The Triad of Failure Modes and a Possible Way Out

Emanuele Sansone
Department of Computer Science, KU Leuven
Leuven, Belgium

Abstract

We present a novel objective function for cluster-based self-supervised learning (SSL) that is designed to circumvent *the triad of failure modes*, namely representation collapse, cluster collapse, and the problem of invariance to permutations of cluster assignments. This objective consists of three key components: (i) A generative term that penalizes representation collapse, (ii) a term that promotes invariance to data augmentations, thereby addressing the issue of label permutations and (ii) a uniformity term that penalizes cluster collapse. Moreover, our proposed objective possesses two notable advantages. Firstly, it can be interpreted from a Bayesian perspective as a lower bound on the data log-likelihood. Secondly, it enables the training of a standard backbone architecture without the need for asymmetric elements like stop gradients, momentum encoders, or specialized clustering layers. Due to its simplicity and theoretical foundation, our proposed objective is well-suited for optimization. Experiments on both toy and real world data demonstrate its effectiveness.

1 Background

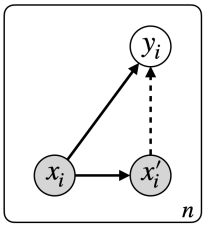


Figure 1: Probabilistic graphical model for cluster-based SSL. i is used to index different training instances, i.e. $i = 1, \dots, n$.

Model. Let us introduce the random quantities used in the model shown in Figure 1: (i) $x \in \Omega$, where Ω is a compact subset of $\mathbb{R}^d$, represents a data vector drawn independently from an unknown distribution $p(x)$ (for instance an image), (ii) $x' \in \Omega$ represents a transformed version of x using a stochastic data augmentation strategy $\mathcal{T}(x'|x)$ (obtained by adding for instance noisy or cropping the original image), and (iii) $y \in \{1, \dots, c\}$ is the symbolic representation of an input data point defined over c categories (namely the cluster label obtained by an output layer defined over the embedding representation). The corresponding probabilistic graphical model is given in Figure 1. The generative process (solid arrows) is defined using the following conditional densities, namely: $p(x'|x, \xi) = \mathcal{T}(x'|x)$ and $p(y|x) = \text{Softmax}(out(proj(enc(x))))$, where $enc : \Omega \to \mathbb{R}^h$ is an encoder used to compute the latent representation, $proj : \mathbb{R}^h \to \mathcal{S}^{h-1}$ is a projector head used to compute the embedding representation, and out computes the cosine similarity between the embedding representation and the column vectors of a matrix of parameters $U \in \mathbb{R}^{h \times c}$ known as the cluster centers/prototypes [2]. The inference process (dashed arrow) is defined as $q(y|x) = \text{SK}(out(proj(enc(x'))))$, viz. a distribution over cluster/prototype assignments obtained through the Sinkhorn-Knopp algorithm (SK). Please refer to [2] for additional details.

NeurIPS 2023 Workshop: Self-Supervised Learning - Theory and Practice.

Objective. The training objective is based on an evidence lower bound on the negative entropy, derived from the probabilistic graphical model of Figure 1(a), namely:

$$\mathbb{E}_{p(x_{1:n})}\{\log p(x_{1:n};\Theta)\} = -H_p(x_{1:n}) + \mathbb{E}_{p(x_{1:n})\mathcal{T}(x'_{1:n}|x_{1:n})}\left\{\log \sum_{y_{1:n}} p(y_{1:n}|x_{1:n};\Theta)\right\} \tag{1}$$

$$\geq -H_p(x_{1:n}) + \underbrace{\sum_{i=1}^{n} \mathbb{E}_{p(x_i)\mathcal{T}(x'_i|x_i)}\left\{\mathbb{E}_{q(y_i|x'_i)} \log p(y_i|x_i;\Theta) + H_q(y_i|x'_i)\right\}}_{\text{Discriminative term } \mathcal{L}_{DI}(\Theta)} \tag{2}$$

where $H_q(y|x')$ is the entropy computed over $q(y|x')$ and Θ includes all parameters of the encoder, projector head and the output layer of the discriminative model. Intuitively, the first addend in $\mathcal{L}_{DI}(\Theta)$ in Eq. 2 forces the symbolic representations of the input data and its augmented version to be similar, whereas the second addend enforces uniformity on the cluster assignments, so as to avoid that all representations collapse to a single cluster. It is important to mention that the objective in Eq. 2 is general enough to cover several proposed criteria in the literature of cluster-based self-supervised learning (cf. [15, 14]), such as DeepCluster [1], SwAV [2] and DINO [3].

2 Objective Function and The Triad of Failure Modes

We devise a new lower bound for cluster-based SSL which avoids introducing asymmetries in the optimization procedure and in the discriminative backbone. We theoretically analyze the properties of the different loss terms involved in the GEDI instantiation with respect to important failure modes.

We are ready to state the following proposition (the proof can be found in Appendix A of the Supplementary Material):

Proposition 2.1. *Eq. (1) can be lower bounded by the following quantity:*

$$-H_p(x_{1:n}) \underbrace{- \sum_{i=1}^{n} \mathbb{E}_{p(x_i)\mathcal{T}(x'_i|x_i)}\left\{CE(p(y_i|x'_i;\Theta), p(y_i|x_i;\Theta))\right\}}_{\mathcal{L}_{INV}(\Theta)} \underbrace{- \sum_{i=1}^{n} CE\left(p(y_i), q(y_i)\right)}_{\mathcal{L}_{PRIOR}(\Theta)} \tag{3}$$

with $q(y) = \frac{1}{n}\sum_{j=1}^{n} p(y_j = y|x_j;\Theta)$ and CE the cross-entropy loss. Additionally, the corresponding maximum value for the last two addends in Eq. (3) is given by the following inequality:[1]

$$\mathcal{L}_{INV}(\Theta) + \mathcal{L}_{PRIOR}(\Theta) \leq -H_p(y_{1:n}) \tag{4}$$

The above proposition has interesting implications. First of all, by maximizing the discriminative term $\mathcal{L}_{INV}(\Theta)$ with respect to Θ, we enforce two properties, namely: (i) label invariance, as we ensure that the predictive distributions of the discriminative model for a sample and its augmented version match each other and (ii) confident predictions, as maximizing the cross-entropy forces also to decrease the entropy of these distributions.[2] Secondly, by choosing a uniform prior, viz. $p(y_i) = \text{Uniform}(\{1,\dots,c\})$, and by maximizing $\mathcal{L}_{PRIOR}(\Theta)$ with respect to Θ, we ensure to obtain a balanced cluster assignment, typical of approaches based on optimal transport objectives and corresponding surrogates [1, 2, 4]. Finally, the proposed lower bound allows for an important key difference over existing cluster-based SSL, as we don't need to introduce asymmetries in the discriminative backbones. Indeed, we note that cluster-based SSL, specifically SwAV, assume $p(y|x;\Theta) = \text{Softmax}(U^T g(x)/\tau)$ and $q(y|x') = \text{Sinkhorn}(\text{StopGrad}(U^T g(x')/\tau))$, where Sinkhorn and StopGrad are two operators performing the Sinkhorn-Knopp algorithm and stopping the gradients, respectively. In contrast, we require that $q(y|x) = p(y|x;\Theta) = \text{Softmax}(f(enc(x))/\tau)$, where $f : \mathbb{R}^h \to \mathbb{R}^c$ is a simple discriminative network head.

Additionally, we lower bound the first addend in Eq. 3 by exploiting the inequality $-H_p(x_{1:n}) \geq -CE(p, p_\Theta)$, and obtain the overall objective, called GEDI (aka GEnerative DIscriminative objective):

$$\mathbb{E}_{p(x_{1:n})}\{\log p(x_{1:n};\Theta)\} \geq \underbrace{\mathcal{L}_{GEN}(\Theta)}_{\text{GEnerative term } -CE(p,p_\Theta)} + \underbrace{\mathcal{L}_{INV}(\Theta) + \mathcal{L}_{PRIOR}(\Theta)}_{\text{DIscriminative terms}} \tag{5}$$

[1] Here, we assume that the predictive model $p(y|x;\Theta)$ has enough capacity to achieve the optimal solution.

[2] Indeed, recall that $CE(p,q) = H_p + KL(p\|q)$. Therefore, maximizing $-CE(p,q)$ forces to have both $KL(p\|q) = 0$ and $H_p = 0$.

Importantly, we can reinterpret the discriminative model $p(y|x;\Theta) = \frac{p(y,x;\Theta)}{p(x;\Theta)}$ as an energy-based generative model $p_\Theta = p(x;\Theta)$, similarly to what is done in the context of supervised learning [8, 10], namely:

$$p(y,x;\Theta) = \frac{e^{f_y(enc(x))/\tau}}{\Gamma(\Theta)} \qquad p_\Theta \doteq p(x;\Theta) = \frac{\sum_{y=1}^{c} e^{f_y(enc(x))/\tau}}{\Gamma(\Theta)} = \frac{e^{\log \sum_{y=1}^{c} e^{f_y(enc(x))/\tau}}}{\Gamma(\Theta)} \tag{6}$$

Training is performed by simply maximizing the lower bound in Eq 5. We leave detailed discussion about the training and its computational requirements to Appendix B in the Supplementary Material. We are now ready to analyze the properties of the GEDI objective.

The Triad of Failure Modes. Here, we formalize three main failure modes for cluster-based SSL [16]. Then, we study the GEDI loss landscape and show that these undesired trivial solutions are not admitted by our objective. This result holds without introducing asymmetries in the optimization procedure and/or network architecture.

Let's start by defining the most important failure modes, namely:

Definition 1 (Failure Mode 1 - Representational Collapse)**.** *There exists a constant vector $k \in \mathbb{R}^h$ such that for all $x \in \mathbb{R}^d$, $enc(x) = k$.*

Definition 2 (Failure Mode 2 - Cluster Collapse)**.** *There exists a cluster $j \in \{1, \dots, c\}$ such that for all $x \in \mathbb{R}^d$, $p(y = j|x;\Theta) = 1$.*

Definition 3 (Failure Mode 3 - Permutation Invariance to Cluster Assignments)**.** *For all possible permutations $\pi : \{1, \dots, c\} \to \{1, \dots, c\}$, a dataset $\mathcal{D} = \{(x_i, t_i, t_i')\}_{i=1}^n$, its permuted version $\mathcal{D}^\pi = \{(x_i, t_{\pi(i)}, t_i')\}_{i=1}^n$ and a loss $\mathcal{L}(\Theta;\cdot)$, evaluated at one of the two datasets, we have that $\mathcal{L}(\Theta;\mathcal{D}) = \mathcal{L}(\Theta;\mathcal{D}^\pi)$. For GEDI, $t_i \doteq f(enc(x_i))$ and $t_i' \doteq f(enc(x_i'))$.*

In other words, Definition 1 considers the case where the encoder maps (collapses) every input to the same output. Definition 2 considers the situation where the predictive model assigns all samples to the same cluster with high confidence. And Definition 3 considers the case where a hypothetical adversary swaps the predictions made by the model on different pair of inputs. Ideally, we would like to have an objective that does not admit these failure modes.

Now, we state the properties of the loss landscape of GEDI with the following theorem (we leave the proof to Section G in the Supplementary Material):

Theorem 1. *Given definitions 1-3, the following statements tells for a particular loss, which modes are admitted as optimal solutions:*

a. $\mathcal{L}_{GEN}(\Theta)$ admits failure modes 2 and 3.

b. $\mathcal{L}_{INV}(\Theta)$ admits failure modes 1 and 2.

c. $\mathcal{L}_{PRIOR}(\Theta)$ admits failure modes 1 and 3.

Importantly, Theorem 1 tells us that $\mathcal{L}_{GEN}(\Theta)$ can be used to penalize representational collapse, $\mathcal{L}_{INV}(\Theta)$ can be used to break the problem of permutation invariance for the cluster assignments, while $\mathcal{L}_{PRIOR}(\Theta)$ can be used to penalize cluster collapse. Consequently, by maximizing the objective in Eq. (5), we are guaranteed to learn solutions which are non-trivial. A table summarizing all these properties is given below.

Table 1: Summary of loss landscape

Does ↓ penalize →?	**Repr. collapse**	**Clus. collapse**	**Perm. Inv.**
$\mathcal{L}_{GEN}(\Theta)$	**Yes**	No	No
$\mathcal{L}_{INV}(\Theta)$	No	No	**Yes**
$\mathcal{L}_{PRIOR}(\Theta)$	No	**Yes**	No
Eq. (5)	**Yes**	**Yes**	**Yes**

Table 2: Clustering performance based on normalized mutual information (NMI) on test set (toy data, viz. moons and circles, and real data, viz. SVHN, CIFAR-10, CIFAR-100). Higher values indicate better clustering performance. Mean and standard deviations are computed from 5 different runs.

Dataset	**JEM** [8]	**SwAV** [2]	**GEDI no unif**	**GEDI no inv**	**GEDI no gen**	**GEDI**
Moons	0.00±0.00	0.76±0.36	0.00±0.00	0.11±0.15	**0.98±0.00**	0.94±0.07
Circles	0.00±0.00	0.00±0.00	0.00±0.00	0.22±0.13	0.83±0.12	**1.00±0.01**
SVHN	0.00	0.21	-	-	0.21	**0.25**
CIFAR10	0.00	0.43	-	-	0.43	**0.45**
CIFAR100	0.00	0.65	-	-	0.86	**0.87**

3 Experiments

We perform experiments to evaluate the discriminative performance of GEDI and its competitors, namely an energy-based model JEM [8] and a self-supervised baseline based on SwAV [2]. The whole analysis is divided into two main experimental settings, the first one based on two synthetic datasets, including moons and circles, the second one based on real-world data, including SVHN, CIFAR-10 and CIFAR-100. We use existing code both as a basis to build our solution and also to run the experiments for the different baselines. In particular, we use the code from [7] for training energy-based models and the repository from [5] for all self-supervised baselines. Implementation details as well as additional experiments on generation, OOD detection and linear probe evaluation are reported in the Supplementary Material (Appendices D-G). In those additional experiments, we show that we can also tackle generative tasks using the trained neural network and achieve comparable performance to existing energy-based models. Moreover, thanks to this generative aspect, the trained network can better detect out-of-distribution data compared to SSL baselines.

Moons and Circles. In Table 2, we observe that JEM fails to solve the clustering task for both datasets. This is quite natural, as JEM is a purely generative approach, mainly designed to perform implicit density estimation (i.e. it is trained based solely on the $\mathcal{L}_{GEN}$ loss). SwAV can only solve the clustering task for the moons dataset, highlighting the fact that its objective function admits failure mode 3. Indeed, we observe in the circles dataset that half of the labels are permuted across the two manifolds (cf. Figure 3 in the Supplementary Material). In contrast, GEDI can recover the true clusters in both datasets, as it is guaranteed to avoid trivial solutions and learn more meaningful cluster assignments. We conduct an ablation study to understand the impact of the different loss terms in GEDI and empirically validate the theoretical results obtained in Section 4.3. We compare four different versions of GEDI, namely the full version (called simply GEDI), GEDI trained without $\mathcal{L}_{GEN}(\Theta)$ (called *no gen*), GEDI trained without $\mathcal{L}_{INV}(\Theta)$ (called *no inv*) and GEDI trained without $\mathcal{L}_{PRIOR}(\Theta)$ (called *no unif*). From the results in Table 2, we observe that: (i) GEDI *no unif* is subject to cluster collapse on both datasets. This is expected as failure mode 2 is not penalized during training due to the omission of $\mathcal{L}_{PRIOR}(\Theta)$; (ii) GEDI *no inv* is subject to the problem of permutation invariance to cluster assignments. Consequently, the obtained cluster labels are not informative and consistent with the underlying manifold structure of the data distribution. Again, this confirms the result of Theorem 1, as failure mode 3 could be avoided by the use of $\mathcal{L}_{INV}(\Theta)$; (iii) GEDI *no gen* achieves superior performance over other SSL baselines. While in theory the objective function for this approach admits representational collapse, in practice we never observed such issue. It might be the case that the learning dynamics of gradient-based optimisation are enough to avoid the convergence to this trivial solution. However, further analysis is required in order to verify this statement; finally (iv) GEDI is guaranteed to avoid the most important failure modes and therefore solve the discriminative task.

SVHN, CIFAR-10, CIFAR-100. From Table 2, we observe that GEDI is able to outperform all other competitors by a large margin. Additionally, we note a difference gap in clustering performance with increasing number of classes (cf. CIFAR-100). This might be explained by the fact that the number of possible label permutations increases with the number of classes and that our loss is more robust to the permutation invariance problem as from Theorem 1. Finally, GEDI *no gen* is comparable and often superior to SwAV, despite being simpler (i.e. avoiding the use of asymmetries and the running of iterative clustering). Please refer to Appendices F and G for further details.

References

[1] M. Caron, P. Bojanowski, A. Joulin, and M. Douze. Deep Clustering for Unsupervised Learning of Visual Features. In *ECCV*, 2018.

[2] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin. Unsupervised Learning of Visual Features by Contrasting Cluster Assignments. In *NeurIPS*, 2020.

[3] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. Emerging Properties in Self-Supervised Vision Transformers. In *ICCV*, 2021.

[4] M. Cuturi. Sinkhorn Distances: Lightspeed Computation of Optimal Transport. In *NeurIPS*, 2013.

[5] V. G. T. da Costa, E. Fini, M. Nabi, N Sebe, and E. Ricci. Solo-learn: A Library of Self-supervised Methods for Visual Representation Learning. *JMLR*, 2022.

[6] Y. Du and I. Mordatch. Implicit Generation and Generalization in Energy-Based Models. *arXiv*, 2019.

[7] D. Duvenaud, J. Kelly, K. Swersky, M. Hashemi, M. Norouzi, and W. Grathwohl. No MCMC for Me: Amortized Samplers for Fast and Stable Training of Energy-Based Models. In *ICLR*, 2021.

[8] W. Grathwohl, K.-C. Wang, J.-H. Jacobsen, D. Duvenaud, M. Norouzi, and K. Swersky. Your Classifier is Secretly an Energy Based Model and You Should Treat It Like One. In *ICLR*, 2020.

[9] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *NeurIPS*, 2017.

[10] B. Kim and J. C. Ye. Energy-Based Contrastive Learning of Visual Representations. In *NeurIPS*, 2022.

[11] E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, and B. Lakshminarayanan. Do deep generative models know what they don't know? In *ICLR*, 2019.

[12] E. Nijkamp, M. Hill, T. Han, S.-C. Zhu, and Y. N. Wu. On the Anatomy of MCMC-Based Maximum Likelihood Learning of Energy-Based Models. In *AAAI*, 2020.

[13] E. Nijkamp, M. Hill, S. C. Zhu, and Y. N. Wu. Learning non-convergent non-persistent short-run mcmc toward energy-based model. In *NeurIPS*, 2019.

[14] E. Sansone and R. Manhaeve. GEDI: GEnerative and DIscriminative Training for Self-Supervised Learning. *arXiv*, 2022.

[15] E. Sansone and R. Manhaeve. Learning Symbolic Representations Through Joint GEnerative and DIscriminative Training. In *ICLR Workshop NeSy-GeMs*, 2023.

[16] X. Wang, H. Fan, Y. Tian, D. Kihara, and X. Chen. On the Importance of Asymmetry for Siamese Representation Learning. In *CVPR*, 2022.

[17] J. Xie, Y. Lu, S. C. Zhu, and Y. N. Wu. A Theory of Generative Convnet. In *ICML*, 2016.

A Proof of Proposition 2.1

Proof. We recall Eq. (1) (we omit the dependence from Θ to avoid clutter), namely:

$$\begin{aligned}&\mathbb{E}_{p(x_{1:n})}\{\log p(x_{1:n})\}\\&= -H_p(x_{1:n}) + \mathbb{E}_{p(x_{1:n})\mathcal{T}(x'_{1:n}|x_{1:n})}\left\{\log\sum_{y_{1:n}} p(y_{1:n}|x_{1:n})\right\}\end{aligned}$$

and add the zero quantity $\log \sum_{y_{1:n}} p(y_{1:n})$ to the right-hand side of previous equation, thus obtaining the new equation

$$\begin{aligned}
&\mathbb{E}_{p(x_{1:n})}\{\log p(x_{1:n})\} \\
&= -H_p(x_{1:n}) + \mathbb{E}_{p(x_{1:n})\mathcal{T}(x'_{1:n}|x_{1:n})}\left\{\log \sum_{y_{1:n}} p(y_{1:n}|x_{1:n})\right\} \\
&\quad + \log \sum_{y_{1:n}} p(y_{1:n})
\end{aligned} \tag{7}$$

We can lower bound the previous equation by exploiting the fact that $\sum_z p(z) \geq \sum_z p(z)q(z)$ for any given auxiliary discrete distribution q, viz.:

$$\begin{aligned}
&\text{Eq. (7)} \geq \\
&\quad - H_p(x_{1:n}) \\
&\quad + \mathbb{E}_{p(x_{1:n})\mathcal{T}(x'_{1:n}|x_{1:n})}\left\{\log \sum_{y_{1:n}} q(y_{1:n}|x'_{1:n})p(y_{1:n}|x_{1:n})\right\} \\
&\quad + \log \sum_{y_{1:n}} p(y_{1:n})q(y_{1:n})
\end{aligned} \tag{8}$$

Now, by applying Jensen's inequality to the last two addends in Eq. (8) and by defining $q(y_{1:n}|x'_{1:n}) = p(y_{1:n}|x'_{1:n})$ and $q(y_{1:n}) = \frac{1}{n}\sum_{j=1}^{n} p(y_j|x_j)$, we obtain the following lower bound:

$$\begin{aligned}
&\text{Eq. (8)} \geq \\
&\quad - H_p(x_{1:n}) \\
&\quad + \mathbb{E}_{p(x_{1:n})\mathcal{T}(x'_{1:n}|x_{1:n})}\left\{\sum_{y_{1:n}} p(y_{1:n}|x'_{1:n}) \log p(y_{1:n}|x_{1:n})\right\} \\
&\quad + \sum_{y_{1:n}} p(y_{1:n}) \log\left(\frac{1}{n}\sum_{j=1}^{n} p(y_j|x_j)\right)
\end{aligned} \tag{9}$$

Additionally, by factorizing the distributions according to the probabilistic graphical model in Fig. 1, namely $p(y_{1:n}|x_{1:n}) = \prod_{i=1}^{n} p(y_i|x_i)$, $p(y_{1:n}|x'_{1:n}) = \prod_{i=1}^{n} p(y_i|x'_i)$ and $p(y_{1:n}) = \prod_{i=1}^{n} p(y_i)$, we achieve the following equality:

$$\begin{aligned}
&\text{Eq. (9)} = \\
&\quad - H_p(x_{1:n}) \\
&\quad + \sum_{i=1}^{n} \mathbb{E}_{p(x_i)\mathcal{T}(x'_i|x_i)}\left\{\sum_{y_i} p(y_i|x'_i) \log p(y_i|x_i)\right\} \\
&\quad + \sum_{i=1}^{n}\sum_{y_i} p(y_i) \log\left(\frac{1}{n}\sum_{j=1}^{n} p(y_j = y_i|x_j)\right)
\end{aligned} \tag{10}$$

And by rewriting the last two addends in Eq. (10) using the definition of cross-entropy, we obtain our final result.

Now, we can conclude the proof by looking at the maxima for $\mathcal{L}_{INV}$ and $\mathcal{L}_{PRIOR}$. Indeed, we observe that both terms compute a negative cross-entropy between two distributions. By leveraging the fact that $CE(p,q) = H_p + KL(p\|q)$ for arbitrary distributions p, q, we can easily see that the maximum of $\mathcal{L}_{INV}$ is attained when the term is 0 (corresponding to minimal entropy and minimal KL), whereas the maximum of $\mathcal{L}_{PRIOR}$ is attained when the term is equal to $-H_p(y_i)$ (corresponding to minimal KL). □

B Training Algorithm and Computational Requirements

Learning a GEDI model. We can train the GEDI model by jointly maximizing the objective in Eq. (5) with respect to the parameters Θ through gradient-based strategies. The overall gradient

Algorithm 1: GEDI Training.

```
Input: $x_{1:n}$, $x'_{1:n}$, Iters, SGLD and Adam optimizer hyperparameters;
Output: Trained model $\Theta$;
For iter $= 1, \dots,$ Iters;
    Generate samples from $p_\Theta$ using SGLD;
    Estimate $\Delta_1\Theta = \nabla_\Theta \mathcal{L}_{GEN}(\Theta)$ using Eq. (11);
    Compute $\Delta_2\Theta = \nabla_\Theta \mathcal{L}_{INV}(\Theta)$;
    Compute $\Delta_3\Theta = \nabla_\Theta \mathcal{L}_{PRIOR}(\Theta)$ ;
    $\Delta\Theta \leftarrow \sum_{i=1}^{3} \Delta_i\Theta$;
    $\Theta \leftarrow$ Adam maximizing using $\Delta\Theta$;
Return $\Theta$;
```

includes the summation of three terms, viz. $-\nabla_\Theta CE(p, p_\Theta)$, $\nabla_\Theta \mathcal{L}_{INV}(\Theta)$ and $\nabla_\Theta \mathcal{L}_{PRIOR}(\Theta)$. While the last two gradient terms can be computed easily by leveraging automatic differentiation, the first one must be computed by exploiting the following identities (obtained by simply substituting Eq. (6) into the definition of cross-entropy and expanding $\nabla_\Theta \Gamma(\Theta)$):

$$\begin{aligned}
-\nabla_\Theta CE(p, p_\Theta) &= \sum_{i=1}^{n} \mathbb{E}_{p(x_i)} \left\{ \nabla_\Theta \log \sum_{y=1}^{c} e^{f_y(enc(x_i))/\tau} \right\} \\
&\quad - n\nabla_\Theta \log \Gamma(\Theta) \\
&= \sum_{i=1}^{n} \mathbb{E}_{p(x_i)} \left\{ \nabla_\Theta \log \sum_{y=1}^{c} e^{f_y(enc(x_i))/\tau} \right\} \\
&\quad n\mathbb{E}_{p_\Theta(x)} \left\{ \nabla_\Theta \log \sum_{y=1}^{c} e^{f_y(enc(x))/\tau} \right\}
\end{aligned} \tag{11}$$

Importantly, the first and the second expectations in Eq. (11) are estimated using the training and the generated data, respectively. To generate data from p_Θ, we use a sampler based on Stochastic Gradient Langevin Dynamics (SGLD), thus following recent best practices to train energy-based models [17, 13, 6, 12]. The whole learning procedure is summarized in Algorithm 1.

Computational requirements. When comparing our GEDI instantiation with traditional SSL training, more specifically to SwAV, we observe two main differences in terms of computation. Firstly, our learning algorithm does not require to run the Sinkhorn-Knopp algorithm, thus saving computation. Secondly, our GEDI instantiation requires additional forward and backward passes to draw samples from the energy-based model p_Θ. However, the number of additional passes through the discriminative model can be limited by the number of SGLD iterations.

C Proof of Theorem 1

Proof. The overall strategy to prove the statements relies on the evaluation of the loss terms over the three failure modes and on checking whether these attain their corresponding maxima.

Let's start by proving statement a and recalling that $\mathcal{L}_{GEN}(\Theta, \mathcal{D}) = -CE(p, p_\Theta)$. Firstly, we test for failure mode 1 (i.e. representational collapse). We observe that for all $x \in \mathbb{R}^d$

$$p_\Theta(x) = \frac{\sum_{y=1}^{c} e^{f_y(k)/\tau}}{\Gamma(\Theta)}$$

thus $p_\Theta(x)$ assigns constant mass everywhere. Clearly, p_Θ is different from p. Therefore, $-CE(p, p_\Theta) < -CE(p, p)$ and failure mode 1 is not admissible. Secondly, we test for failure mode 2 (i.e. cluster collapse). We can equivalently rewrite the definition of cluster collapse by stating that there exists $j \in \{1, \dots, c\}$, such that for all $x \in \mathbb{R}^d$ and for all $y \neq j$, $f_j(enc(x)) - f_y(enc(x)) \to \infty$. Additionally, we observe that

$$p_\Theta(x) \underbrace{=}_{\xi_x \doteq enc(x)} \frac{\sum_{y=1}^{c} e^{f_y(\xi_x)/\tau}}{\int \sum_{y=1}^{c} e^{f_y(\xi_x)/\tau} dx}$$

$$
\begin{aligned}
&= \frac{e^{f_j(\xi_x)/\tau}\left[1+\sum_{y\neq j}\cancelto{0}{e^{(f_y(\xi_x)-f_j(\xi_x))/\tau}}\right]}{\int e^{f_j(\xi_x)/\tau}\left[1+\sum_{y\neq j}\cancelto{0}{e^{(f_y(\xi_x)-f_j(\xi_x))/\tau}}\right]dx} \\
&= \frac{e^{f_j(\xi_x)/\tau}}{\int e^{f_j(\xi_x)/\tau}dx} \qquad (12)
\end{aligned}
$$

where we have used the failure mode condition to obtain the last equality. Now, note that Eq. (12) defines a standard energy-based model. Consequently, given enough capacity for the predictive model, it is trivial to verify that there exists Θ such that the condition about failure mode is met and p_Θ is equal to p. Cluster collapse is therefore an admissible solution. Thirdly, we test for permutation invariance for the cluster assignments. Indeed, we have that

$$
\begin{aligned}
\mathcal{L}_{GEN}(\Theta,\mathcal{D}) &= \sum_{i=1}^{n}\mathbb{E}_{p(x_i)}\left\{\log p_\Theta(x_i)\right\} \\
&= \sum_{i=1}^{n}\mathbb{E}_{p(x_i)}\left\{\log\frac{\sum_{y=1}^{c}e^{t_i(y)/\tau}}{\int\sum_{y=1}^{c}e^{t_i(y)/\tau}dx}\right\} \\
&= \sum_{i=1}^{n}\mathbb{E}_{p(x_{1:n})}\left\{\log\frac{\sum_{y=1}^{c}e^{t_i(y)/\tau}}{\int\sum_{y=1}^{c}e^{t_i(y)/\tau}dx}\right\} \qquad (13)
\end{aligned}
$$

where $t_i(y) = f_y(enc(x_i))$. Similarly, we have that

$$
\begin{aligned}
&\mathcal{L}_{GEN}(\Theta,\mathcal{D}^\pi) \\
&\quad\underbrace{=}_{\text{from Eq. (13)}}\sum_{i=1}^{n}\mathbb{E}_{p(x_{1:n})}\left\{\log\frac{\sum_{y=1}^{c}e^{t_{\pi(i)}(y)/\tau}}{\int\sum_{y=1}^{c}e^{t_{\pi(i)}(y)/\tau}dx}\right\} \\
&\quad= \sum_{i=1}^{n}\mathbb{E}_{p(x_{\pi(i)})}\left\{\log\frac{\sum_{y=1}^{c}e^{t_{\pi(i)}(y)/\tau}}{\int\sum_{y=1}^{c}e^{t_{\pi(i)}(y)/\tau}dx}\right\} \\
&\quad\underbrace{=}_{l\doteq\pi(i)}\sum_{l=1}^{n}\mathbb{E}_{p(x_l)}\left\{\log\frac{\sum_{y=1}^{c}e^{t_l(y)/\tau}}{\int\sum_{y=1}^{c}e^{t_l(y)/\tau}dx}\right\} \\
&\quad= \mathcal{L}_{GEN}(\Theta,\mathcal{D})
\end{aligned}
$$

Hence, failure mode 3 is an admissible solution.

Let's continue by proving statement b and recalling that

$$
\mathcal{L}_{INV}(\Theta,\mathcal{D}) = -\sum_{i=1}^{n}\mathbb{E}_{p(x_i)\mathcal{T}(x_i'|x_i)}\left\{CE(p(y_i|x_i';\Theta),p(y_i|x_i;\Theta))\right\} \qquad (14)
$$

Firstly, we test for representational collapse. In this case, we have that for all $i\in\{1,\dots,n\}$

$$
p(y_i|x_i;\Theta) = p(y_i|x_i';\Theta) = \text{Softmax}(f(k)/\tau)
$$

Based on this result, we observe that the cross-entropy terms in Eq. (14) can be made 0 by proper choice of k. Therefore, representational collapse is an admissible solution. Secondly, we test for cluster collapse. Here, it is easy to see that the cross-entropy terms in Eq. (14) are all 0. Therefore, also cluster collapse is admissible. Thirdly, we test for permutation invariance for the cluster assignments. On one hand, we have that the cross-entropy terms for $\mathcal{L}_{INV}(\Theta,\mathcal{D}))$ in Eq. (14) can be rewritten in the following way:

$$
\begin{aligned}
&CE(p(y_i|x_i';\Theta),p(y_i|x_i;\Theta)) \\
&\quad= CE\left(\frac{e^{t_i'(y_i)/\tau}}{\sum_{y=1}^{c}e^{t_i'(y)/\tau}},\frac{e^{t_i(y_i)/\tau}}{\sum_{y=1}^{c}e^{t_i(y)/\tau}}\right) \qquad (15)
\end{aligned}
$$

and the optimal solution is achieved only when $t'_i = t_i$ for all $i \in \{1, \dots, n\}$. On the other hand, the cross-entropy terms for $\mathcal{L}_{INV}(\Theta, \mathcal{D}^\pi)$ are given by the following equality:

$$\begin{aligned} & CE(p(y_i|x'_i;\Theta), p(y_i|x_i;\Theta)) \\ &= CE\left(\frac{e^{t'_i(y_i)/\tau}}{\sum_{y=1}^c e^{t'_i(y)/\tau}}, \frac{e^{t_{\pi(i)}(y_i)/\tau}}{\sum_{y=1}^c e^{t_{\pi(i)}(y)/\tau}}\right) \end{aligned} \tag{16}$$

However, the optimal solution cannot be achieved in general as $t'_i \neq t_{\pi(i)}$ for some $i \in \{1, \dots, n\}$.[3] Therefore, $\mathcal{L}_{INV}$ is not permutation invariant to cluster assignments.

Let's conclude by proving statement c and recalling that

$$\mathcal{L}_{PRIOR}(\Theta, \mathcal{D}) = -\sum_{i=1}^n CE\left(p(y_i), \frac{1}{n}\sum_{l=1}^n p(y_l = y_i|x_l;\Theta)\right) \tag{17}$$

Firstly, we test for representational collapse. One can easily observe that if $enc(x) = k$ for all $x \in \mathbb{R}^d$, $p(y|x;\Theta)$ becomes uniform, namely $p(y|x;\Theta) = 1/c$ for all $y \in \{1, \dots, c\}$. Consequently, $\frac{1}{n}\sum_{l=1}^n p(y_l = y_i|x_l;\Theta) = 1/c$ for all $i \in \{1, \dots, n\}$. Now, since $p(y_i) = 1/c$ for all $i \in \{1, \dots, n\}$, the cross-entropy terms in Eq. (17) reach their maximum value $-H_p(y_i)$ for all $i \in \{1, \dots, n\}$. Therefore, representational collapse attains the global maximum of $\mathcal{L}_{PRIOR}$ and is an admissible solution. Secondly, we test for cluster collapse. By using the definition of cluster collapse, we observe that

$$\frac{1}{n}\sum_{l=1}^n p(y_l = y_i|x_l;\Theta) = \begin{cases} 0 & y_i = j \\ 1 & y_i \neq j \end{cases} \tag{18}$$

Therefore, the resulting distribution is non-uniform, differently from $p(y_i)$. The cross-entropy terms in Eq. (17) are not optimized and cluster collapse is not admissible. Thirdly, we test for permutation invariance of cluster assignments. We observe that

$$\begin{aligned} \frac{1}{n}\sum_{l=1}^n p(y_l = y_i|x_l;\Theta) &= \frac{1}{n}\sum_{l=1}^n \frac{e^{t_l(y_i)/\tau}}{\sum_{y=1}^c e^{t_l(y)/\tau}} \\ &= \frac{1}{n}\sum_{l=1}^n \frac{e^{t_{\pi(l)}(y_i)/\tau}}{\sum_{y=1}^c e^{t_{\pi(l)}(y)/\tau}} \end{aligned} \tag{19}$$

which is permutation invariant to cluster assignments. Consequently, also $\mathcal{L}_{PRIOR}(\Theta, \mathcal{D}) = \mathcal{L}_{PRIOR}(\Theta, \mathcal{D}^\pi)$. This concludes the proof. □

D Hyperparameters for Synthetic Data

For the backbone enc, we use a MLP with two hidden layers and 100 neurons per layer, an output layer with 2 neurons and ReLU activation functions. For the projection head $proj$ (f for GEDI and its variants), we use a MLP with one hidden layer and 4 neurons and an output layer with 2 neurons (batch normalization is used in all layers for Barlow and SwAV as required by their original formulation). All methods use a batch size of 400. Baseline JEM (following the original paper):

- Number of iterations $20K$
- Learning rate $1e-3$
- Optimizer Adam $\beta_1 = 0.9$, $\beta_2 = 0.999$
- SGLD steps 10
- Buffer size 10000
- Reinitialization frequency 0.05
- SGLD step-size $\frac{0.01^2}{2}$
- SGLD noise 0.01

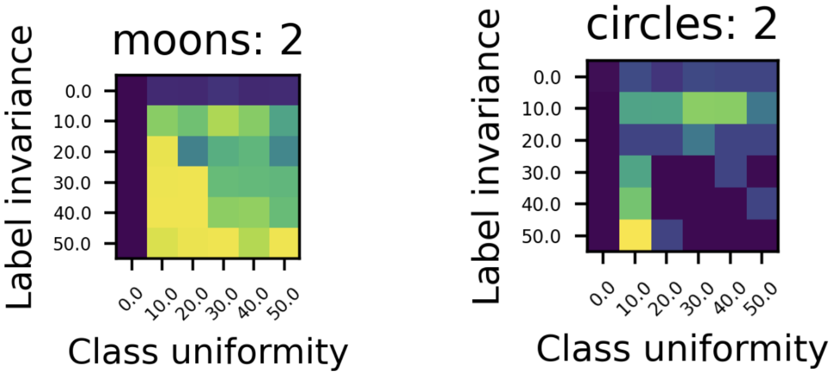
And for self-supervised learning methods, please refer to Table 3.

We also provide an analysis of sensitivity to hyperparameters for GEDI. Please refer to Figure 2.

[3] Indeed, note that $t'_i = t_{\pi(i)}$ for all i occurs only when we are in one of the first two failure modes.

Table 3: Hyperparameters used in the synthetic experiments.

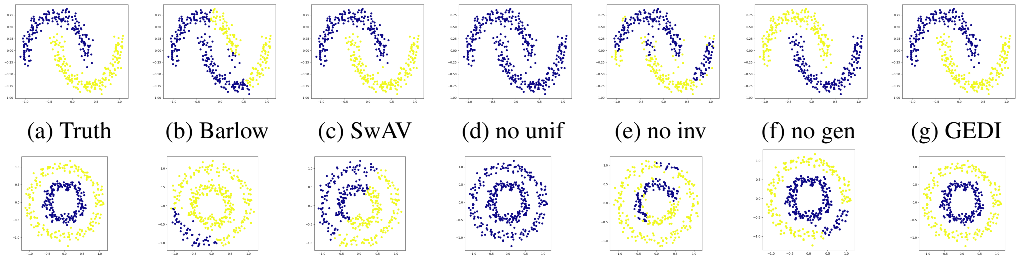
Methods	Barlow	SwAV	GEDI (no gen)	GEDI
Iters			$20k$	
Learning rate			$1e-3$	
Optimizer			Adam $\beta_1 = 0.9, \beta_2 = 0.999$	
Data augmentation noise σ			0.03	
SGLD steps T	-	-	1	1
Buffer size $\lvert B\rvert$	-	-	10000	10000
Reinitialization frequenc	-	-	0.05	0.05
SGLD step size	-	-	$\frac{0.01^2}{2}$ $\frac{0.01^2}{2}$	$\frac{0.01^2}{2}$
SGLD noise	-	-	0.01	0.01
Weight for $\mathcal{L}_{GEN}$	-	-	1	1
Weight for $\mathcal{L}_{INV}$	-	-	50	50
Weight for $\mathcal{L}_{PRIOR}$	-	-	10	10



(a) Moons

(b) Circles

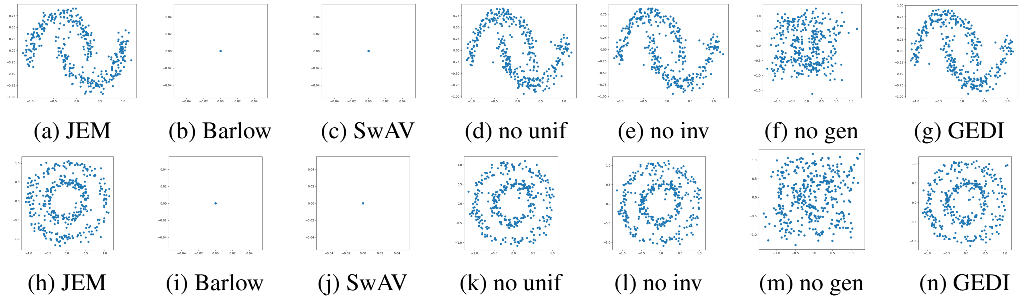
Figure 2: Sensitivity analysis on the discriminative performance of GEDI for different loss weights (in the range $\{0, 10, 20, 30, 40, 50\}$). Performance are averaged over 5 different random seeds. Yellow means perfect NMI.



(a) Truth (b) Barlow (c) SwAV (d) no unif (e) no inv (f) no gen (g) GEDI

(h) Truth (i) Barlow (j) SwAV (k) no unif (l) no inv (m) no gen (n) GEDI

Figure 3: Qualitative visualization of the clustering performance for the different strategies on moons (a-g) and on circles (h-n) datasets. Colors identify different cluster predictions. Only GEDI and GEDI (no gen) are able to perform well on both datasets.


(a) JEM (b) Barlow (c) SwAV (d) no unif (e) no inv (f) no gen (g) GEDI

(h) JEM (i) Barlow (j) SwAV (k) no unif (l) no inv (m) no gen (n) GEDI

Figure 4: Qualitative visualization of the generative performance for the different strategies on moons (a-g) and on circles (h-n) datasets. Colors identify different cluster predictions. All GEDI approaches (except for no gen perform comparably well to the generative solution JEM).

Table 4: Resnet architecture. Conv2D(A,B,C) applies a 2d convolution to input with B channels and produces an output with C channels using stride (1, 1), padding (1, 1) and kernel size (A, A).

Name	**Layer**	**Res. Layer**

Block 1	Conv2D(3,3,F)
	LeakyRELU(0.2)
	Conv2D(3,F,F)
	AvgPool2D(2)
		AvgPool2D(2)
		Conv2D(1,3,F) no padding

	Sum	

Block 2	LeakyRELU(0.2)
	Conv2D(3,F,F)
	LeakyRELU(0.2)
	Conv2D(3,F,F)
	AvgPool2D(2)	

Block 3	LeakyRELU(0.2)
	Conv2D(3,F,F)
	LeakyRELU(0.2)
	Conv2D(3,F,F)	

Block 4	LeakyRELU(0.2)
	Conv2D(3,F,F)
	LeakyRELU(0.2)
	Conv2D(3,F,F)
	AvgPool2D(all)	

E Additional Experiments for Toy Data

Please, refer to Figure 3 for the discriminative performance and Figure 4 for the generative ones.

F Hyperparameters for SVHN, CIFAR10, CIFAR100

For the backbone enc, we use a ResNet with 8 layers as in [7], where its architecture is shown in Table 4. For the projection head $proj$ (f for GEDI and its variants), we use a MLP with one hidden layer and $2 * F$ neurons and an output layer with F neurons (batch normalization is used in all layers for Barlow and SwAV as required by their original formulation + final L_2 normalization). $F = 128$ for SVHN, CIFAR-10 (1 million parameters) and $F = 256$ for CIFAR-100 (4.1 million parameters). For JEM, we use the same settings of [7]. All methods use a batch size of 64. Baseline JEM (following the original paper):

- Number of epochs 20, 200, 200 for SVHN, CIFAR-10, CIFAR-100, respectively.

- Learning rate $1e-4$
- Optimizer Adam
- SGLD steps 20
- Buffer size 10000
- Reinitialization frequency 0.05
- SGLD step-size 1
- SGLD noise 0.01
- Data augmentation (Gaussian noise) 0.03

And for self-supervised learning methods, please refer to Table 5.

Table 5: Hyperparameters (in terms of sampling, optimizer, objective and data augmentation) used in all experiments.

Class	**Name param.**	**SVHN**	**CIFAR-10**	**CIFAR-100**
Data augment.	Color jitter prob.	0.1	0.1	0.1
	Gray scale prob.	0.1	0.1	0.1
	Random crop	Yes	Yes	Yes
	Additive Gauss. noise (std)	0.03	0.03	0.03
	Random horizontal flip	No	Yes	Yes
SGLD	SGLD iters	20	20	20
	Buffer size	10k	10k	10k
	Reinit. frequency	0.05	0.05	0.05
	SGLD step-size	1	1	1
	SGLD noise	0.01	0.01	0.01
Optimizer	Batch size	64	64	64
	Epochs	20	200	200
	Adam β_1	0.9	0.9	0.9
	Adam β_2	0.999	0.999	0.999
	Learning rate	$1e-4$	$1e-4$	$1e-4$
Weights for losses	$\mathcal{L}_{GEN}$	1	1	1
	$\mathcal{L}_{INV}$	50	50	50
	$\mathcal{L}_{PRIOR}$	25	25	50
	$\mathcal{L}_{NeSY}$	-	-	-

G Experiments on SVHN, CIFAR-10, CIFAR-100

We consider three well-known computer vision benchmarks, namely SVHN, CIFAR-10 and CIFAR-100. We use a simple 8-layer Resnet network for the backbone encoder for both SVHN and CIFAR-10 (around 1M parameters) and increase the hidden layer size for CIFAR-100 (around 4.1M parameters) as from [7]. We use a MLP with a single hidden layer for $proj$ (the number of hidden neurons is double the number of inputs), we choose $h = 256$ for CIFAR-100 and $h = 128$ for all other cases. Additionally, we use data augmentation strategies commonly used in the SSL literature, including color jitter, and gray scale conversion to name a few. We train JEM, Barlow, SwAV, GEDI no gen and GEDI using Adam optimizer with learning rate $1e-4$ and batch size 64 for 20, 200 and 200 epochs for each respective dataset (SVHN, CIFAR-10 AND CIFAR-100). Further details about the hyperparameters are available in the Supplementary Material (Section I). Similarly to the toy experiments, we evaluate the clustering performance by using the Normalized Mutual Information (NMI) score. Additionally, we evaluate the generative performance qualitatively using the Frechet Inception Distance [9] as well as the OOD detection capabilities following the methodology in [8].

Table 6: Generative and discriminative performance on test set (SVHN, CIFAR-10, CIFAR-100). Normalized mutual information (NMI) and Frechet Inception Distance (FID) are used as evaluation metrics for the discriminative and generative tasks, respectively. Higher values of NMI and lower values of FID indicate better performance.

Task	Method	SVHN	CIFAR-10	CIFAR-100
Discriminative (NMI)	JEM	0.00	0.00	0.00
	Barlow	0.23	0.17	0.58
	SwAV	0.21	0.43	0.65
	GEDI (no gen)	0.21	0.44	0.86
	GEDI	**0.25**	**0.45**	**0.87**
Generative (FID)	JEM	**201**	263	226
	Barlow	353	392	352
	SwAV	584	415	416
	GEDI (no gen)	454	424	447
	GEDI	218	**197**	**222**

Table 7: OOD detection in terms of AUROC on test set (CIFAR-10, CIFAR-100). Training is performed on SVHN.

Dataset	JEM	Barlow	SwAV	GEDI no gen	GEDI
CIFAR-10	0.73	0.17	0.26	01	**0.80**
CIFAR-100	0.72	0.24	0.32	0.15	**0.80**

From Table 6, we observe that GEDI is able to outperform all other competitors by a large margin, thanks to the properties of both generative and self-supervised models. We observe that the difference gap in clustering performance increases with a larger number of classes (cf. CIFAR-100). This might be explained by the fact that the number of possible label permutations can increase with the number of classes and that our loss is more robust to the permutation invariance problem as from Theorem 1. We observe also that GEDI no gen is comparable and often superior to SwAV, despite being simpler (i.e. avoiding the use of asymmetries and the running of iterative clustering). In terms of generation performance, GEDI is the only approach that compares favorably with JEM. We provide a qualitative set of samples generated by the different discriminative models in Figure 5.

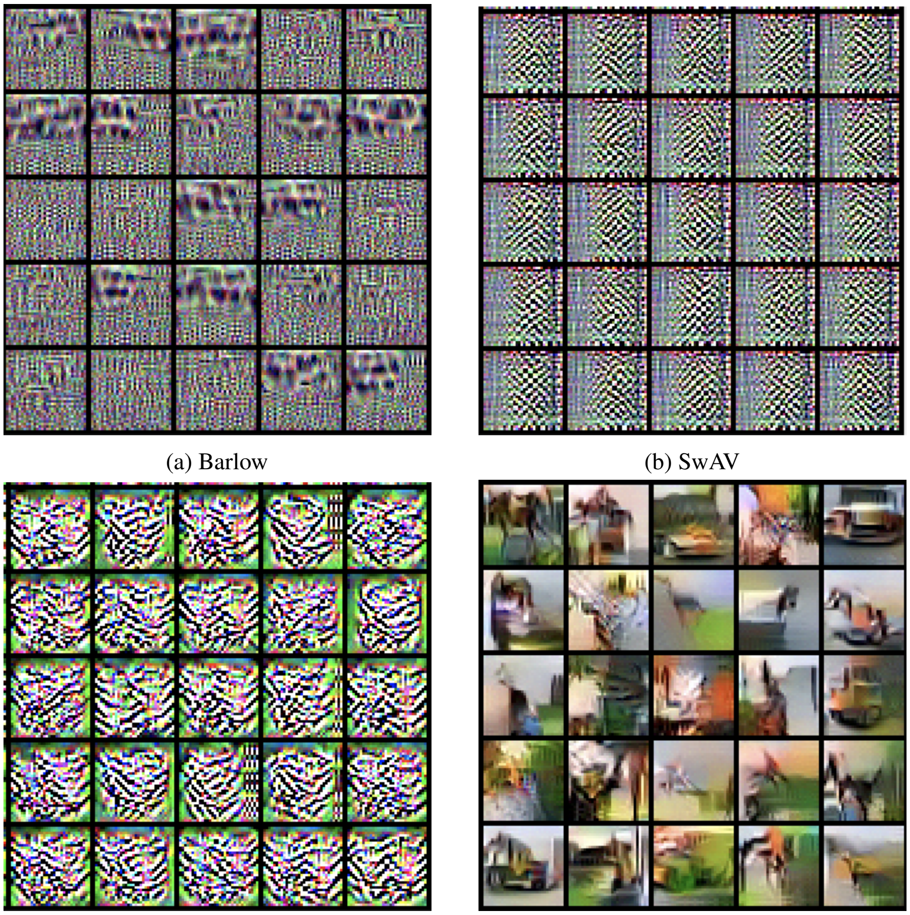
Last but not least, we investigate the OOD detection capabilities of the different methods. Table 9 provides a quantitative summary of the performance for a subset of experiments (the complete set is available in Section J). We observe that GEDI is more robust compared to other discriminative baselines, thanks to its generative nature.

Overall, these experiments provide real-world evidence on the benefits of the proposed unification and theoretical results.

We conduct a linear probe evaluation of the representations learnt by the different models Table 8. These experiments provide insights on the capabilities of learning representations producing linearly separable classes. From Table 8, we observe a large difference in results between Barlow and SwAV. Our approach provides interpolating results between the two approaches.

We also provide additional qualitative analyisis on the generation performance on SVHN and CIFAR-100. Please, refer to Figure 6 and Figure 7.

Finally, we evaluate the performance in terms of OOD detection, by following the same methodology used in [8]. We use the OOD score criterion proposed in [8], namely $s(x) = -\|\frac{\partial \log p_\Psi(x)}{\partial x}\|_2$. From Table 7, we observe that GEDI achieves almost optimal performance. While these results are exciting, it is important to mention that they are not generally valid. Indeed, when training on CIFAR-10 and performing OOD evaluation on the other datasets, we observe that all approaches achieve similar performance both on CIFAR-100 and SVHN, suggesting that all datasets are considered in-distribution, see Table 9. A similar observation is obtained when training on CIFAR-100 and evaluating on CIFAR-10 and SVHN, see Table 10. Importantly, this is a phenomenon which has been



(a) Barlow

(b) SwAV

(c) GEDI no gen

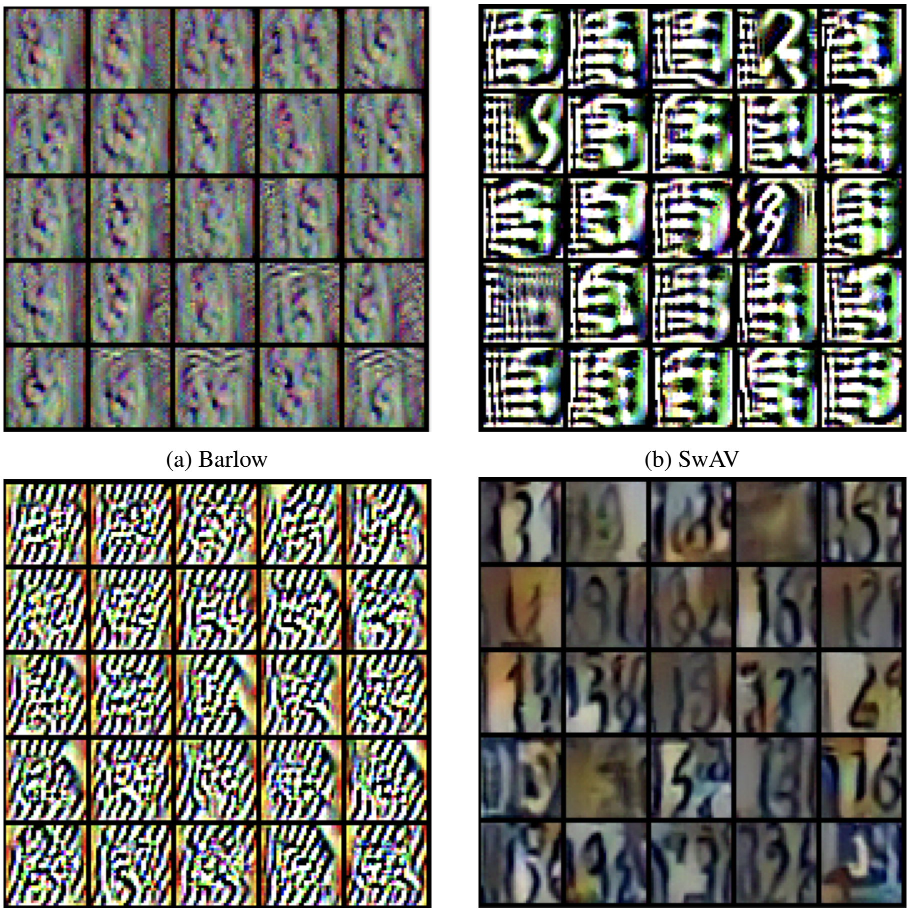
(d) GEDI

Figure 5: Qualitative visualization of the generative performance for the different discriminative strategies on CIFAR-10. Results are obtained by running Stochastic Langevin Dynamics for 500 iterations.

only recently observed by the scientific community on generative models. Tackling this problem is currently out of the scope of this work. For further discussion about the issue, we point the reader to the works in [11].

Table 8: Supervised linear evaluation in terms of accuracy on test set (SVHN, CIFAR-10, CIFAR-100). The linear classifier is trained for 100 epochs using SGD with momentum, learning rate $1e-3$ and batch size 100.

Dataset	**JEM**	**Barlow**	**SwAV**	**GEDI no gen**	**GEDI**
SVHN	0.20	0.74	0.45	0.35	0.55
CIFAR-10	0.23	0.65	0.46	0.54	0.53
CIFAR-100	0.03	0.27	0.13	0.14	0.12



(a) Barlow

(b) SwAV

(c) GEDI no gen

(d) GEDI

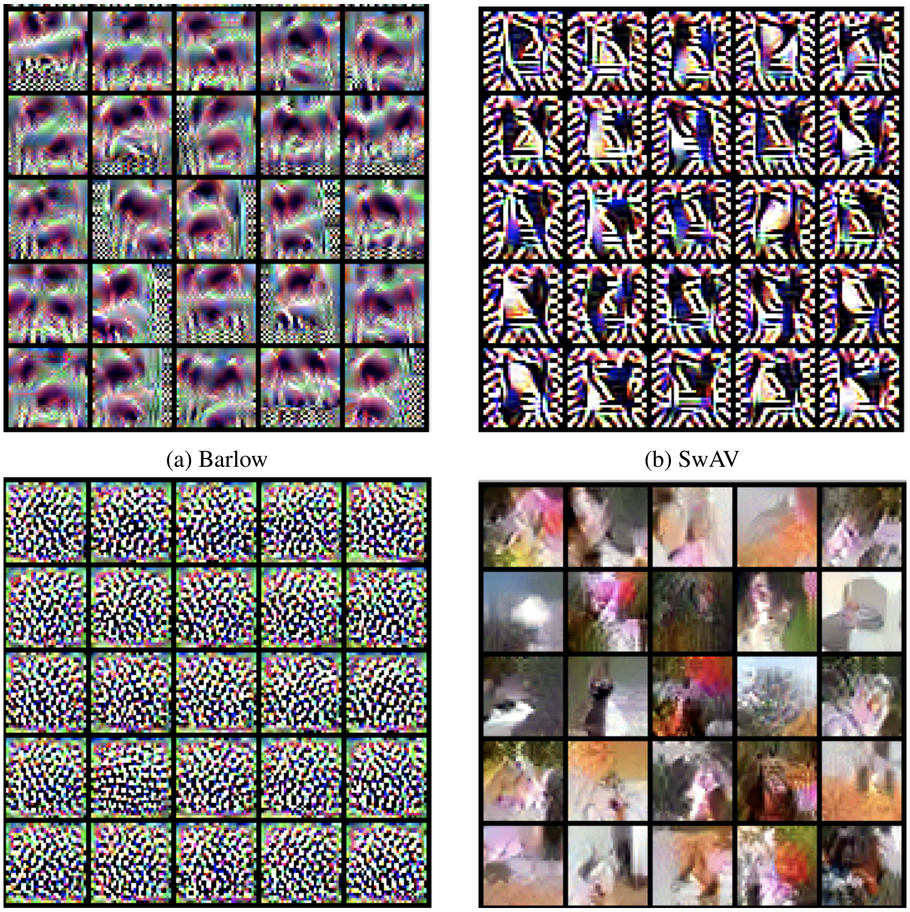
Figure 6: Qualitative visualization of the generative performance for the different discriminative strategies on SVHN. Results are obtained by running Stochastic Langevin Dynamics for 500 iterations.

Table 9: OOD detection in terms of AUROC on test set (SVHN, CIFAR-100). Training is performed on CIFAR-10.

Dataset	**JEM**	**Barlow**	**SwAV**	**GEDI no gen**	**GEDI**
SVHN	0.44	0.32	**0.62**	0.11	0.57
CIFAR-100	0.53	0.56	0.51	0.51	**0.61**

Table 10: OOD detection in terms of AUROC on test set (SVHN, CIFAR-10). Training is performed on CIFAR-100.

Dataset	**JEM**	**Barlow**	**SwAV**	**GEDI no gen**	**GEDI**
SVHN	0.44	0.45	0.3	**0.55**	0.53
CIFAR-10	**0.49**	0.43	0.47	0.46	0.48



(a) Barlow

(b) SwAV

(c) GEDI no gen

(d) GEDI

Figure 7: Qualitative visualization of the generative performance for the different discriminative strategies on CIFAR-100. Results are obtained by running Stochastic Langevin Dynamics for 500 iterations.