SimVAE: Narrowing the gap between Discriminative & Generative Representation Learning

Alice Bizeul **Carl Allen**
Department of Computer Science, ETH Zurich and ETH AI Center
{alice.bizeul, carl.allen}@ai.ethz.ch

Abstract

Self-supervised representation learning is a powerful paradigm that leverages the relationship between semantically similar data, such as augmentations, extracts of an image or sound clip, or multiple views/modalities. Recent methods, e.g. SimCLR, CLIP and DINO, have made significant strides, yielding representations that achieve state-of-the-art results on multiple downstream tasks. Though often intuitive, a comprehensive theoretical understanding of their underlying mechanisms or *what* they learn eludes. Meanwhile, generative approaches, such as variational autoencoders (VAEs), fit a specific latent variable model and have principled appeal, but lag significantly in terms of performance. We present a theoretical analysis of self-supervised discriminative methods and a graphical model that reflects the assumptions they implicitly make and unifies these methods. We show that fitting this model under an ELBO objective improves representations over previous VAE methods on several common benchmarks, narrowing the gap to discriminative methods, and can also preserve information lost by discriminative approaches. This work brings new theoretical insight to modern machine learning practice.

1 Introduction

Self-supervised learning (SSL) has become a prominent approach to unsupervised representation learning. Under this paradigm, a model is trained on an auxiliary task without annotated labels and representations of the data are learned in the process. Recently, *contrastive* SSL has achieved remarkable performance, exemplified by SimCLR [3], SWaV [2] and CLIP [18]. However, the success of contrastive SSL predominantly relies on heuristic strategies and intuitive but ad hoc design choices. A theoretical mechanism underlying their impressive performance remains unclear, limiting confidence in their reliability, restricting their interpretability and inhibiting their improvement.

To address these challenges, we consider the relationship between discriminative and generative representation learning and how an *encoder*, that maps data samples to representations, corresponds to the posterior distribution of a generative model, which it effectively reverses. We consider the implicit latent structure learned by several discriminative self-supervised algorithms, including the widely used InfoNCE loss function adopted by numerous other SSL models [e.g. 3, 18]. We show that these methods reflect a common hierarchical latent variable model, but do not fit its posterior due to their discriminative nature. Instead, representations of semantically related data "collapse" together, losing the (typically *stylistic*) information that distinguishes them, despite its potential use in downstream tasks. By prioritising some properties of the data over others, this effectively pre-supposes the set of downstream tasks and limits the generality of representations. We propose *SimVAE*, a *generative* alternative that explicitly models the latent structure implied by discriminative methods (Figure 2), introducing the implicit latent structure of SimCLR [3] into the variational auto-encoder (VAE) [13].

Mathematics of Modern Machine Learning Workshop at NeurIPS 2023.
Full paper available at https://openreview.net/forum?id=eJFBMqCE4X.

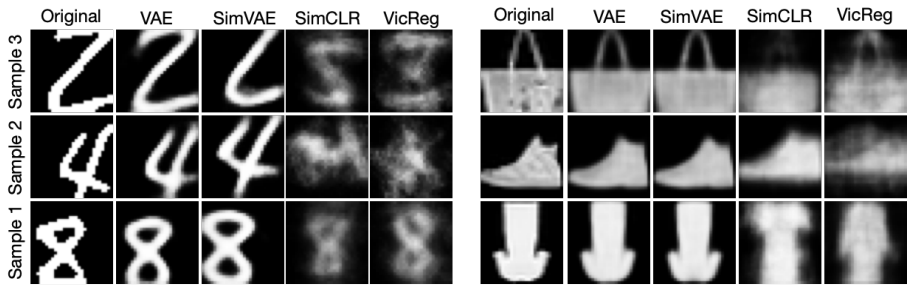

Figure 1: Qualitative assessment of representation information: images reconstructed from representations learned by unsupervised learning (VAE), generative SSL (our SimVAE) and discriminative SSL (SimCLR, VicReg). Datasets: MNIST (*l*), Fashion MNIST (*r*), original images in left columns.

Generative methods are inherently more challenging than their discriminative counterparts, and we should not necessarily expect to bridge the performance gap in one step. Encouragingly, SimVAE is competitive with, or outperforms, several discriminative methods on simple datasets, but is less competitive on more complex datasets, suggesting the need for further research. Importantly, SimVAE significantly outperforms (>10% on some metrics) previous VAE-based generative methods, narrowing the gap to discriminative SSL.

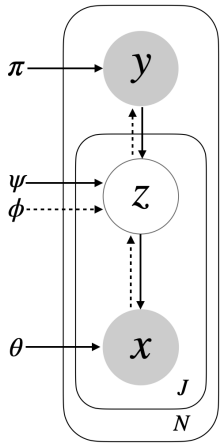

Figure 2: Graphical model for SSL; J related samples;

2 Background and Related Work

InfoNCE [17] extends the Word2Vec [16] approach to other domains where pairs (x, x') can be extracted "close by" or at random (e.g. image patches or sound clips). The InfoNCE loss for a data sample x, positive sample x^+ and L negative samples $X^- = \{x_l^-\}_{l=1}^L$ is given by:

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

where z is the representation of x and $\text{sim}(\cdot, \cdot)$ is a *similarity function*, e.g. dot product. A similar loss is proposed by [20]. Many works use the InfoNCE loss, e.g. using synthetic augmentations [3] or other modalities [18] of x as x^+; taking representations from different encoder layers [11]; or propose alternatives to negative sampling, e.g. MoCo [8], VicReg [1].

3 Representation Learning

Representations learning approaches are either discriminative or generative, with many recent self-supervised approaches being the former. *Discriminative* approaches tend to train the encoder under a loss function that induces geometric properties in the representation space that are intuitively desirable, e.g. for representations of related data samples to be "close" relative to random samples. A *generative* model $p(x) = \int_z p(x|z)p(z)$ can be interpreted as sampling a latent variable $z \sim p(\text{z})$ that defines the underlying characteristics of a data point; then sampling $x \sim p(\text{x}|z)$ to produce a manifestation of those properties. The posterior $p(z|x)$ effectively *reverses the generative process* to infer a distribution over z and thus the underlying semantic properties of x, reflecting uncertainty and providing a semantically meaningful representation of x.

In fact, the **generative and discriminative paradigms are two sides of the same coin** since any encoder f defines a posterior delta distribution $p(z|x) = \delta_{z-f(x)}$ (with zero uncertainty); and for any $z \in \mathcal{Z}$, a distribution $p(\text{x}|z)$ is implicitly defined by the probabilities of samples mapping to it $\{x \in \mathcal{X} \mid f(x) = z\}$. Generative approaches are more challenging but offer a principled, interpretable basis for representation learning, an uncertainty estimate over representations (i.e. the posterior), the ability to generate synthetic data, and insight into the information captured by representations from their regenerations.

We investigate the latent structure imposed by several discriminative methods, including InfoNCE [17], which underpins other methods such as SimCLR and CLIP. We posit a latent variable model (figure 2) to describe the latent structure those methods implicitly induce and propose a principled approach to learning it under an ELBO objective.

3.1 Discriminative Self-Supervised Learning

Instance Discrimination (ID) [5, 23] trains a classifier using the sample index $i \in [1, N]$ as a "label" for each sample and its augmentations (if any). The softmax cross-entropy loss can be viewed from a latent perspective (VC, §2) for the latent variable model i $\rightarrow$ z $\rightarrow$ x (*cf* Figure 2):

$$\log p(i|x_i^j) \geq \int_z q_\phi(z|x_i^j) \log p(i|z) \tag{2}$$

where j indexes an augmentation (inc. identity). By maximising Equation 2, representations of each "class" collapse together rather than fit any meaningful posterior giving *style-invariant* representations.

Deep Clustering (DC) [2] repeatedly identifies a large number of clusters in the latent space (by K-means) and uses temporary cluster assignments as pseudo-labels for discriminative training similar to ID. This induces similar latent structure as ID, and representations of each cluster collapse together.

InfoNCE-based Contrastive Learning approximates the softmax of ID, reducing memory/computational cost: (1) Under softmax cross entropy loss all representations of a class y converge to class parameter w_y. In expectation, $z^\top z'$, for stochastically sampled z' of the same class, approximates $z^\top w_y$, without the need to store w_y. (2) The softmax denominator is sampled, meaning optimal representations now satisfy $\text{sim}(z, z') = \text{PMI}(x, x') + c(x)$ [17]. However, using *bounded* cosine similarity $\text{sim}(z, z') = \frac{z^\top z'}{|z||z'|} \in [-1, 1]$ restricts that equality. Optimal representations are the same for related samples and maximally dispersed otherwise, comparable to those learned by softmax.

Summary: these different discriminative SSL methods *correspond to a common hierarchical latent variable model* (Figure 2). Being discriminative, they do not reflect a meaningful posterior, instead representations of semantically related data collapse together losing information that differentiates them (e.g. *style*), that downstream tasks may require. By preserving "class" at the expense of other information, **contrastive methods may *over-fit* representation learning to style-agnostic tasks**.

3.2 Generative Self-Supervised Learning (SimVAE)

Towards avoiding the pitfalls of discriminative approaches, we propose *SimVAE*, a generative approach to learn representations under the latent variable model posited from discriminative methods (Figure 2). Let $\mathbf{x} = \{x^j\}$, $\mathbf{z} = \{z^j\}$ with $j \in [1, J]$. The ELBO for J *semantically related* samples is given by:

$$\log p(\mathbf{x}) \geq \sum_j \int_{z^j} q(z^j|x^j) \log \tfrac{p(x^j|z^j)}{q(z^j|x^j)} + \int_{\mathbf{z}} q(\mathbf{z}|\mathbf{x}) \int_y q(y|\mathbf{z}) \Big[\sum_j \log \tfrac{p(z^j|y)}{q(y|z^j)} + \log p(y) \Big] \tag{3}$$

where $q(\mathbf{z}|\mathbf{x}) \approx \prod_j q(z^j|x^j)$. Note, we can choose *any* J to process multiple related samples.

4 Results

We focus on image cropping and color jitter (for natural images only) [3] to construct related samples. We perform standard evaluation of the learned representations (see appendix A.4 for details). Table 1 reports downstream supervised classification (using linear, MLP, KNN probes) and unsupervised (GMM) clustering accuracy across SSL methods and datasets. SimVAE is comparable to or outperforms disciminative and generative baselines on simple datasets, and significantly outperforms all VAE methods on natural images, including the self-supervised CRVAE. While a significant gap remains between SimVAE and disciminative methods on natural images, we significantly reduce the deficit compared to previous VAE-based methods.

Diversity of Encoded Information Figure 1 shows image reconstructions, using decoders trained post-hoc for discriminatively learned representations. This illustrates that style information (e.g., position and scale) is lost by discriminative SSL but preserved by generative methods. We quantitatively evaluate the ability to predict Celeb-A attributes, beyond gender classification shared in table 1, given that some relate to the augmentation strategy, e.g. hair color prediction requires color information. Table 2 shows that SimVAE outperforms other generative and all discriminative methods when style information is needed.

	LP-CA	MP-CA
Random	52.9 ± 0.4	51.2 ± 0.1
SimCLR	**65.3** ± 0.1	**65.7** ± 1.2
VicReg	62.7 ± 0.3	63.8 ± 0.5
VAE	75.4 ± 0.4	67.4 ± 0.4
SimVAE	**80.9** ± 0.5	**75.1** ± 0.3

Table 2: Celeb-A hair color CA. Mean accuracy and std error (3 random seeds)

			LP-CA	MP-CA	KNN-CA	GMM-CA
Fashion		Random	51.2 ± 0.6	49.8 ± 0.8	66.5 ± 0.4	48.6 ± 0.2
	Ⓓ	SimCLR	**77.4** ± 0.2	**79.0** ± 0.1	**79.3** ± 0.1	**63.6** ± 2.2
	Ⓓ	VicReg	70.7 ± 0.9	72.6 ± 0.6	76.0 ± 0.2	57.7 ± 0.8
	Ⓓ	MoCo	65.0 ± 1.3	71.2 ± 0.1	76.9 ± 0.2	56.6 ± 1.1
	Ⓖ	VAE	77.0 ± 0.5	80.2 ± 0.3	83.7 ± 0.2	57.9 ± 0.8
	Ⓖ	β-VAE ($\beta = 1.2$)	77.2 ± 0.1	79.7 ± 0.2	83.5 ± 0.4	57.5 ± 0.2
	Ⓖ	CR-VAE	77.7 ± 0.4	80.1 ± 0.1	**84.0** ± 0.2	67.5 ± 1.2
	Ⓖ	SimVAE	**78.6** ± 0.0	**81.1** ± 0.1	**84.0** ± 0.0	**69.9** ± 0.0
Celeb-A		Random	64.4 ± 0.9	65.3 ± 1.0	62.0 ± 0.9	59.2 ± 0.3
	Ⓓ	SimCLR	94.2 ± 0.2	92.7 ± 0.4	92.0 ± 0.3	**71.6** ± 0.6
	Ⓓ	VicReg	**94.3** ± 0.3	**94.7** ± 0.1	**92.7** ± 0.4	53.9 ± 0.2
	Ⓖ	VAE	81.5 ± 1.0	87.7 ± 0.5	79.6 ± 0.7	**58.8** ± 0.2
	Ⓖ	β-VAE ($\beta = 1.2$)	81.9 ± 0.2	86.7 ± 0.4	79.8 ± 0.1	**59.5** ± 0.6
	Ⓖ	CR-VAE	81.6 ± 0.3	87.7 ± 0.4	79.6 ± 0.6	**58.9** ± 0.4
	Ⓖ	SimVAE	**87.1** ± 0.3	**91.6** ± 0.4	**85.2** ± 0.1	**58.4** ± 0.6
CIFAR10		Random	15.7 ± 0.9	16.3 ± 0.4	13.1 ± 0.6	28.2 ± 0.2
	Ⓓ	SimCLR	65.2 ± 0.2	67.8 ± 0.2	65.2 ± 0.2	49.8 ± 2.8
	Ⓓ	VicReg	**68.8** ± 0.2	**69.6** ± 0.2	**68.2** ± 0.4	**54.3** ± 0.7
	Ⓓ	MoCo	53.3 ± 1.3	56.4 ± 1.6	54.0 ± 2.0	35.0 ± 2.8
	Ⓖ	VAE	24.7 ± 0.4	30.3 ± 0.4	25.6 ± 0.5	23.4 ± 0.7
	Ⓖ	β-VAE ($\beta = 1.2$)	24.4 ± 0.4	29.8 ± 0.2	25.1 ± 0.4	23.8 ± 0.4
	Ⓖ	CR-VAE	24.7 ± 0.4	30.4 ± 0.1	25.4 ± 0.4	23.9 ± 0.8
	Ⓖ	SimVAE	**36.4** ± 0.0	**45.5** ± 0.2	**42.8** ± 0.0	**34.7** ± 0.5

Table 1: Top-1% self-supervised CA (↑) for FashionMNIST, CIFAR10, and Celeb-A (i.e., gender classification) using a linear probe (LP), MLP probe (MP), k-Nearest Neighbors (KNN), and Gaussian Mixture Model (GMM) classification methods; Scores report the average and standard error across three random seeds; Bold numbers highlight the best scores with-in each method sub-group namely generative, Ⓖ , and discriminative methods, Ⓓ.

5 Discussion

We introduce the SimVAE training objective, based on the ELBO for a graphical model that embodies the assumptions implicit in several discriminative self-supervised methods. Our results validate this latent assumption and show the efficacy of SimVAE relative to previous VAE approaches, including CRVAE that aims for comparable latent structure. SimVAE demonstrably reduces the performance gap to discriminative SSL objectives, including those based on the popular InfoNCE objective.

SimVAE offers a more principled approach to modeling *sets* a of semantically related observations, facilitating the simultaneous representation of both content and style information, and taking a positive step towards fully task-agnostic representations. Additionally, the posterior provides an estimate of uncertainty, which may be important for critical downstream tasks, and the prior allows for explicit design choices, offering the prospect of separating latent factors to achieve disentangled representations.

While we consider SimVAE to be a positive advancement in representation learning, challenges remain in bridging the gap between generative and discriminative methods. Previous research shows that leveraging more complex model architectures, e.g. NVAE [22], StyleGAN [12], and CRVAE [19], can significantly enhance the ability of generative models. In this work, we hold the model architecture constant for fair comparison of the loss functions, but the additional complexity of generative methods and the increased information that representations are required to retain, may require more expressive architectures (e.g. [6]). Further, we note that increased variability of augmentations tend to improve discriminative methods but increase the challenge for generative approaches (e.g. see appendix A.5), suggesting a further direction for future investigation.

Acknowledgments and Disclosure of Funding

Alice and Carl are gratefully supported by ETH AI Centre PhD and Postdoctoral Fellowships (resp.). Carl is also supported by a small projects grant awarded by the Haslerstiftung (no. 23072).

References

[1] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021.

[2] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020.

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

[4] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

[5] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. *Advances in neural information processing systems*, 27, 2014.

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[8] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

[9] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[10] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International conference on learning representations*, 2017.

[11] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

[12] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410, 2019.

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

[14] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[15] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pp. 3730–3738, 2015.

[16] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013.

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

[18] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

[19] Samarth Sinha and Adji Bousso Dieng. Consistency regularization for variational auto-encoders. *Advances in Neural Information Processing Systems*, 34:12943–12954, 2021.

[20] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. *Advances in neural information processing systems*, 29, 2016.

[21] Michael Tschannen, Josip Djolonga, Paul K Rubenstein, Sylvain Gelly, and Mario Lucic. On mutual information maximization for representation learning. In *International Conference on Learning Representations*, 2020.

[22] Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *Advances in neural information processing systems*, 33:19667–19679, 2020.

[23] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3733–3742, 2018.

[24] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

A Appendix

A.1 Representation alignment in InfoNCE with cosine similarity

Pointwise mutual information (PMI) is a measurement of association that compares the probability of two events x and x' happening jointly with their probability of happening independently, defined as:

$$\text{PMI}(x, x') = \log \frac{p(x, x')}{p(x)p(x')} = \log \frac{p(x'|x)}{p(x')} \tag{4}$$

PMI values reflect, in log scale, the likelihood of observing x' having observed x relative to otherwise. In the case of synthetic augmentation, $p(x'|x) \gg p(x')$ if x' is an augmentation of x, and $p(x'|x) = 0$ otherwise, hence PMI(x, x') is a small positive value reflective of the number of augmentations, e.g. 5, or unboundedly negative.

The InfoNCE[17] objective is optimised when representations z, z' of samples x, x' satisfy $sim(z, z') = PMI(x, x') + c(x)$, where $sim(\cdot)$ is the similarity function, e.g. cosine similarity (sim$(z, z') = \frac{z^T z}{||z||_2 ||z'||_2}$), and c is a scalar that can vary with x. Us of the bounded popular cosine similarity function restricts the ability for the optimality condition to be reached, instead the optimization of this *restricted* InfoNCE objective leads to representations of similar data being aligned ($z = z'$) and representations of dissimilar data being maximally dispersed.

A.2 Relationship between Representations and PMI

When considering why representations learned by InfoNCE are useful, which intuitively pertains to the *information* they capture, the fact that the loss function is optimised when representations satisfy a relationship to pointwise mutual *information* seems highly relevant (§2). Even more so, since an analogous relationship underpins properties of word2vec learned word embeddings (§2). However, several further observations undermine this natural line of thought:

(i) Closer approximations of mutual information do not appear to improve representations [21];

(ii) As discussed in §3.1, employing **cosine similarity** sim(x,x') $= \frac{z^T z'}{|z||z'|} \in [-1, 1]$ often leads to better downstream performance than using *unbounded* similarity functions, e.g. dot product, even though PMI values can fall far outside the bounded range [-1,1]; and

(iii) Several recent self-supervised methods take a different contrastive approach, with the aim of circumventing negative sampling, showing no clear relationship to PMI and yet perform well [1].

A.3 Objective derivation

Let $\mathbf{x} = \{x^1, ..., x^j\}$, with $j \leq N$, be a set of N samples generated through augmentations, as described in section A.4. Let $\theta = \{\theta_x, \theta_z, \pi\}$ and $\phi = \{\phi_z, \phi_y\}$ be parameters of the model and approximate posterior, respectively. We derive the Evidence Lower Bound (ELBO) used as the SimVAE optimization objective and described in section 3.2 as:

$$
\begin{aligned}
\min_\theta D_{\mathrm{KL}}[\, p(\mathbf{x}) \,\|\, p_\theta(\mathbf{x}) \,] &= \max_\theta \mathop{\mathbb{E}}_{\mathbf{x}} \big[\log p_\theta(\mathbf{x}) \big] \\
&= \max_{\theta,\phi} \mathop{\mathbb{E}}_{\mathbf{x}} \Big[\int_{\mathbf{z}} \sum_y q_\phi(y, \mathbf{z}|\mathbf{x}) \log p_\theta(\mathbf{x}) \Big] \\
&= \max_{\theta,\phi} \mathop{\mathbb{E}}_{\mathbf{x}} \Big[\int_{\mathbf{z}} \sum_y q_\phi(y, \mathbf{z}|\mathbf{x}) \log p_\theta(\mathbf{x}) \tfrac{q_\phi(y,\mathbf{z}|\mathbf{x})}{q_\phi(y,\mathbf{z}|\mathbf{x})} \Big] \\
&= \max_{\theta,\phi} \mathop{\mathbb{E}}_{\mathbf{x}} \Big[\int_{\mathbf{z}} \sum_y q_\phi(y, \mathbf{z}|\mathbf{x}) \log \tfrac{p_{\theta_x}(\mathbf{x}|\mathbf{z}) p_{\theta_z}(\mathbf{z}|y) p_\pi(y)}{p_\theta(y,\mathbf{z}|\mathbf{x})} \tfrac{q_\phi(y,\mathbf{z}|\mathbf{x})}{q_\phi(y,\mathbf{z}|\mathbf{x})} \Big] \\
&= \max_{\theta,\phi} \mathop{\mathbb{E}}_{\mathbf{x}} \Big[\int_{\mathbf{z}} \sum_y q_\phi(y, \mathbf{z}|\mathbf{x}) \log \tfrac{p_{\theta_x}(\mathbf{x}|\mathbf{z}) p_{\theta_z}(\mathbf{z}|y) p_\pi(y)}{q_\phi(y,\mathbf{z}|\mathbf{x})} \Big] + D_{\mathrm{KL}}[\, q_\phi(y, \mathbf{z}|\mathbf{x}) \,\|\, p_\theta(y, \mathbf{z}|\mathbf{x}) \,] \\
&\geq \max_{\theta,\phi} \mathop{\mathbb{E}}_{\mathbf{x}} \Big[\int_{\mathbf{z}} \sum_y q_\phi(y, \mathbf{z}|\mathbf{x}) \log \tfrac{p_{\theta_x}(\mathbf{x}|\mathbf{z}) p_{\theta_z}(\mathbf{z}|y) p_\pi(y)}{q_\phi(y,\mathbf{z}|\mathbf{x})} \Big] \\
&= \max_{\theta,\phi} \mathop{\mathbb{E}}_{\mathbf{x}} \Big[\int_{\mathbf{z}} \sum_y q_{\phi_z}(\mathbf{z}|\mathbf{x}) p_{\phi_y}(y|\mathbf{z}) \log \tfrac{p_{\theta_x}(\mathbf{x}|\mathbf{z}) p_{\theta_z}(\mathbf{z}|y) p_\pi(y)}{q_{\phi_z}(\mathbf{z}|\mathbf{x}) q_{\phi_y}(y|\mathbf{z})} \Big] \\
&= \max_{\theta,\phi} \mathop{\mathbb{E}}_{\mathbf{x}} \Big[\int_{\mathbf{z}} q_{\phi_z}(\mathbf{z}|\mathbf{x}) \big\{ \log \tfrac{p_{\theta_x}(\mathbf{x}|\mathbf{z})}{q_{\phi_z}(\mathbf{z}|\mathbf{x})} + \sum_y q_{\phi_y}(y|\mathbf{z}) \log \tfrac{p_{\theta_z}(\mathbf{z}|y) p_\pi(y)}{q_{\phi_y}(y|\mathbf{z})} \big\} \Big] \\
&= \max_{\theta,\phi} \mathop{\mathbb{E}}_{\mathbf{x}} \underbrace{\int_{\mathbf{z}} q_{\phi_z}(\mathbf{z}|\mathbf{x}) \log p_{\theta_x}(\mathbf{x}|\mathbf{z})}_{-\mathrm{recon}(\mathbf{x})} \underbrace{- \int_{\mathbf{z}} q_{\phi_z}(\mathbf{z}|\mathbf{x}) \log q_{\phi_z}(\mathbf{z}|\mathbf{x})}_{H_{q_\phi(\mathbf{z}|\mathbf{x})}} \\
&\qquad + \int_{\mathbf{z}} q_{\phi_z}(\mathbf{z}|\mathbf{x}) \sum_y p_{\pi,\theta_z}(y|\mathbf{z}) \log p_{\theta_z}(\mathbf{z}|y) p_\pi(y)
\end{aligned}
$$

where recon$(\cdot)$ refers to the *reconstruction loss*, H to the entropy and D_{KL} to the KL-divergence. In the last step, we use $\max_{\phi_y} q_{\phi_y}(y|\mathbf{z}) = p_{\pi,\theta_z}(y|\mathbf{z}) \doteq \frac{p_{\theta_z}(\mathbf{z}|y) p_\pi(y)}{\sum_{y'} p_{\theta_z}(\mathbf{z}|y') p_\pi(y')}$ using Bayes' rule since y is assumed to be discrete in this case. In the setting with $N = 2$ related samples, $\mathbf{x} = \{x, x'\}$, the SimVAE objective can be formulated as:

$$
\begin{aligned}
\min_\theta D_{\mathrm{KL}}[\, p(\mathbf{x}) \,\|\, p_\theta(\mathbf{x}) \,] \geq \max_{\theta,\phi} \mathop{\mathbb{E}}_{\mathbf{x}} & \underbrace{\int_z q_\phi(z|x) \log p_{\theta_x}(x|z)}_{-\mathrm{recon}(x)} + \underbrace{\int_{z'} q_\phi(z'|x') \log p_{\theta_x}(x'|z')}_{-\mathrm{recon}(x')} \\
& \underbrace{- \int_z q_\phi(z|x) \log q_\phi(z|x)}_{H_{q_\phi(z|x)}} \underbrace{- \int_{z'} q_\phi(z'|x') \log q_\phi(z'|x')}_{H_{q_\phi(z'|x')}} \\
& + \int_{\mathbf{z}} q_\phi(\mathbf{z}|\mathbf{x}) \sum_y p_{\pi,\theta_z}(y|\mathbf{z}) \log p_{\theta_z}(\mathbf{z}|y) p_\pi(y)
\end{aligned}
$$

Algorithm 1 provides an overview of the main computational steps required for the training of the SimVAE evidence lower bound detailed above.

Algorithm 1 SimVAE

Require: data $\{\mathbf{x}_k\}_{k=1}^{M}$; batch size N; data dimension D; augmentation set $\mathcal{T}$; latent dimension L; number of augmentations A; encoder network $f_{\boldsymbol{\phi}}$; decoder network $g_{\boldsymbol{\theta}}$; prior variance $\{\sigma_l^*\}_{l=1}^{L}$

for randomly sampled mini-batch $\{\mathbf{x}_k\}_{k=1}^{N}$ **do**

augment mini-batch

$\{t_a\}_{a=1}^{A} \sim \mathcal{T}$;

$\{\mathbf{x}_k^a\}_{a=1}^{A} = \{t_a(\mathbf{x}_k)\}_{a=1}^{A}$;

forward pass : $\mathbf{z} \sim p(\mathbf{z}|\mathbf{x}), \tilde{\mathbf{x}} \sim p(\mathbf{x}|\mathbf{z})$

$\{(\boldsymbol{\mu}_k^a, \boldsymbol{\Sigma}_k^a) = f_{\boldsymbol{\phi}}(\mathbf{x}_k^a)\}_{a=1}^{A}$;

$\{\mathbf{z}_k^a \sim \mathcal{N}(\boldsymbol{\mu}_k^a, \boldsymbol{\Sigma}_k^a)\}_{a=1}^{A}$;

$\{\tilde{\mathbf{x}}_k^a = g_{\boldsymbol{\theta}}(\mathbf{z}_k^a)\}_{a=1}^{A}$;

compute & minimize loss terms

$\mathcal{L}_{\text{rec}}^k = \frac{1}{\boldsymbol{\sigma} ND} \sum_{a=1}^{A} \sum_{d=1}^{D} (x_{k,d}^a - \tilde{x}_{k,d}^a)^2$

$\mathcal{L}_{\text{H}}^k = L \log(2\pi e) + \frac{1}{2} \sum_{a=1}^{A} \log(|\boldsymbol{\Sigma}_k^a|)$

$\boldsymbol{\mu}_k^* = \frac{1}{A} \sum_{a=1}^{A} \mathbf{z}_k^a$

$\mathcal{L}_{\text{prior}}^k = N + AL \log(\sqrt{2\pi}) + A \sum_{l=1}^{L} \log(\sigma_l^*) + \sum_{a=1}^{A} \sum_{l=1}^{L} \frac{1}{2\sigma_l^*} (z_{k,l}^a - \mu_{k,l}^*)^2$

$\min(\mathcal{L} = \frac{1}{N} \sum_{k=1}^{N} \mathcal{L}_{\text{rec}}^k + \mathcal{L}_{\text{H}}^k + \mathcal{L}_{\text{prior}}^k)$ w.r.t $\boldsymbol{\phi}, \boldsymbol{\theta}$ by SGD;

end for

return $\boldsymbol{\phi}, \boldsymbol{\theta}$;

A.4 Experimental Details

A.4.1 Datasets

FashionMNIST The FashionMNIST dataset [24] 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.

CIFAR10 The CIFAR10 dataset [14] 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.

Celeb-A The Celeb-A dataset [15] 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 color, 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 color were aggregated into a 5-class attribute (i.e., bald, brown hair, blond hair, gray hair, black hair). Images with missing or ambiguous hair color information were discarded at evaluation.

All datasets were sourced from Pytorch's dataset collection.

A.4.2 Data augmentation strategy

Taking inspiration from SimCLR's [3] augmentation strategy which highlights the importance of random image cropping and color jitter on downstream performance, our augmentation strategy includes random image cropping, random image flipping and random color jitter. The color augmentations are only applied to the non gray-scale datasets (i.e., CIFAR10 [14] & Celeb-A dataset [15]). Due to the varying complexity of the datasets we explored, hyperparameters such as the cropping strength were

adapted to each dataset to ensure that semantically meaningful features remained after augmentation. The augmentation strategy hyperparameters used for each dataset are detailed in table 3.

Dataset	**Crop**		**Vertical Flip**	**Color 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
Celeb-A	0.6	[0.75,1.3]	0.5	0.8	0.2	0.8

Table 3: Data augmentation strategy for each dataset: (from left to right) cropping scale, cropping ratio, probability of vertical and horizontal flipping, brightness-saturation-contrast jitter strength, hue jitter strength, probability of color jitter

A.4.3 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.

Datasets and Evaluation Metrics We evaluated SimVAE on three benchmark datasets including two with natural images: FashionMNIST [24], Celeb-A [15] and CIFAR10 [14]. We augment images following the SimCLR [3] protocol which includes cropping and flipping as well as color jitter for natural images. We evaluate representations' utility for downstream classification tasks using a linear probe, a non-linear MLP probe, and k-nearest neighbors (kNN) [4] trained on the pre-trained frozen representations using image labels [3, 2]. Additionally, we conducted a fully unsupervised evaluation by fitting a Gaussian mixture model (GMM) to the frozen features for which the number of clusters was set to its ground-truth value. Downstream performance is measured in terms of classification accuracy (CA). A model's generative quality was evaluated using the Fréchet Inception Distance (FID) [9], reconstruction error as well as the Normalized Mutual Information (NMI) and Adjusted Rank Index (ARI) clustering scores (see appendix A.5).

Baselines methods We compare SimVAE to other VAE-based models including the vanilla VAE [13], β-VAE [10] and CR-VAE [19], as well as to state-of-the-art self-supervised discriminative methods including SimCLR [3], VicREG [1], and MoCo [8]. As a lower bound, we also provide results obtained for randomly initialized embeddings. To ensure fair comparison, the augmentation strategy, representation dimensionality, batch size, and encoder-decoder architectures were kept 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.4.3 and A.4.4 for further details on training baselines and decoder models.

Hyperparameters We use MLP and Resnet18 [7] network architectures for simple and natural image datasets respectively. We fix the dimension of representations z to 10 for FashionMNIST, and to 64 for Celeb-A and CIFAR10 datasets. For all generative approaches, we adopt Gaussian posteriors, priors, and likelihoods, employing diagonal covariance matrices as in [13]. We fix covariances of the prior and likelihood distributions and perform a hyper-parameter search. SimVAE conveniently allows for the simultaneous incorporation of sets of related observations. After tuning, we fix the number of augmentations to 6 (see Figure 4 for an ablation). For baselines, all sensitive hyperparameters were tuned independently for each dataset and method.

Network Architectures The encoder network architectures used for SimCLR, MoCo, VicReg, and VAE-based approaches including SimVAE for simple (i.e., FashionMNIST) and complex datasets (i.e., CIFAR10, Celeb-A) are detailed in table 4a, table 5a respectively. Generative models which include all VAE-based methods also require decoder networks for which the architectures are detailed in table 4b and table 5b. The encoder and decoder architecture networks are kept constant across methods including the latent dimensionality to ensure a fair comparison across methods.

Optimisation & Hyper-parameter tuning All methods were trained using an Adam optimizer until training loss convergence. A learning rate tuning was performed for each method independently

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

(a) Encoder

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

(b) Decoder

Table 4: Multi-layer perceptron network architectures used for 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 5: Resnet18 network architectures used for CIFAR10 & Celeb-A training

across the range $1e^{-3}$ to $8e^{-5}$. A fixed batch size of 128 was used across methods and datasets. The β, τ, λ parameters for the β-VAE, SimCLR and CRVAE methods were tuned across the [0.1,0.2,0.5], [0.1,0.5,1.0] and [0.01,0.1,1.0] ranges respectively based on downstream performance. $\beta = 0.1$, $\lambda = 0.01$ were selected and $\tau = 1.0$, $\tau = 0.5$ were chosen for simple and natural datasets respectively. The likelihood probability variance for VAE-based methods including SimVAE was kept to $\sigma^2 = 1.0$ and the prior probability, $p(z|y)$, variance parameter for SimVAE was tuned and fixed to 0.003, 0.005, 0.005 for FashionMNIST, CIFAR10 and Celeb-A respectively.

A.4.4 Evaluation Implementation Details

Following common practices [3], 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.4.5 Generation Protocol

In this section, we detail the image generation protocol as well as the evaluation of the quality of the 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 4b and 5b, 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 convergence is achieved (i.e. $\sim$ 200 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.5 Additional Results

A.5.1 Self-supervised classification

Clustering metrics Table 6 and table 7 report the normalized mutual information (NMI) and adjusted rank index (ARI) for the fitting of a GMM to latent representations z.

Dataset		Random	VAE	β-VAE	CR-VAE	SimVAE
Fashion	ARI	28.7 ± 0.6	44.2 ± 1.1	44.7 ± 0.2	23.3 ± 0.8	$\mathbf{55.7 \pm 0.0}$
	NMI	51.5 ± 0.2	66.7 ± 0.7	66.4 ± 0.4	46.1 ± 2.2	$\mathbf{76.8 \pm 0.2}$
Celeb-A	ARI	3.4 ± 0.3	5.7 ± 0.2	6.2 ± 0.7	6.6 ± 0.9	2.6 ± 0.7
	NMI	4.2 ± 0.4	3.9 ± 0.2	4.7 ± 0.9	5.0 ± 0.7	2.9 ± 0.7
CIFAR10	ARI	0.09 ± 0.0	0.7 ± 0.2	0.7 ± 0.2	0.9 ± 0.1	$\mathbf{8.6 \pm 0.3}$
	NMI	27.9 ± 0.1	17.7 ± 0.5	18.7 ± 0.3	18.9 ± 0.1	$\mathbf{37.2 \pm 0.4}$

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

Dataset		MoCo	VicReg	SimCLR
Fashion	ARI	30.9 ± 0.5	37.1 ± 1.3	$\mathbf{50.3 \pm 1.9}$
	NMI	50.4 ± 0.6	64.5 ± 0.7	$\mathbf{71.2 \pm 1.0}$
Celeb-A	ARI	–	$\mathbf{18.7} \pm 0.8$	0.0 ± 0.1
	NMI	–	$\mathbf{24.3} \pm 0.3$	0.0 ± 0.0
CIFAR10	ARI	27.2 ± 1.0	31.2 ± 0.2	$\mathbf{49.6 \pm 1.3}$
	NMI	16.5 ± 0.4	$\mathbf{53.4 \pm 0.1}$	26.9 ± 0.8

Table 7: Normalized mutual information (NMI) and Adjusted Rank Index (ARI) for all discriminative baselines and datasets; Average scores and standard errors are computed across three random seeds

Augmentation Protocol Strength Figure 3 reports the downstream CA across methods for various augmentations stategy. More precisely, we progressively increase the cropping scale and color jitter amplitude. Unsurprinsingly [3], 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.

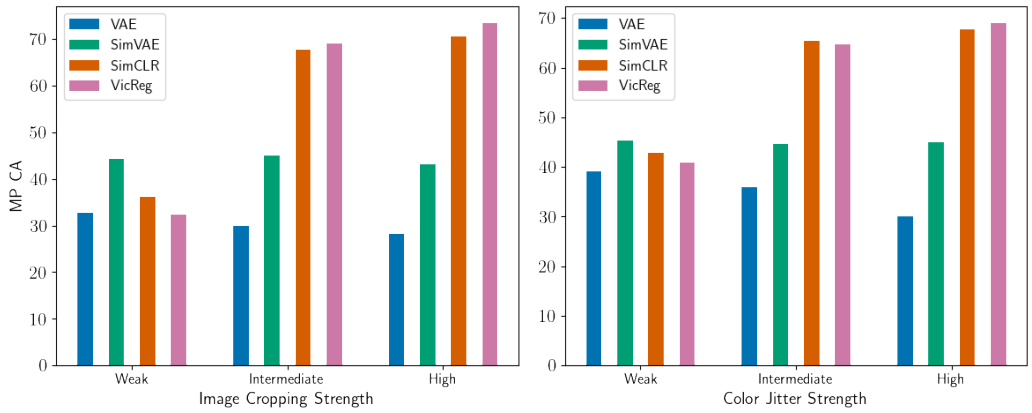

Figure 3: Ablation experiment across the number of augmentations considered during training of the SimVAE model using the MNIST (left) and FashionMNIST (right) datasets. Two, four, six and eight augmentations were considered. The average and standard deviation of the downstream classification accuracy using KNN and GMM probes are reported across three seeds.

 # Augmentation Ablation Figure 4 reports the downstream classification accuracy for increasing numbers of augmentations considered simultaneously during the training of SimVAE. A larger number of augmentations result in a performance increase up to a certain limit (i.e., 6-8 augmentations). Further exploration is needed to understand how larger sets of augmentations can be effectively leveraged potentially by allowing for batch size increase.

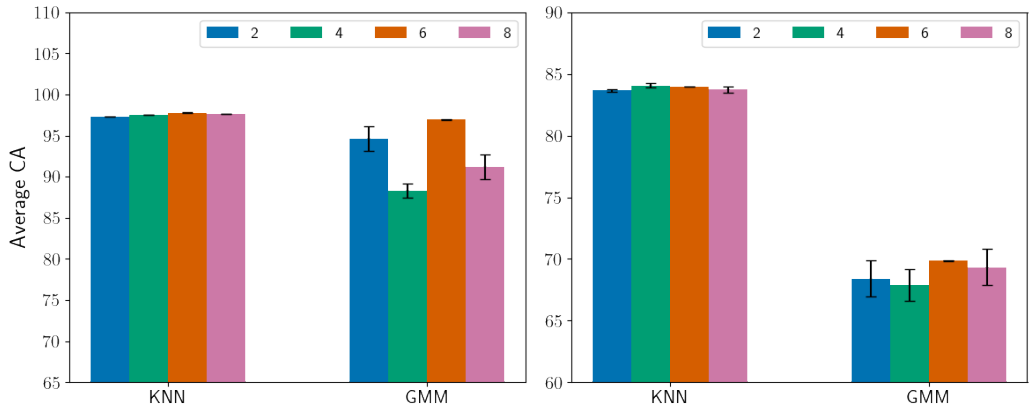

Figure 4: Ablation experiment across the number of augmentations considered during training of the SimVAE model using the MNIST (left) and FashionMNIST (right) datasets. Two, four, six and eight augmentations were considered. The average and standard deviation of the downstream classification accuracy using KNN and GMM probes are reported across three seeds. Batch size of 128 for all reported methods and number of augmentations.

A.5.2 Image Generation

In this section, we explore and report the quality of images generated through SimVAE and all considered baselines through visualisations (for VAE-based approaches only) and quantitative measurements.

Generated Images Figure 5 report examples of randomly generated images for each digit class and clothing item using the SimVAE trained on MNIST and FashionMNIST respectively.

Figure 5: Conditional sampling for each one of the FashionMNIST clothing type using pre-trained SimVAE model

		RE	FID	NLL
Fashion	VAE	4.4 ± 0.1	99.4 ± 0.6	5696.5 ± 0.1
	β-VAE	4.6 ± 0.1	99.9 ± 0.7	5696.7 ± 0.1
	CR-VAE	4.3 ± 0.0	98.7 ± 0.0	5696.7 ± 0.0
	SimVAE	$\mathbf{3.4} \pm 0.1$	$\mathbf{96.1} \pm 1.0$	5695.6 ± 0.0
Celeb-A	VAE	56.6 ± 0.2	162.9 ± 2.8	$-$
	β-VAE	60.3 ± 1.0	163.8 ± 2.3	$-$
	CR-VAE	57.4 ± 0.1	$\mathbf{159.3} \pm 5.4$	$-$
	SimVAE	$\mathbf{35.3} \pm 0.2$	$\mathbf{157.8} \pm 2.3$	$-$
CIFAR10	VAE	$\mathbf{21.4} \pm 0.2$	365.4 ± 3.3	22330.8 ± 0.2
	β-VAE	22.3 ± 0.2	376.7 ± 1.7	22327.7 ± 0.2
	CR-VAE	22.5 ± 0.0	374.4 ± 0.4	22327.3 ± 0.8
	SimVAE	22.1 ± 0.1	$\mathbf{349.9} \pm 2.1$	22327.3 ± 0.2

Table 8: Generation quality evaluation of all generative methods across three random seeds: (from left to right) mean squared reconstruction error (RE, $\downarrow$), fréchet inception distance (FID, $\downarrow$), negative log-likelihood (NLL,$\downarrow$)

Generative quality Table 8 reports the FID scores, reconstruction error and approximate negative log-likelihoods using 1000 importance-weighted samples for all generative baselines and SimVAE.