders with jointly optimized latent
 525 dependency structure. In ICLR, 2018.

526 He, K., Zhang, X., Ren, S., and Sun, J. Deep residual
 527 learning for image recognition. In CVPR, 2016.

528 He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Mo-
 529 mentum contrast for unsupervised visual representation
 530 learning. In CVPR, 2020.

- 495 He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick,
 496 R. Masked autoencoders are scalable vision learners. In
 497 CVPR, 2022.
- 498 Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and
 499 Hochreiter, S. Gans trained by a two time-scale update
 500 rule converge to a local nash equilibrium. In NeurIPS,
 501 2017.
- 502 Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X.,
 503 Botvinick, M., Mohamed, S., and Lerchner, A. betavae:
 504 Learning basic visual concepts with a constrained
 505 variational framework. In ICLR, 2017.
- 506 Hjelm, R. D., Fedorov, A., Lavoie-Marchildon, S., Grewal,
 507 K., Bachman, P., Trischler, A., and Bengio, Y. Learning
 508 deep representations by mutual information estimation
 509 and maximization. In ICLR, 2019.
- 510 Karras, T., Laine, S., and Aila, T. A style-based genera-
 511 tor architecture for generative adversarial networks. In
 512 CVPR, 2019.
- 513 Kingma, D. P. and Welling, M. Auto-encoding variational
 514 bayes. In ICLR, 2014.
- 515 Krizhevsky, A., Hinton, G., et al. Learning multiple layers
 516 of features from tiny images. 2009.
- 517 LeCun, Y. The mnist database of handwritten digits.
 518 <http://yann.lecun.com/exdb/mnist/>, 1998.
- 519 Lee, J. D., Lei, Q., Saunshi, N., and Zhuo, J. Predicting
 520 what you already know helps: Provable self-supervised
 521 learning. In NeurIPS, 2021.
- 522 Levy, O. and Goldberg, Y. Neural word embedding as
 523 implicit matrix factorization. In NeurIPS, 2014.
- 524 Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face
 525 attributes in the wild. In ICCV, 2015.
- 526 McAllester, D. and Stratos, K. Formal limitations on the
 527 measurement of mutual information. In AISTATS, 2020.
- 528 Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and
 529 Dean, J. Distributed representations of words and phrases
 530 and their compositionality. In NeurIPS, 2013.
- 531 Misra, I. and Maaten, L. v. d. Self-supervised learning of
 532 pretext-invariant representations. In CVPR, 2020.
- 533 Nakamura, H., Okada, M., and Taniguchi, T. Represen-
 534 tation uncertainty in self-supervised learning as variational
 535 inference. In ICCV, 2023.
- 536 Oord, A. v. d., Li, Y., and Vinyals, O. Representation learn-
 537 ing with contrastive predictive coding. arXiv preprint
 538 arXiv:1807.03748, 2018.
- 539 Ozsoy, S., Hamdan, S., Arik, S., Yuret, D., and Erdogan, A.
 540 Self-supervised learning with an information maximiza-
 541 tion criterion. In NeurIPS, 2022.
- 542 Papyan, V., Han, X., and Donoho, D. L. Prevalence of
 543 neural collapse during the terminal phase of deep learn-
 544 ing training. Proceedings of the National Academy of
 545 Sciences, 117(40):24652–24663, 2020.
- 546 Poole, B., Ozair, S., Van Den Oord, A., Alemi, A., and
 547 Tucker, G. On variational bounds of mutual information.
 548 In ICML, 2019.
- 549 Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G.,
 550 Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J.,
 551 et al. Learning transferable visual models from natural
 552 language supervision. In ICML, 2021.
- 553 Ranganath, R., Tran, D., and Blei, D. Hierarchical varia-
 554 tional models. In ICML, 2016.
- 555 Rolfe, J. T. Discrete variational autoencoders. In ICLR,
 556 2017.
- 557 Sansone, E. and Manhaeve, R. Gedi: Generative and dis-
 558 criminative training for self-supervised learning. arXiv
 559 preprint arXiv:2212.13425, 2022.
- 560 Saunshi, N., Ash, J., Goel, S., Misra, D., Zhang, C., Arora,
 561 S., Kakade, S., and Krishnamurthy, A. Understanding
 562 contrastive learning requires incorporating inductive bi-
 563 ases. In ICML, 2022.
- 564 Shwartz-Ziv, R., Balestrieri, R., Kawaguchi, K., Rudner,
 565 T. G., and LeCun, Y. An information-theoretic perspec-
 566 tive on variance-invariance-covariance regularization. In
 567 NeurIPS, 2023.
- 568 Sinha, A., Song, J., Meng, C., and Ermon, S. D2c:
 569 Diffusion-decoding models for few-shot conditional gen-
 570 eration. In NeurIPS, 2021.
- 571 Sinha, S. and Dieng, A. B. Consistency regularization for
 572 variational auto-encoders. In NeurIPS, 2021.
- 573 Sohn, K. Improved deep metric learning with multi-class
 574 n-pair loss objective. In NeurIPS, 2016.
- 575 Sønderby, C. K., Raiko, T., Maaløe, L., Sønderby, S. K., and
 576 Winther, O. Ladder variational autoencoders. In NIPS,
 577 2016.
- 578 Song, C., Liu, F., Huang, Y., Wang, L., and Tan, T. Auto-
 579 encoder based data clustering. In Progress in Pattern
 580 Recognition, Image Analysis, Computer Vision, and
 581 Applications, 2013.
- 582 Tian, Y. Understanding deep contrastive learning via
 583 coordinate-wise optimization. In NeurIPS, 2022.

- 550 Tosh, C., Krishnamurthy, A., and Hsu, D. Contrastive
 551 learning, multi-view redundancy, and linear models. In *Algorithmic Learning Theory*, pp. 1179–1206. PMLR,
 552 2021.
- 553
- 554 Tsai, Y.-H. H., Wu, Y., Salakhutdinov, R., and Morency, L.-
 555 P. Self-supervised learning from a multi-view perspective.
 556 In *ICLR*, 2020.
- 557
- 558 Tschannen, M., Djolonga, J., Rubenstein, P. K., Gelly, S.,
 559 and Lucic, M. On mutual information maximization for
 560 representation learning. In *ICLR*, 2020.
- 561
- 562 Valpola, H. From neural pca to deep unsupervised learning.
 563 In *Advances in Independent Component Analysis and*
 564 *learning machines*, pp. 143–171. Elsevier, 2015.
- 565
- 566 Von Kriegelgen, J., Sharma, Y., Gresele, L., Brendel, W.,
 567 Schölkopf, B., Besserve, M., and Locatello, F. Self-
 568 supervised learning with data augmentations provably
 569 isolates content from style. In *NeurIPS*, 2021.
- 570
- 571 Wang, T. and Isola, P. Understanding contrastive repres-
 572 entation learning through alignment and uniformity on the
 573 hypersphere. In *ICML*, 2020.
- 574
- 575 Wang, Y., Zhang, Q., Wang, Y., Yang, J., and Lin, Z. Chaos
 576 is a ladder: A new theoretical understanding of con-
 577 trastive learning via augmentation overlap. In *ICLR*,
 578 2021.
- 579
- 580 Wang, Y., Tang, S., Zhu, F., Bai, L., Zhao, R., Qi, D., and
 581 Ouyang, W. Revisiting the transferability of supervised
 582 pretraining: an mlp perspective. In *CVPR*, 2022.
- 583
- 584 Wu, C., Pfrommer, J., Zhou, M., and Beyerer, J. Generative-
 585 contrastive learning for self-supervised latent repres-
 586 entations of 3d shapes from multi-modal euclidean input.
 587 *arXiv preprint arXiv:2301.04612*, 2023.
- 588
- 589 Wu, Z., Xiong, Y., Yu, S. X., and Lin, D. Unsupervised fea-
 590 ture learning via non-parametric instance discrimination.
 591 In *CVPR*, 2018.
- 592
- 593 Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a
 594 novel image dataset for benchmarking machine learning
 595 algorithms. *arXiv preprint arXiv:1708.07747*, 2017.
- 596
- 597 Xie, J., Girshick, R., and Farhadi, A. Unsupervised deep
 598 embedding for clustering analysis. In *ICML*, 2016.
- 599
- 600 Xie, Z., Zhang, Z., Cao, Y., Lin, Y., Bao, J., Yao, Z., Dai, Q.,
 601 and Hu, H. Simmim: A simple framework for masked
 602 image modeling. In *CVPR*, 2022.
- 603
- 604 Yang, B., Fu, X., Sidiropoulos, N. D., and Hong, M. To-
 605 wards k-means-friendly spaces: Simultaneous deep learn-
 606 ing and clustering. In *ICML*, 2017.
- Zhang, M., Xiao, T. Z., Paige, B., and Barber, D. Improv-
 ing vae-based representation learning. *arXiv preprint*
arXiv:2205.14539, 2022.
- Zimmermann, R. S., Sharma, Y., Schneider, S., Bethge, M.,
 and Brendel, W. Contrastive learning inverts the data
 generating process. In *ICML*, 2021.

A. Appendix

A.1. Background: Relevant VAE architectures

The proposed hierarchical latent variable model for self-supervised learning (Fig. 1) is fitted to the data distribution by maximising the ELBO_{SSL} and can be viewed as a VAE with a hierarchical prior.

VAEs have been extended to model hierarchical latent structure (e.g. Valpola, 2015; Ranganath et al., 2016; Rolfe, 2017; He et al., 2018; Sønderby et al., 2016; Edwards & Storkey, 2016), which our work relates to. Notably, Edwards & Storkey (2016) propose the same graphical model as Fig. 1, but methods differ in how posteriors are factorised, which is a key aspect for learning informative representations that depend only on the sample they represent. Wu et al. (2023) and Sinha et al. (2021) combine aspects of VAEs and contrastive learning but do not propose a latent variable model for SSL. Nakamura et al. (2023) look to explain SSL methods via the ELBO, but with a second posterior approximation not a generative model.

A.2. Derivation of ELBO_{SSL} and SimVAE Objective

Let $\mathbf{x} = \{x^1, \dots, x^J\}$ be a set of J semantically related samples, $\omega = \{\theta, \psi, \pi\}$ be parameters of the generative model for SSL (Fig. 1) and ϕ parameterise the approximate posterior $q_\phi(\mathbf{z}|\mathbf{x})$. We derive the Evidence Lower Bound (ELBO_{SSL}) that underpins the training objectives of (projective) discriminative SSL methods and is used to train SimVAE (§3.4).

$$\begin{aligned}
\min_{\omega} D_{\text{KL}}[p(\mathbf{x}|y) \| p_{\omega}(\mathbf{x}|y)] &= \max_{\omega} \mathbb{E}_{\mathbf{x}, y} [\log p_{\omega}(\mathbf{x}|y)] \\
&= \max_{\omega, \phi} \mathbb{E}_{\mathbf{x}, y} \left[\int_{\mathbf{z}} q_\phi(\mathbf{z}|\mathbf{x}) \log p_{\omega}(\mathbf{x}|y) \right] \\
&= \max_{\omega, \phi} \mathbb{E}_{\mathbf{x}, y} \left[\int_{\mathbf{z}} q_\phi(\mathbf{z}|\mathbf{x}) \log \frac{p_{\theta}(\mathbf{x}|\mathbf{z}) p_{\psi}(\mathbf{z}|y)}{p_{\omega}(\mathbf{z}|\mathbf{x}, y)} \frac{q_{\phi}(\mathbf{z}|\mathbf{x})}{q_{\phi}(\mathbf{z}|\mathbf{x})} \right] \\
&= \max_{\omega, \phi} \mathbb{E}_{\mathbf{x}, y} \left[\int_{\mathbf{z}} q_\phi(\mathbf{z}|\mathbf{x}) \log \frac{p_{\theta}(\mathbf{x}|\mathbf{z}) p_{\psi}(\mathbf{z})|y}{q_{\phi}(\mathbf{z}|\mathbf{x})} \right] + D_{\text{KL}}[q_{\phi}(\mathbf{z}|\mathbf{x}) \| p_{\omega}(\mathbf{z}|\mathbf{x}, y)] \\
&\geq \max_{\omega, \phi} \mathbb{E}_{\mathbf{x}, y} \left[\int_{\mathbf{z}} q_{\phi}(\mathbf{z}|\mathbf{x}) \left\{ \log \frac{p_{\theta}(\mathbf{x}|\mathbf{z})}{q_{\phi}(\mathbf{z}|\mathbf{x})} + \log p_{\psi}(\mathbf{z}|y) \right\} \right] \quad (\text{cf Eq. 2}) \\
&= \max_{\omega, \phi} \mathbb{E}_{\mathbf{x}, y} \left[\sum_j \left\{ \int_{z^j} q_{\phi}(z^j|x^j) \log \frac{p_{\theta}(x^j|z^j)}{q_{\phi}(z^j|x^j)} \right\} + \int_{\mathbf{z}} q_{\phi}(\mathbf{z}|\mathbf{x}) \log p_{\psi}(\mathbf{z}|y) \right] \quad (= \text{Eq. 4})
\end{aligned}$$

Terms of ELBO_{SSL} are analogous to those of the standard ELBO: *reconstruction error*, *entropy* of the approximate posterior $H(q_{\phi}(\mathbf{z}|\mathbf{x}))$ and the (conditional) prior. Algorithm 1 provides an overview of the computational steps required to maximise ELBO_{SSL} under Gaussian assumptions described in §3.4, referred to as SimVAE. As our experimental setting considers augmentations as semantically related samples, Algorithm 1 incorporates a preliminary step to augment data samples.

A.3. Detailed derivation of InfoNCE Objective

For data sample $x_{i_0} \in \mathbf{x}_{i_0}$, let $x'_{i_0} \in \mathbf{x}_{i_0}$ be a semantically related *positive* sample and $\{x'_{i_r}\}_{r=1}^k$ be random *negative* samples. Denote by $\mathbf{x} = \{x_{i_0}, x'_{i_0}\}$ the positive pair, by $X^- = \{x'_{i_1}, \dots, x'_{i_k}\}$ all negative samples, and by $X = \mathbf{x} \cup X^-$ all samples. The InfoNCE objective is derived as follows (by analogy to Oord et al. (2018)).

$$\begin{aligned}
&\mathbb{E}_X [\log p(y=0|X)] && \text{(predict } y = \text{index of positive sample in } X^- \text{)} \\
&= \mathbb{E}_X [\log \int_Z p(y=0|Z) q(Z|X)] && \text{(introduce latent variables } Z: Y \rightarrow Z \rightarrow X \text{)} \\
&\geq \mathbb{E}_X \left[\int_Z q(Z|X) \log p(y=0|Z) \right] && \text{(by Jensen's inequality)} \\
&= \mathbb{E}_X \left[\int_Z q(Z|X) \log \frac{p(Z'|z_{i_0}, y=0) \cancel{p(y=0)}}{\sum_{r=0}^k p(Z'|z_{i_0}, y=r) \cancel{p(y=r)}} \right] && \text{(Bayes rule, note } p(y=r) = \frac{1}{k}, \forall r \text{)} \\
&= \mathbb{E}_X \left[\int_Z q(Z|X) \log \frac{p(z'_{i_0}|z_{i_0}) \prod_{s \neq 0} p(z'_{i_s})}{\sum_{r=0}^k p(z'_{i_r}|z_{i_0}) \prod_{s \neq r} p(z'_{i_s})} \right] && \text{(from sample similarity/independence)} \\
&= \mathbb{E}_X \left[\int_Z q(Z|X) \log \frac{p(z'_{i_0}|z_{i_0})/p(z'_{i_0})}{\sum_{r=0}^k p(z'_{i_r}|z_{i_0})/p(z'_{i_r})} \right] && \text{(divide through by } \prod_{s=0}^k p(z'_{i_s}) \text{)}
\end{aligned}$$

660
661
662
663
664
665
666

$$= \mathbb{E}_X \left[\int_Z q(Z|X) \log \frac{p(z'_{i_0}, z_{i_0}) / p(z'_{i_0})}{\sum_{r=0}^k p(z'_{i_r}, z_{i_0}) / p(z'_{i_r})} \right] \quad (10)$$

The final expression is parameterised using a similarity function $\text{sim}(z, z')$ to give the objective.

667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694

$$-\mathcal{L}_{\text{INCE}} \doteq \mathbb{E}_X \left[\int_Z q(Z|X) \log \frac{\text{sim}(z'_{i_0}, z_{i_0})}{\sum_{r=0}^k \text{sim}(z'_{i_r}, z_{i_0})} \right]$$

Oord et al. (2018) show, and Poole et al. (2019) confirm, that this loss is a lower bound on the mutual information, which improves with the number of negative samples k .

670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714

$$\begin{aligned} & -\mathcal{L}_{\text{INCE}} \\ &= \mathbb{E}_X \left[\int_Z q(Z|X) \log \frac{p(z'_{i_0}, z_{i_0}) / p(z'_{i_0})}{\sum_{r=0}^k p(z'_{i_r}, z_{i_0}) / p(z'_{i_r})} \right] \quad (\text{multiply Eq. 10 through by } p(z_{i_0})) \\ &= \mathbb{E}_X \left[\int_Z q(Z|X) \log p(z_{i_0}|z'_{i_0}) - \log \left(p(z_{i_0}|z'_{i_0}) + \sum_{r=1}^k p(z_{i_0}|z'_{i_r}) \right) \right] \\ &= -\mathbb{E}_X \left[\int_Z q(Z|X) \log \left(1 + \sum_{r=1}^k \frac{p(z_{i_0}|z'_{i_r})}{p(z_{i_0}|z'_{i_0})} \right) \right] \quad (\text{divide through by } p(z_{i_0}|z'_{i_0})) \\ &\approx -\mathbb{E}_{\mathbf{x}} \left[\int_{\mathbf{z}} q(\mathbf{z}|\mathbf{x}) \log \left(1 + (k-1) \mathbb{E}_{x'_j} \left[\int_{z'_j} q(z'_j|x'_j) \frac{p(z_{i_0}|z'_j)}{p(z_{i_0}|z'_{i_0})} \right] \right) \right] \\ &= -\mathbb{E}_{\mathbf{x}} \left[\int_{\mathbf{z}} q(\mathbf{z}|\mathbf{x}) \log \left(1 + (k-1) \frac{p(z_{i_0})}{p(z_{i_0}|z'_{i_0})} \right) \right] \\ &\leq -\mathbb{E}_{\mathbf{x}} \left[\int_{\mathbf{z}} q(\mathbf{z}|\mathbf{x}) \log (k-1) \frac{p(z_{i_0})}{p(z_{i_0}|z'_{i_0})} \right] \\ &= \mathbb{E}_{\mathbf{x}} \left[\int_{\mathbf{z}} q(\mathbf{z}|\mathbf{x}) \log \frac{p(z_{i_0}, z'_{i_0})}{p(z_{i_0})p(z'_{i_0})} \right] + \log \frac{1}{(k-1)} \\ &\stackrel{k \rightarrow \infty}{\rightarrow} \mathbb{E}_{\mathbf{x}} \left[\int_{\mathbf{z}} q_{\phi}(\mathbf{z}|\mathbf{x}) (\log p(\mathbf{z}|y_{i_0}) - \sum_j \log p(z_{i_0}^j)) \right] \end{aligned}$$

In the last step, we revert to the terminology used in the main paper for ease of reference.

A.4. Derivation of parameter-free $p(\mathbf{z}_i|y_i) = s(\mathbf{z}_i)$

Instance Discrimination methods consider $J=1$ sample x_i at a time, labelled by its index $y_i=i$, and computes $p(x_i|y=i; \theta_i)$ from stored instance-specific parameters θ_i . This requires parameters proportional to the dataset size, which could be prohibitive, whereas parameter number is often independent of the dataset size, or grows slowly. We show that contrastive methods (approximately) optimise the same objective, but without parameters, and here explain how that is possible. Recall that the “label” i is semantically meaningless and simply identifies samples of a common distribution $p(\mathbf{x}|y=i) \doteq p(\mathbf{x}|y_i)$. For $J \geq 2$ semantically related samples $\mathbf{x}_i = \{x_i^j\}_{j=1}^J$, $x_i^j \sim p(\mathbf{x}|y_i)$, their latent variables are conditionally independent, hence $p(\mathbf{z}_i|y_i) = \int_{\psi} p(\psi_i) p(\mathbf{z}_i|y_i; \psi_i) = \int_{\psi} p(\psi_i) \prod_j p(z_i^j|y_i; \psi_i) = s(\mathbf{z}_i)$, a function of the latent variables that non-parametrically approximates the joint distribution of latent variables of semantically related data. (Note that unsemantically related data are independent and the joint distribution over their latent variables is a product of marginals).

We assume a Gaussian prior $p(\psi_i) = \mathcal{N}(\psi_i; 0, \gamma^2 I)$ and class-conditionals $p(z_i^j|\psi_i) = \mathcal{N}(z_i^j; \psi_i, \sigma^2)$ (for fixed variance σ^2).

$$\begin{aligned} p(\mathbf{z}_i|y_i) &= \int_{\psi_i} p(\mathbf{z}_i|\psi_i) p(\psi_i) = \int_{\psi_i} p(\psi_i) \prod_j p(z_i^j|\psi_i) \\ &\propto \int_{\psi_i} \exp\left\{-\frac{1}{2\gamma^2}\psi_i^2\right\} \prod_j \exp\left\{-\frac{1}{2\sigma^2}(z_i^j - \psi_i)^2\right\} \\ &= \int_{\psi_i} \exp\left\{-\frac{1}{2\sigma^2}\left(\frac{\sigma^2}{\gamma^2}\psi_i^2 + \sum_j (z_i^j - \psi_i)^2\right)\right\} \end{aligned}$$

$$\begin{aligned}
&= \int_{\psi_i} \exp\left\{-\frac{1}{2\sigma^2}\left(\left(\sum_j z_i^{j2}\right) - 2\left(\sum_j z_i^j\right)\psi_i + \left(\frac{\sigma^2}{\gamma^2} + J\right)\psi_i^2\right)\right\} \\
&= \int_{\psi_i} \exp\left\{-\frac{\sigma^2/\gamma^2+J}{2\sigma^2}\left(\psi_i - \frac{1}{\sigma^2/\gamma^2+J}\sum_j z_i^j\right)^2\right\} + \exp\left\{-\frac{1}{2\sigma^2}\left(\sum_j z_i^{j2} - \frac{1}{\sigma^2/\gamma^2+J}\left(\sum_j z_i^j\right)^2\right)\right\} \quad (*) \\
&\propto \exp\left\{-\frac{1}{2\sigma^2}\left(\sum_j z_i^{j2} - \frac{1}{\sigma^2/\gamma^2+J}\left(\sum_j z_i^{j2} + \sum_{j \neq k} z_i^j z_i^k\right)\right)\right\} \\
&= \exp\left\{-\frac{1}{2\sigma^2}\left((1 - \frac{1}{\sigma^2/\gamma^2+J})\sum_j z_i^{j2} + \frac{1}{\sigma^2/\gamma^2+J}\sum_{j \neq k} z_i^j z_i^k\right)\right\} \\
&\propto \exp\left\{-\frac{1}{2\sigma^2(\sigma^2/\gamma^2+J)}\sum_{j \neq k} z_i^j z_i^k\right\} \quad (\text{if } \|z\|_2 = 1)
\end{aligned}$$

The result can be rearranged into a Gaussian form (a well known result when all distributions are Gaussian), but the last line also shows that, under the common practice of setting embeddings to unit length ($\|z\|_2 = 1$), $s(\cdot)$ can be calculated directly from dot products, or cosine similarities (up to a proportionality constant, which does not affect optimisation).

If we instead assume a uniform prior, we can take the limit of the line marked (*) as $\gamma \rightarrow \infty$:

$$\begin{aligned}
&\exp\left\{-\frac{1}{2\sigma^2}\left(\left(\sum_j z_i^{j2}\right) - \frac{1}{\sigma^2/\gamma^2+J}\left(\sum_j z_i^j\right)^2\right)\right\} \\
&\rightarrow \exp\left\{-\frac{1}{2\sigma^2}\left(\left(\sum_j z_i^{j2}\right) - \frac{1}{J}\left(\sum_j z_i^j\right)^2\right)\right\} \\
&= \exp\left\{-\frac{1}{2\sigma^2}\left(\left(\sum_j z_i^{j2}\right) - J\bar{z}_i^2\right)\right\} \\
&= \exp\left\{-\frac{1}{2\sigma^2}\left(\sum_j z_i^{j2} - 2J\bar{z}_i^2 + J\bar{z}_i^2\right)\right\} \\
&= \exp\left\{-\frac{1}{2\sigma^2}\left(\sum_j z_i^{j2} - 2\bar{z}_i\left(\sum_j z_i^j\right) + \sum_j \bar{z}_i^2\right)\right\} \\
&= \exp\left\{-\frac{1}{2\sigma^2}\sum_j (z_i^{j2} - 2z_i^j \bar{z}_i + \bar{z}_i^2)\right\} \\
&= \exp\left\{-\frac{1}{2\sigma^2}\sum_j (z_i^j - \bar{z}_i)^2\right\}
\end{aligned} \tag{11}$$

A.5. Relationship between InfoNCE Representations and PMI

For data sampled $x \sim p(x)$ and augmentations $x' \sim p_\tau(x'|x)$ sampled under a synthetic augmentation strategy, Oord et al. (2018) show that the InfoNCE objective for a sample x is optimised if their respective representations z, z' satisfy

$$\exp\{sim(z, z')\} = c \frac{p(x, x')}{p(x)p(x')}, \tag{12}$$

where $sim(\cdot, \cdot)$ is the similarity function (e.g. dot product), and c is a proportionality constant, specific to x . Since c may differ arbitrarily with x it can be considered an arbitrary function of x , but for simplicity we consider a particular x and fixed c . Further, $c > 0$ is strictly positive since it is a ratio between positive (exponential) and non-negative (probability ratio) terms. Accordingly, representations satisfy

$$sim(z, z') = PMI(x, x') + c', \tag{13}$$

where $c' = \log c \in \mathbb{R}$ and $PMI(x, x')$ is the *pointwise mutual information* between samples x and x' . Pointwise mutual information (PMI) is an information theoretic term that reflects the probability of events occurring jointly versus independently. For an arbitrary sample and augmentation this is given by:

$$PMI(x, x') \doteq \log \frac{p(x, x')}{p(x)p(x')} = \log \frac{p_\tau(x'|x)}{p(x')}. \tag{14}$$

We note that $p_\tau(x'|x) = 0$ if x can *not* be augmented to produce x' ; and that, in a continuous domain, such as images, two augmentations are identical with probability zero. Thus augmentations of different samples are expected to not overlap and the marginal is given by $p(x') = \int_x p_\tau(x'|x)p(x) = p_\tau(x'|x^*)p(x^*)$, where x^* is the sample augmented to give x' . Thus

$$\frac{p_\tau(x'|x)}{p(x')} = \frac{p_\tau(x'|x)}{p_\tau(x'|x^*)p(x^*)} = \begin{cases} 1/p(x^*) & \text{if } x^* = x \text{ (i.e. } x' \text{ is an augmentation of } x) \\ 0 & \text{otherwise;} \end{cases} \quad (15)$$

and $\text{PMI}(x, x') = -\log p(x) \geq k > 0$ or $\text{PMI}(x, x') = -\infty$, respectively. Here $k = -\log \arg \max_x p(x)$ is a finite value based on the most likely sample. For typical datasets, this can be approximated empirically by $\frac{1}{N}$ where N is the size of the original dataset (since that is how often the algorithm observes each sample), hence $k = \log N$, often of the order 5 – 10 (depending on the dataset).

If the main objective were to accurately approximate PMI (subject to a constant c') in Eq. 13, e.g. to approximate *mutual information*, or if representation learning *depended* on it, then, at the very least, the domain of $\text{sim}(\cdot, \cdot)$ must span its range of values, seen above as from $-\infty$ for negative samples to a small positive value (e.g. 5–10) for positive samples. Despite this, the popular *bounded* cosine similarity function ($\text{cossim}(z, z') = \frac{z^T z'}{\|z\|_2 \|z'\|_2} \in [-1, 1]$) is found to outperform the *unbounded* dot product, even though the cosine similarity function necessarily cannot span the range required to reflect true PMI values, while the dot product can. This strongly suggests that representation learning does not require representations to specifically learn PMI, or for the overall loss function to approximate mutual information.

Instead, with the cosine similarity *constraint*, the InfoNCE objective is *as optimised as possible* if representations of a data sample and its augmentations are fully aligned ($\text{cossim}(z, z') = 1$) and representations of dissimilar data are maximally misaligned $\text{cossim}(z, z') = -1$, since these minimise the error from the true PMI values for positive and negative samples (described above). Constraints, such as the dimensionality of the representation space vs the number of samples, may prevent these revised theoretical optima being fully achieved, but the loss function is optimised by clustering representations of a sample and its augmentations and spreading apart those clusters. Note that this is the same geometric structure as induced under softmax cross-entropy loss (Dhuliawala et al., 2023).

We note that our theoretical justification for representations *not* capturing PMI is supported by the empirical observation that closer approximations of mutual information do not appear to improve representations (Tschanen et al., 2020). Also, more recent contrastive self-supervised methods increase the cosine similarity between semantically related data but spread apart representation without negative sampling of InfoNCE, yet outperform the InfoNCE objective despite having no obvious relationship to PMI (Grill et al., 2020; Bardes et al., 2022).

A.6. Information Loss due to Representaiton Collapse: a discussion

While it may seem appealing to lose information by way of representation collapse, e.g. to obtain representations *invariant* to nuisance factors, this is a problematic notion from the perspective of *general-purpose* representation learning, where the downstream task is unknown or there may be many, since what is noise for one task may be of use in another. For example, “blur” is often considered noise, but a camera on an autonomous vehicle may be better to *detect* blur (e.g. from soiling) than be *invariant* to it and eventually fail when blurring becomes too severe. We note that humans can observe a scene *including* many irrelevant pieces of information, e.g. we can tell when an image is blurred or that we are looking through a window, and “disentangle” that from the rest of the image. This suggests that factors can be *preserved* and *disentangled*.

To stress the point that representation collapse is not desirable in and of itself, we note that collapsing together representations of semantically related data \mathbf{x}_i would be problematic if subsets \mathbf{x}_i overlap. For example, in the discrete case of *word2vec*, words are considered semantically related if they co-occur within a fixed window. Representation collapse, here, would mean that co-occurring words belonging to the same \mathbf{x}_i would have *the same representation*, which is clearly undesirable.

825 **A.7. Experimental Details**

826 **A.7.1. SIMVAE ALGORITHM**

828 **Algorithm 1** SimVAE

```

830 Require: data  $\{\mathbf{x}_k\}_{k=1}^M$ ; batch size  $N$ ; data dim  $D$ ; latent dim  $L$ ; augmentation set  $\mathcal{T}$ ; num views  $J$ ; encoder  $f_\phi$ ; decoder
831  $g_\theta$ ;  $\text{Var}(\mathbf{z}|y) \boldsymbol{\sigma}^2$ ;
832 for randomly sampled mini-batch  $\{\mathbf{x}_k\}_{k=1}^N$  do
833   for augmentation  $t^j \sim \mathcal{T}$  do
834      $\mathbf{x}_k^j = t^j(\mathbf{x}_k)$ ; # augment samples
835      $\boldsymbol{\mu}_k^j, \boldsymbol{\Sigma}_k^j = f_\phi(\mathbf{x}_k^j)$ ; # forward pass:  $\mathbf{z} \sim p_\phi(\mathbf{z}|\mathbf{x})$ 
836      $\mathbf{z}_k^j \sim \mathcal{N}(\boldsymbol{\mu}_k^j, \boldsymbol{\Sigma}_k^j)$ ;
837      $\tilde{\mathbf{x}}_k^j = g_\theta(\mathbf{z}_k^j)$ ; #  $\tilde{\mathbf{x}} = \mathbb{E}[\mathbf{x}|\mathbf{z}; \theta]$ 
838   end for
839    $\mathcal{L}_{\text{rec}}^k = \frac{1}{D} \sum_{j=1}^J \|\mathbf{x}_k^j - \tilde{\mathbf{x}}_k^j\|_2^2$  # minimize loss
840    $\mathcal{L}_{\text{H}}^k = \frac{1}{2} \sum_{j=1}^J \log(|\boldsymbol{\Sigma}_k^j|)$ 
841    $\mathcal{L}_{\text{prior}}^k = \frac{1}{2} \sum_{j=1}^J \|(\mathbf{z}_k^j - \frac{1}{J} \sum_{j=1}^J \mathbf{z}_k^j)/\boldsymbol{\sigma}\|_2^2$ 
842    $\min(\sum_{k=1}^N \mathcal{L}_{\text{rec}}^k + \mathcal{L}_{\text{H}}^k + \mathcal{L}_{\text{prior}}^k)$  w.r.t.  $\phi, \theta$  by SGD;
843   end for
844 return  $\phi, \theta$ ;


---



```

847 **A.7.2. DATASETS**

849 **MNIST** The MNIST dataset ([LeCun, 1998](#)) gathers 60'000 training and 10'000 testing images representing digits from 0 to 9 in various caligraphic styles. Images were kept to their original 28x28 pixel resolution and were binarized. The 10-class digit classification task was used for evaluation.

851 **FashionMNIST** The FashionMNIST dataset ([Xiao et al., 2017](#)) is a collection of 60'000 training and 10'000 test images depicting Zalando clothing items (i.e., t-shirts, trousers, pullovers, dresses, coats, sandals, shirts, sneakers, bags and ankle boots). Images were kept to their original 28x28 pixel resolution. The 10-class clothing type classification task was used for evaluation.

855 **CIFAR10** The CIFAR10 dataset ([Krizhevsky et al., 2009](#)) offers a compact dataset of 60,000 (50,000 training and 10,000 testing images) small, colorful images distributed across ten categories including objects like airplanes, cats, and ships, with various lighting conditions. Images were kept to their original 32x32 pixel resolution.

857 **CelebA** The CelebA dataset ([Liu et al., 2015](#)) comprises a vast collection of celebrity facial images. It encompasses a diverse set of 183'000 high-resolution images (i.e., 163'000 training and 20'000 test images), each depicting a distinct individual. The dataset showcases a wide range of facial attributes and poses and provides binary labels for 40 facial attributes including hair & skin colour, presence or absence of attributes such as eyeglasses and facial hair. Each image was cropped and resized to a 64x64 pixel resolution. Attributes referring to hair colour were aggregated into a 5-class attribute (i.e., bald, brown hair, blond hair, gray hair, black hair). Images with missing or ambiguous hair colour information were discarded at evaluation.

861 All datasets were sourced from Pytorch's dataset collection.

862 **A.7.3. DATA AUGMENTATION STRATEGY**

863 Taking inspiration from SimCLR's ([Chen et al., 2020](#)) augmentation strategy which highlights the importance of random 864 image cropping and colour jitter on downstream performance, our augmentation strategy includes random image cropping, 865 random image flipping and random colour jitter. The colour augmentations are only applied to the non gray-scale datasets 866 (i.e., CIFAR10 ([Krizhevsky et al., 2009](#)) & CelebA dataset ([Liu et al., 2015](#))). Due to the varying complexity of the datasets 867 we explored, hyperparameters such as the cropping strength were adapted to each dataset to ensure that semantically 868 meaningful features remained after augmentation. The augmentation strategy hyperparameters used for each dataset are 869 detailed in table 2.

	Dataset	Crop		Vertical Flip	colour Jitter		
		scale	ratio	prob.	b-s-c	hue	prob.
	MNIST	0.4	[0.75,1.3]	0.5	-	-	-
	Fashion	0.4	[0.75,1.3]	0.5	-	-	-
	CIFAR10	0.6	[0.75,1.3]	0.5	0.8	0.2	0.8
	CelebA	0.6	[0.75,1.3]	0.5	0.8	0.2	0.8

Table 2: Data augmentation strategy for each dataset: (left to right) cropping scale, cropping ratio, probability of vertical/horizontal flip, brightness-saturation-contrast jitter, hue jitter, probability of colour jitter

A.7.4. TRAINING IMPLEMENTATION DETAILS

This section contains all details regarding the architectural and optimization design choices used to train SimVAE and all baselines. Method-specific hyperparameters are also reported below.

Network Architectures The encoder network architectures used for SimCLR, MoCo, VicREG, and VAE-based approaches including SimVAE for simple (i.e., MNIST, FashionMNIST) and complex datasets (i.e., CIFAR10, CelebA) are detailed in Table 3a, Table 4a respectively. Generative models which include all VAE-based methods also require decoder networks for which the architectures are detailed in Table 3b and Table 4b. The latent dimensionality for MNIST and FashionMNIST is fixed at 10 and increased to 64 for the CelebA and CIFAR10 datasets. The encoder and decoder architecture networks are kept constant across methods including the latent dimensionality to ensure a fair comparison.

Layer Name	Output Size	Block Parameters	Layer Name	Output Size	Block Parameters
fc1	500	784x500 fc, relu	fc1	2000	10x2000 fc, relu
fc2	500	500x500 fc, relu	fc2	500	2000x500 fc, relu
fc3	2000	500x2000 fc, relu	fc3	500	500x500 fc, relu
fc4	10	2000x10 fc	fc4	784	500x784 fc

(a) Encoder

(b) Decoder

Table 3: Multi-layer perceptron network architectures used for MNIST & FashionMNIST training

Layer Name	Output	Block Parameters
conv1	32x32	4x4, 16, stride 1 batchnorm, relu 3x3 maxpool, stride 2
conv2_x	32x32	3x3, 32, stride 1 3x3, 32, stride 1
conv3_x	16x16	3x3, 64, stride 2 3x3, 64, stride 1
conv4_x	8x8	3x3, 128, stride 2 3x3, 128, stride 1
conv5_x	4x4	3x3, 256, stride 2 3x3, 256, stride 1
fc	64	4096x64 fc

(a) Encoder

Layer Name	Output	Block Parameters
fc	256x4x4	64x4096 fc
conv1_x	8x8	3x3, 128, stride 2 3x3, 128, stride 1
conv2_x	16x16	3x3, 64, stride 2 3x3, 64, stride 1
conv3_x	32x32	3x3, 32, stride 2 3x3, 32, stride 1
conv4_x	64x64	3x3, 16, stride 2 3x3, 16, stride 1
conv5	64x64	5x5, 3, stride 1

(b) Decoder

Table 4: Resnet18 network architectures used for CIFAR10 & CelebA datasets

Optimisation & Hyper-parameter tuning All methods were trained using an Adam optimizer until training loss convergence. The batch size was fixed to 128. Hyper-parameter tuning was performed based on the downstream MLP classification accuracy across datasets. The final values of hyperparameters were selected to reach the best average downstream performance across datasets. While we observed stable performances across datasets for the VAE family of models, VicREG and

MoCo, SimCLR is more sensitive, leading to difficulties when having to define a unique set of parameters across datasets. For VAEs, the learning rate was set to $8e^{-5}$, and the likelihood probability, $p(x|z)$, variance parameter was set to 0.02 for β -VAE, CR-VAE and SimVAE. CR-VAE's λ parameter was set to 0.1. SimVAE's prior probability, $p(z|y)$, variance was set to 0.15 and the number of augmentations to 10. VicREG's parameter μ was set to 25 and learning rate to 1e-4. SimCLR's temperature parameter, τ , was set to 0.7 and learning rates were adapted for each dataset due to significant performance variations across datasets ranging from $8e^{-5}$ to $1e^{-3}$.

A.7.5. EVALUATION IMPLEMENTATION DETAILS

Following common practices (Chen et al., 2020), downstream performance is assessed using a linear probe, a multi-layer perceptron probe, a k-nearest neighbors (kNN) algorithm, and a Gaussian mixture model (GMM). The linear probe consists of a fully connected layer whilst the mlp probe consists of two fully connected layers with a relu activation for the intermediate layer. Both probes were trained using an Adam optimizer with a learning rate of $3e^{-4}$ for 200 epochs with batch size fixed to 128. Scikit-learn's Gaussian Mixture model with a full covariance matrix and 200 initialization was fitted to the representations using the ground truth cluster number. The kNN algorithm from Python's Scikit-learn library was used with k spanning from 1 to 15 neighbors. The best performance was chosen as the final performance measurement. No augmentation strategy was used at evaluation.

A.7.6. GENERATION PROTOCOL

Here we detail the image generation protocol and the quality evaluation of generated samples.

Ad-hoc decoder training VAE-based approaches, including SimVAE, are fundamentally generative methods aimed at approximating the logarithm of the marginal likelihood distribution, denoted as $\log p(x)$. In contrast, most traditional self-supervised methods adopt a discriminative framework without a primary focus on accurately modeling $p(x)$. However, for the purpose of comparing representations, and assessing the spectrum of features present in z , we intend to train a decoder model for SimCLR & VicREG models. This decoder model is designed to reconstruct images from the fixed representations initially trained with these approaches. To achieve this goal, we train decoder networks using the parameter configurations specified in Tables 3b and 4b, utilizing the mean squared reconstruction error as the loss function. The encoder parameters remain constant, while we update the decoder parameters using an Adam optimizer with a learning rate of $1e^{-4}$ until a minimal validation loss is achieved (i.e. $\sim 10\text{-}80$ epochs).

Conditional Image Generation To allow for a fair comparison, all images across all methods are generated by sampling z from a multivariate Gaussian distribution fitted to the training samples' representations. More precisely, each Gaussian distribution is fitted to z conditioned on a label y . Scikit-Learn Python library Gaussian Mixture model function (with full covariance matrix) is used.

A.8. Additional Results & Ablations

Content classification evaluation with linear & gaussian mixture model prediction heads Table 5 reports the top-1% self-supervised classification accuracy using a linear prediction head and a gaussian mixture model. From Table 5, we draw similar conclusion as with Table 1: SimVAE significantly bridges the gap between discriminative and generative self-supervised learning methods when considering a supervised linear predictor and fully unsupervised methods for downstream prediction. Table 6 report the normalized mutual information (NMI) and adjusted rank index (ARI) for the fitting of the GMM prediction head.

Content & Style classification Figure 5 reports average classification accuracy using a MLP probe (over 3 runs) for the prediction of 20 CelebA facial attributes for SimVAE, generative and discriminative baselines.

Augmentation protocol strength ablation Figure 6 reports the downstream classification accuracy across methods for various augmentations strategies. More precisely, we progressively increase the cropping scale and color jitter amplitude. Unsurprisingly (Chen et al., 2020), discriminative methods exhibit high sensitivity to the augmentation strategy with stronger disruption leading to improved content prediction. The opposite trend is observed with vanilla generative methods where reduced variability amongst the data leads to increased downstream performance. Interestingly, SimVAE is robust to augmentation protocol and performs comparably across settings.

of augmentation ablation Figure 7 reports the downstream classification accuracy for increasing numbers of augmentations considered simultaneously during the training of SimVAE for MNIST and CIFAR10 datasets. On average, a larger

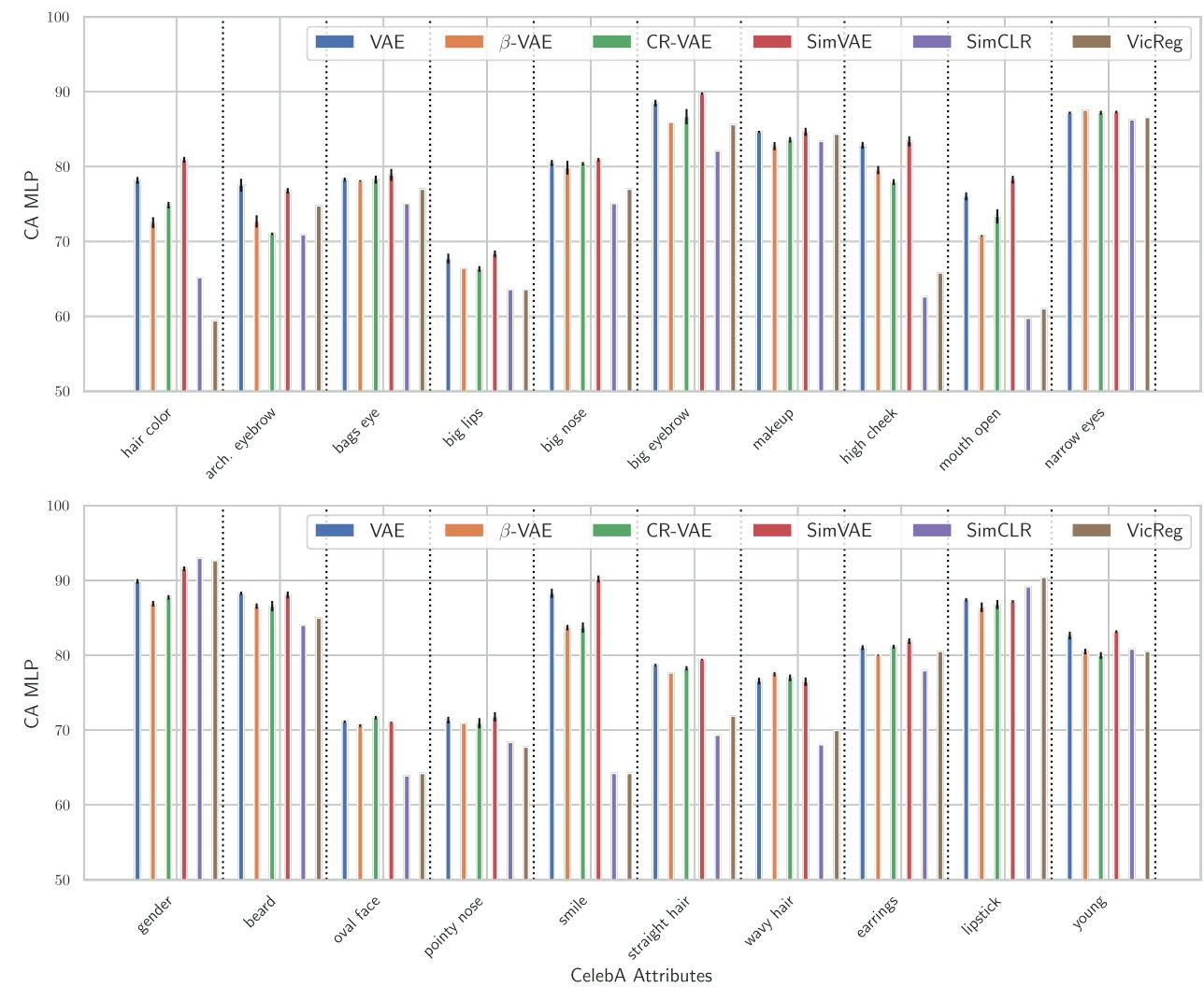


Figure 5: CelebA 20 facial attributes prediction using a MP. Average scores and standard errors are reported across 3 random seeds.

		Acc-LP	Acc-GMM		Acc-LP	Acc-GMM
MNIST	Random	39.7 ± 2.4	42.2 ± 1.2	CelebA	64.4 ± 0.9	59.2 ± 0.3
	SimCLR	96.8 ± 0.1	83.7 ± 0.6		94.2 ± 0.2	71.6 ± 0.6
	MoCo	88.6 ± 1.7	70.5 ± 4.0		MoCo	
	VicREG	96.7 ± 0.0	79.8 ± 0.6		VicREG 94.3 ± 0.3	53.9 ± 0.2
	VAE	97.2 ± 0.2	96.3 ± 0.4		VAE	81.5 ± 1.0
	β-VAE	97.8 ± 0.0	96.2 ± 0.2		β-VAE	81.9 ± 0.2
	CR-VAE	97.5 ± 0.0	96.9 ± 0.0		CR-VAE	81.6 ± 0.3
FashionMNIST	SimVAE	98.0 ± 0.0	96.6 ± 0.0		SimVAE	87.1 ± 0.3
	Random	51.2 ± 0.6	48.6 ± 0.2	CIFAR10	Random	15.7 ± 0.9
	SimCLR	73.0 ± 0.3	53.6 ± 0.3		SimCLR	65.4 ± 0.1
	MoCo	65.0 ± 1.3	56.6 ± 1.1		MoCo	53.3 ± 1.3
	VicREG	71.7 ± 0.1	60.2 ± 1.1		VicREG 68.2 ± 0.0	35.0 ± 2.8
	VAE	79.0 ± 0.5	57.9 ± 0.8		VAE	24.7 ± 0.4
	β-VAE	79.6 ± 0.0	68.0 ± 0.3		β-VAE	26.9 ± 0.0
	CR-VAE	79.7 ± 0.0	63.4 ± 0.4		CR-VAE	26.8 ± 0.0
	SimVAE	80.0 ± 0.0	71.1 ± 0.0		SimVAE	40.1 ± 0.0

Table 5: Top-1% self-supervised Acc (\uparrow) for MNIST, FashionMNIST, CIFAR10, and CelebA (gender classification) using a linear probe (LP) and Gaussian Mixture Model (GMM) classification methods; We report mean and standard errors over three runs; Bold indicate best scores in each method class: generative (teal), discriminative methods (red).

Dataset		VAE	β-VAE	CR-VAE	SimVAE	MoCo	VicREG	SimCLR
MNIST	ARI	89.0 ± 1.0	93.3 ± 0.3	94.0 ± 0.0	93.1 ± 0.0	58.3 ± 3.8	72.0 ± 0.7	77.4 ± 0.2
	NMI	94.9 ± 0.4	96.7 ± 0.2	96.9 ± 0.0	96.6 ± 0.0	71.4 ± 2.5	86.8 ± 0.4	89.6 ± 0.1
Fashion	ARI	44.3 ± 0.9	53.3 ± 0.4	47.6 ± 0.4	56.8 ± 0.0	30.9 ± 0.5	41.2 ± 0.5	33.2 ± 0.3
	NMI	69.1 ± 0.6	75.6 ± 0.1	72.6 ± 0.1	77.1 ± 0.0	50.4 ± 0.6	66.9 ± 0.3	62.1 ± 0.2
CelebA	ARI	5.7 ± 0.2	6.2 ± 0.7	6.6 ± 0.9	2.6 ± 0.7	—	18.7 ± 0.8	0.0 ± 0.1
	NMI	3.9 ± 0.2	4.7 ± 0.9	5.0 ± 0.7	2.9 ± 0.7	—	24.3 ± 0.3	0.0 ± 0.0
CIFAR10	ARI	0.6 ± 0.0	2.9 ± 0.1	2.0 ± 0.0	12.2 ± 0.1	27.2 ± 1.0	25.7 ± 0.2	52.2 ± 0.1
	NMI	31.7 ± 0.0	33.5 ± 0.1	32.4 ± 0.0	42.8 ± 0.1	16.5 ± 0.4	55.3 ± 0.1	21.7 ± 0.1

Table 6: Normalized mutual information (NMI) and Adjusted Rank Index (ARI) for all methods and datasets; Average scores and standard errors are computed across three runs

number of augmentations result in a performance increase. Further exploration is needed to understand how larger sets of augmentations can be effectively leveraged potentially by allowing for batch size increase. From Figure 7, we fix our number of augmentations to 10 across datasets.

Likelihood $p(x|z)$ variance ablation We explore the impact of the likelihood, $p(x|z)$, variance, σ^2 , across each pixel dimension on the downstream performance using the MNIST and CIFAR10 datasets. Figure 8 highlights how the predictive performance is inversely correlated with the σ^2 on the variance range considered for the CIFAR10 dataset. A similar ablation was performed on all VAE-based models and led to a similar conclusion. We therefore fixed σ^2 to 0.02 for β-VAE, CR-VAE and SimVAE across datasets.

Computational Resources Models for MNIST, FashionMNIST and CIFAR10 were trained on a RTX2080ti GPU with 11G RAM. Models for CelebA were trained on an RTX3090 GPU with 24G RAM. We observe that while the family of generative models requires more time per iteration, the loss convergence occurs faster while discriminative methods converge at a slower rate when considering the optimal set of hyperparameters. As a consequence, generative baselines and SimVAE were trained for 400 epochs while discriminative methods were trained for 600 to 800 epochs.

Image Generation We report the quality of images generated by SimVAE and considered baselines through visualisations (VAE-based approaches only) and quantitative measurements.

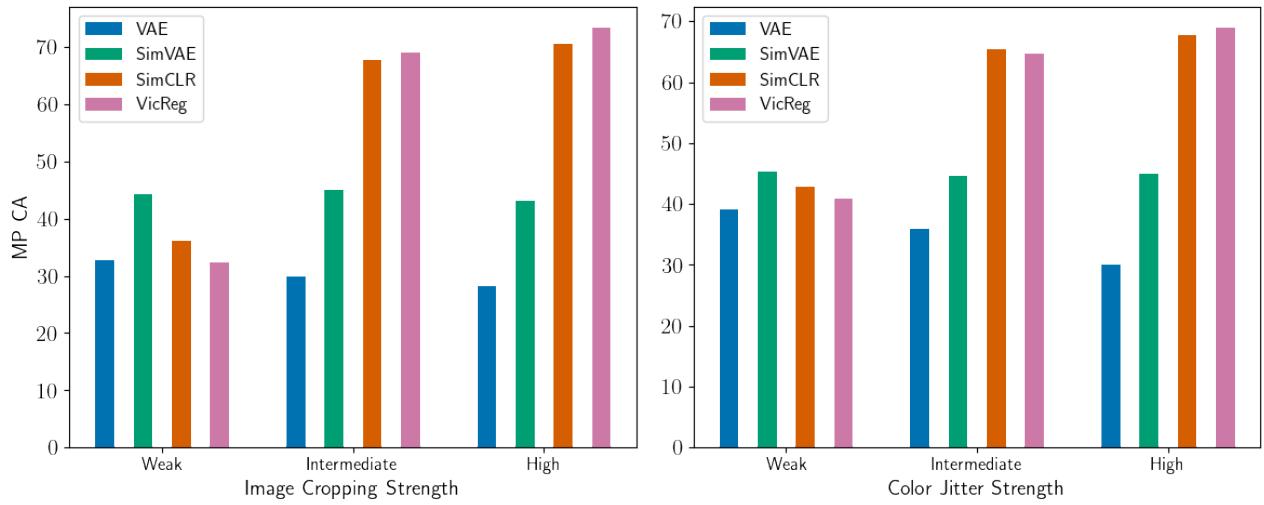


Figure 6: Ablation experiment across the augmentation strength for (left) image cropping and (right) color jitter strength considered during training of the SimVAE and baseline models using the CIFAR dataset.

Generated Images Figure 9 report examples of randomly generated images for each digit class and clothing item using the SimVAE trained on MNIST FashionMNIST, CIFAR10 and CelebA respectively.

Generative quality Table 7 reports the FID scores, reconstruction error for all generative baselines and SimVAE.

		MSE (\downarrow)	FID (\downarrow)
MNIST	VAE	0.029 ± 0.0	150.1 ± 0.2
	β -VAE	0.029 ± 0.0	155.3 ± 0.5
	CR-VAE	0.030 ± 0.0	153.0 ± 0.9
	SimVAE	0.026 ± 0.0	152.7 ± 0.3
Fashion	VAE	0.012 ± 0.0	99.4 ± 0.6
	β -VAE	0.008 ± 0.0	99.9 ± 0.7
	CR-VAE	0.008 ± 0.0	98.7 ± 0.0
	SimVAE	0.009 ± 0.0	96.1 ± 1.0
CelebA	VAE	0.016 ± 0.0	162.9 ± 2.8
	β -VAE	0.005 ± 0.0	163.8 ± 2.3
	CR-VAE	0.005 ± 0.0	159.3 ± 5.4
	SimVAE	0.004 ± 0.0	157.8 ± 2.3
CIFAR10	VAE	0.008 ± 0.0	365.4 ± 3.3
	β -VAE	0.004 ± 0.0	376.7 ± 1.7
	CR-VAE	0.004 ± 0.0	374.4 ± 0.4
	SimVAE	0.003 ± 0.0	349.9 ± 2.1

Table 7: Generation quality evaluated by: mean squared reconstruction error (RE), fréchet inception distance (FID). Mean and standard errors are reported across three runs.

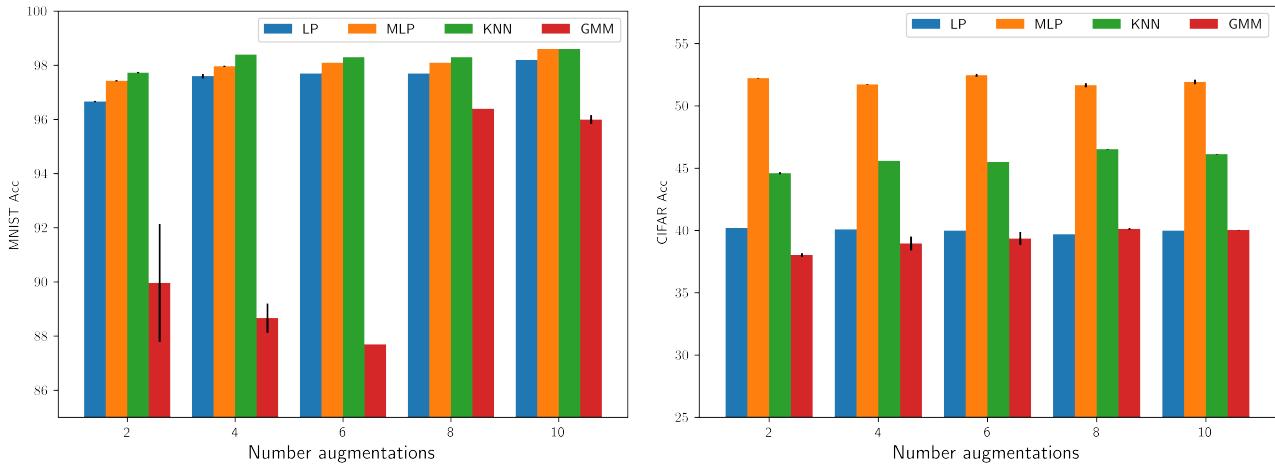


Figure 7: Ablation experiment across the number of augmentations considered during training of the SimVAE model using the MNIST (left) and CIFAR10 (right) datasets. Two, four, six, eight, and 10 augmentations were considered. The average and standard deviation of the downstream classification accuracy using Linear, MLP probes, and a KNN & GMM estimators are reported across three seeds. Batch size of 128 for all reported methods and number of augmentations. Means and standard errors are reported for three runs.

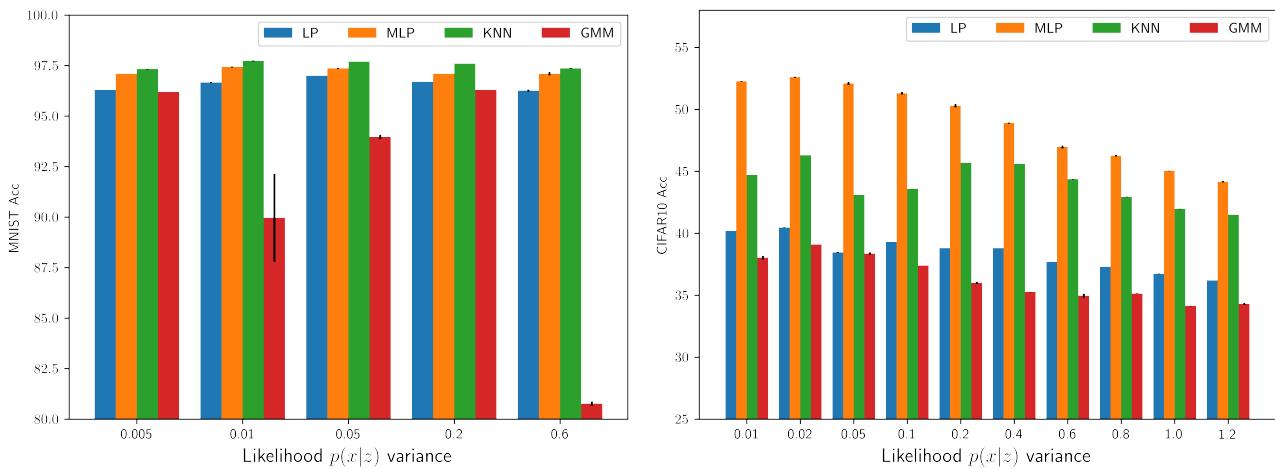


Figure 8: Ablation experiment across the likelihood $p(x|z)$ variance considered during training of the SimVAE model using the MNIST (left) and CIFAR10 (right) datasets. The average and standard deviation of the downstream classification accuracy using Linear, MLP probes and a KNN & GMM estimators are reported across three seeds. Means and standard errors are reported for three runs.

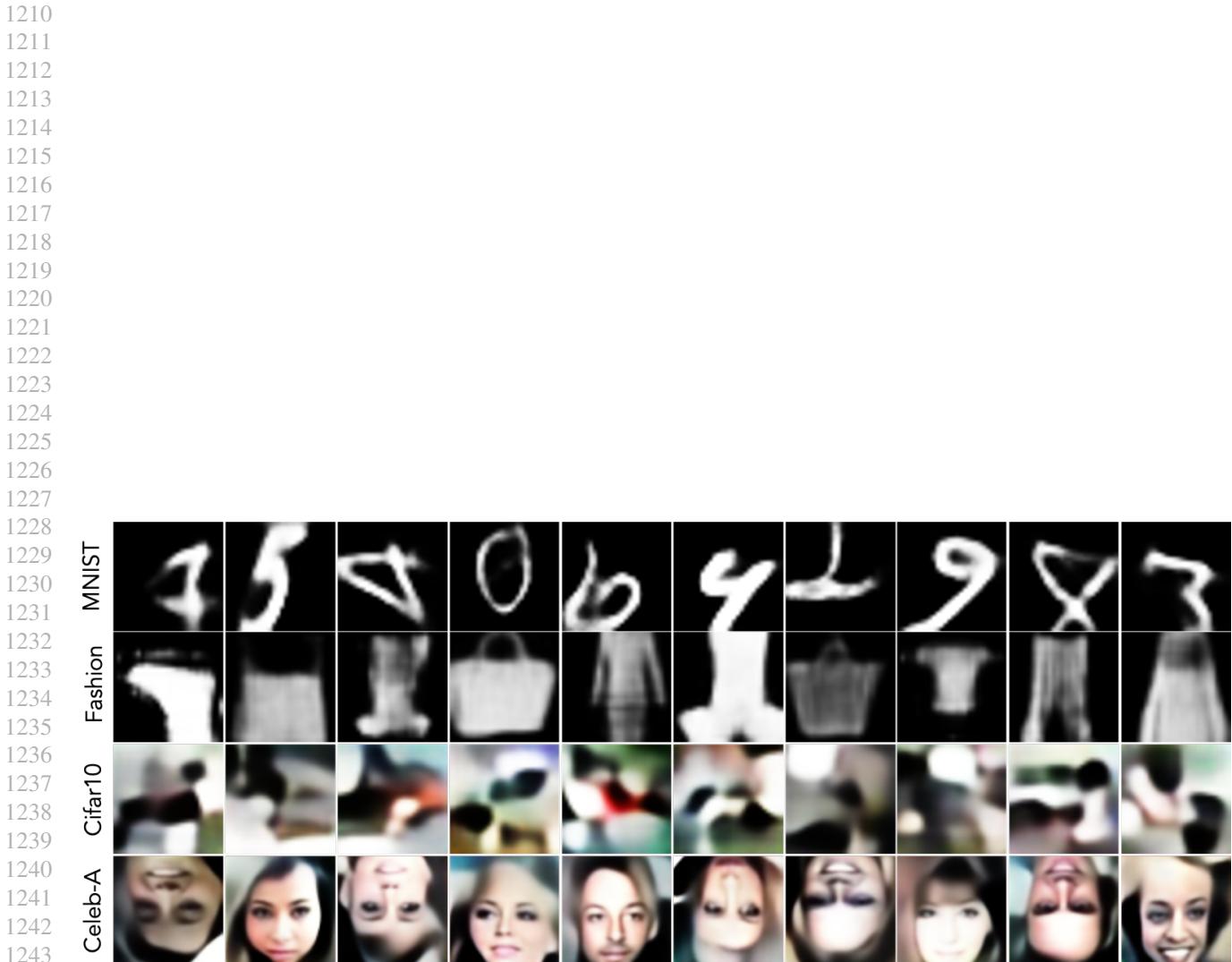


Figure 9: Samples generated from SimVAE model using MNIST, FashionMNIST, Cifar10 and CelebA training datasets