Improving Pre-trained Self-Supervised Embeddings Through Effective Entropy Maximization

Deep Chakraborty
UMass Amherst

Yann LeCun
New York University
Meta - FAIR

Tim G. J. Rudner
New York University

Erik Learned-Miller
UMass Amherst

Abstract

A number of different architectures and loss functions have been applied to the problem of self-supervised learning (SSL), with the goal of developing embeddings that provide the best possible pre-training for as-yet-unknown, lightly supervised downstream tasks. One of these SSL criteria is to maximize the entropy of a set of embeddings in some compact space. But the goal of maximizing the embedding entropy often depends—whether explicitly or implicitly—upon high dimensional entropy estimates, which typically perform poorly in more than a few dimensions. In this paper, we motivate an effective entropy maximization criterion (E2MC), defined in terms of easy-to-estimate, low-dimensional constraints. We demonstrate that using it to continue training an already-trained SSL model for only a handful of epochs leads to a consistent and, in some cases, significant improvement in downstream performance. We perform careful ablation studies to show that the improved performance is due to the proposed add-on criterion. We also show that continued pre-training with alternative criteria does not lead to notable improvements, and in some cases, even degrades performance.

1 INTRODUCTION

Self-supervised learning (SSL) methods are widely employed for pre-training features on unlabeled data and are highly effective for subsequent fine-tuning on a wide variety of downstream tasks (Chen et al., 2020; Grill et al., 2020; Caron et al., 2020; Bardes et al., 2021).

In this paper, we ask whether it is possible to formulate a well-motivated, general-purpose criterion that allows *further improving* already-trained, highly-optimized SSL embeddings with only a handful of epochs of *continued pre-training*.

Like several previous works (Bojanowski and Joulin, 2017; Wang and Isola, 2020; Liu et al., 2022; Ozsoy et al., 2022), we start with the principle of maximizing the entropy of embeddings. One well-known motivation for this is that for a *discrete* embedding space, maximizing the entropy of a deterministic mapping preserves as much information as possible about the inputs. That is, such a maximum-entropy embedding maximizes the mutual information between the embedding and the input distribution (see, for example, Hjelm et al., 2018). Similar results hold for continuous embeddings under appropriate noise models (see, for example, discussion of the Gaussian channel in Cover and Thomas, 1991).

By maximizing the amount of information retained, one hopes to prepare as well as possible for future, as-yet-unknown, discrimination tasks. Our contribution is thus not the maximization of embedding entropy, but rather how we go about it.

A fundamental problem with entropy maximization. For any input distribution, a fixed neural network induces a distribution $p(z)$ on an embedding space. Since any neural network embedding is trained with a finite sample, we have no direct access to $p(z)$ and must attempt to maximize its entropy from a sample. Unfortunately, the amount of data required to get useful entropy estimates grows exponentially with the number of dimensions (McAllester and Stratos, 2020). Practical estimators of joint entropy start to break down after just a handful (≤ 10) of intrinsic dimensions (Miller, 2003). Thus, to claim that we are actually maximizing entropy in hundreds or thousands of dimensions is implausible. Instead, we focus on enforcing necessary, but not sufficient, conditions for maximum entropy.

Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258. Copyright 2025 by the author(s).

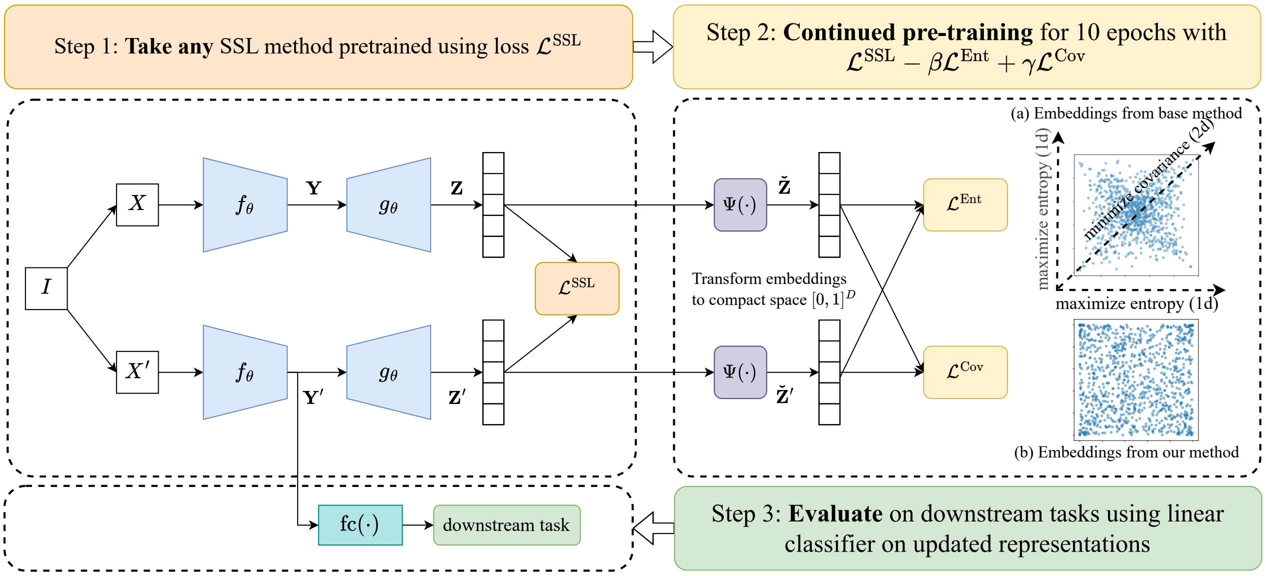

Step 1: **Take any** SSL method pretrained using loss $\mathcal{L}^{\text{SSL}}$
Step 2: **Continued pre-training** for 10 epochs with $\mathcal{L}^{\text{SSL}} - \beta\mathcal{L}^{\text{Ent}} + \gamma\mathcal{L}^{\text{Cov}}$
I
X
X'
f_θ
g_θ
$\mathbf{Y}$
$\mathbf{Y}'$
$\mathbf{Z}$
$\mathbf{Z}'$
$\mathcal{L}^{\text{SSL}}$
$\Psi(\cdot)$
$\tilde{\mathbf{Z}}$
$\tilde{\mathbf{Z}}'$
Transform embeddings to compact space $[0,1]^D$
$\mathcal{L}^{\text{Ent}}$
$\mathcal{L}^{\text{Cov}}$
(a) Embeddings from base method
maximize entropy (1d)
minimize covariance (2d)
maximize entropy (1d)
(b) Embeddings from our method
fc$(\cdot)$
downstream task
Step 3: **Evaluate** on downstream tasks using linear classifier on updated representations

Figure 1: An overview of our continued pre-training with E2MC approach with three main stages: SSL model selection, training using augmented criterion, and evaluating updated representations on downstream tasks.

In particular, we choose conditions to enforce for which we have sufficient data: low-dimensional statistics.

These statistics are:

1. The one-dimensional entropy of each marginal component of our embeddings.
2. The correlation of all pairs of marginals.

They have the following key properties:

1. They are *necessary* prerequisites for a maximum entropy joint distribution.
2. We have plenty of data to estimate them, due to their low dimensionality.

At this point, we restate the fact that these statistics alone are *not sufficient* to enforce maximum entropy of a joint distribution. It is well known that joint distributions that are decorrelated and have maximum entropy marginals can have higher order (3rd order and higher) dependencies, dramatically reducing their joint entropy.

Surprisingly, we find that—without explicitly enforcing higher-dimensional constraints in our criterion—higher-order marginals of our embeddings naturally tend towards uniformity, resulting in practically useful embeddings. We demonstrate how this criterion can be added-on to *any* pre-existing already-trained SSL model, which when further trained (*continued pre-training*) for a handful of epochs (as few as ten), leads to consistent improvements in downstream classification tasks. In a resource-constrained compute environment, where a necessary downstream application is label deficient, gains from our proposed modifications are particularly higher, and can be used to leverage the full potential of powerful off-the-shelf SSL models by rapidly adapting their embeddings. Our code is available at `https://github.com/deepc94/e2mc`.

The main contributions of this paper are as follows:

1. We consider the problem of further improving already-trained, highly-optimized SSL embeddings using only limited computational resources.
2. We motivate an *effective entropy maximization criterion* (E2MC) grounded in information-theoretic principles and show that it can be used as an add-on criterion for popular SSL methods.
3. We perform an empirical evaluation and find that with only a handful of epochs of continued pre-training under the proposed criterion, we achieve consistent and, in some cases, significant improvements in downstream-task performance across a selection of computer vision tasks.

2 BACKGROUND

2.1 Joint-embedding self-supervised learning

We refer to SSL methods that bring two 'similar' input views (say, translated versions of the same image) closer together in the representation space while spreading apart different images, either explicitly (Chen et al., 2020) or implicitly (Grill et al., 2020), as *joint embedding methods*. These methods typically use Siamese style neural networks (Bromley et al., 1993) f_θ (encoder) to compute representation vectors $Y = f_\theta(X)$ and $Y' = f_\theta(X')$, where X, X' are the two input image views. These representation vectors are then further transformed by an MLP g_θ (projector) to produce the final embeddings $Z_\theta = g_\theta(Y)$ and $Z'_\theta = g_\theta(Y')$. Z, Z' are then optionally normalized (e.g., on the surface of a hypersphere) and used to compute one or more SSL loss functions $\mathcal{L}^{\mathrm{SSL}}(\theta)$ (See 'Step 1' in Figure 1). Once training is complete, the projector is discarded, and the representation vector Y is used for downstream tasks.

Most methods employ regularization of some sort on the Z embeddings in order to prevent trivial solutions or enforce desirable properties, or both. We have a similar goal wherein we take any such pre-trained SSL model and update its embeddings Z by pre-training it for few additional epochs using our E2MC approach (c.f. Step 2 in Figure 1). Below, we briefly review some popular SSL methods, which we later improve using our proposed criterion.

2.1.1 Variance-Invariance-Covariance Regularization (VICReg)

VICReg (Bardes et al., 2021) is a feature decorrelation-based SSL method composed of the following:

(a) *Invariance*: minimizes the euclidean distance between the embeddings of the original images and their augmented views Z, Z', to learn features that remain consistent through input transformations.

(b) *Regularization*: consists of a *variance preservation* term, that prevents the embedding components Z_j from collapsing to a constant, and a *covariance minimization* term that prevents redundant information from being encoded between any pair of embedding components Z_j and Z_k ($j \neq k$).

The resulting loss function is defined as

$$\mathcal{L}^{\mathrm{SSL}}(\theta) = \frac{\lambda}{n}\left\|Z_\theta - Z'_\theta\right\|_2^2 + \frac{\nu}{nd}\left\|K_\theta - \mathrm{diag}(K_\theta)\right\|_F^2 + \frac{\mu}{d}\,\mathrm{Tr}\Big(\max\Big(0, \eta - \sqrt{\mathrm{diag}(K_\theta) + \epsilon}\Big)\Big), \tag{1}$$

where $||\cdot||_F$ is the Frobenius norm, and we defined $K_\theta = \bar{Z}_\theta^\top \bar{Z}_\theta$, where $\bar{Z}_\theta = Z_\theta - \frac{1}{n}\sum_{i=1}^{n} Z_\theta^i$. η is the target variance, and λ, ν, and μ are the coefficients for the invariance, covariance, and variance terms respectively. The variance and covariance terms are computed symmetrically from both views Z and Z', and averaged.

2.1.2 Swapped Assignments between Multiple Views (SwAV)

SwAV (Caron et al., 2020) is an online-clustering based SSL method with the following features:

(a) *Swapped prediction*: minimizes the cross-entropy between the cluster assignment q_k of one augmented view, and the cluster prediction p'_k using the other augmented view, to ensure consistent mapping of all views to the same cluster.

(b) *Online clustering*: computes cluster centroids c_k for k clusters, and optimal cluster assignments q_k (preventing collapse) on the fly.

(c) *Multi-crop*: uses more than two views (usually lower resolution crops), to improve performance.

The loss function used is

$$\mathcal{L}^{\mathrm{SSL}}(\theta) = -\sum\nolimits_k q_{\theta k} \log p'_{\theta k} \ , \tag{2}$$

$$\text{where} \quad p'_{\theta k} = \frac{\exp\left(\frac{1}{\tau}\tilde{z}'^{\top}_{\theta} c_{\theta k}\right)}{\sum_{k'} \exp\left(\frac{1}{\tau}\tilde{z}'^{\top}_{\theta} c_{\theta k'}\right)} \quad \text{and} \quad \tilde{z}'_\theta = \frac{z'_\theta}{\left\|z'_\theta\right\|_2},$$

where τ is a temperature parameter, and the loss is computed over all data cases and augmentations.

2.1.3 Simple Siamese Representation Learning (SimSiam)

SimSiam (Chen and He, 2021) is a feature distillation based SSL method with the following features:

(a) *Similarity*: maximizes the cosine similarity between the embedding of one augmented view and the *predicted* embedding from the other view.

(b) *Asymmetry*: uses asymmetric Siamese branches with a predictor on one, and a *stopgrad* on the other, to allow gradient flow through only one branch at a time.

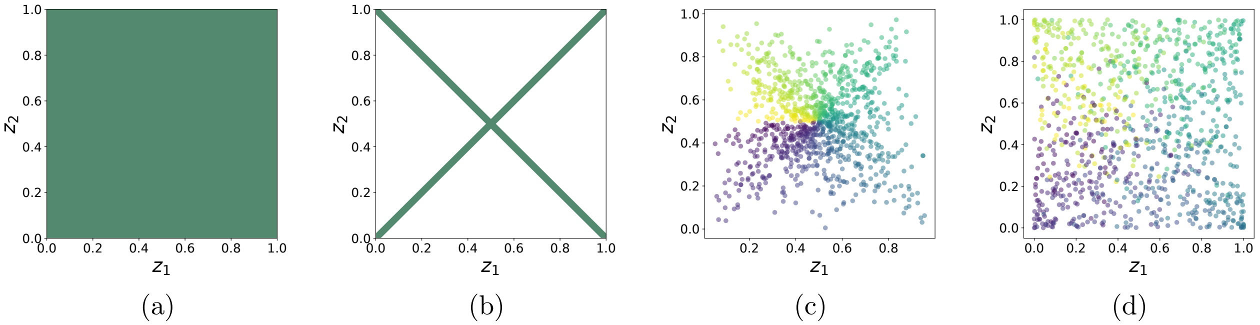
The loss function used is

$$\mathcal{L}^{\mathrm{SSL}}(\theta) = -\frac{p_\theta(Z_\theta)}{\left\|p_\theta(Z_\theta)\right\|_2} \cdot \frac{Z'_\theta}{\left\|Z'_\theta\right\|_2} \ , \tag{3}$$

where p_θ is a MLP that predicts the embedding Z' from Z, and $\|\cdot\|_2$ is the $l2$-norm. The loss is computed symmetrically for both views Z and Z'.

2.2 Differential entropy estimation of one-dimensional random variables

A key ingredient of our method is the entropy estimation of the one-dimensional marginals of our embeddings. Differential entropy estimation of one-dimensional distributions from a sample has a variety of solutions (see, for example, Beirlant et al., 1997). In this work, we use the m-spacings entropy estimate (Vasicek, 1976). This estimator is statistically efficient, easy to compute, differentiable, and has been successfully applied to the independent components analysis (ICA) problem (Learned-Miller and Fisher, 2003). Though self-supervised embeddings are typically high dimensional, we will later show how we can leverage this one-dimensional estimator in our maximum-entropy criterion.



(a) (b) (c) (d)

Figure 2: **(a)**. A 2-*d* uniform distribution. **(b)** An "X" distribution. Both (analytic) distributions have uniform (max-entropy) marginals and decorrelated components, and minimize our loss function. **(c)** Example 2-d marginal distribution over a random pair from VICReg (Bardes et al., 2021) (after transformation to compact space). **(d)** Our embeddings over the same pair of dimensions, where empirical results show, to our surprise, distributions with uniform 2-*d* marginals despite the fact that this is not *explicitly* enforced by our loss. The colors denote the relative positions of actual points in embedding space *before* (c) and *after* (d) the application of our maximum-entropy criterion demonstrating how they are spread out by our method.

The m-spacings estimator of a one-dimensional distribution's differential entropy is defined as

$$\widehat{\mathcal{H}}_j(\check{Z}^1, ..., \check{Z}^n) \qquad (4)$$
$$= \frac{1}{(n-m)} \sum\nolimits_{i=1}^{n-m} \log\left(\frac{n+1}{m}\left(\check{Z}_j^{(i+m)} - \check{Z}_j^{(i)}\right)\right)$$

for the jth dimension of $\check{Z}$, where $\check{Z} \in [0,1]^D$ is the compact version of the embedding Z (details in Sec. 3.2). Parenthetical superscripts indicate the position in the ordering $\check{Z}_j^{(1)} \leq \check{Z}_j^{(2)} \leq \cdots \leq \check{Z}_j^{(n)}$, and $\check{Z}_j^{(i+m)} - \check{Z}_j^{(i)}$ is known as a spacing of order m (typically $m = \sqrt{n}$).

3 A MAXIMUM-ENTROPY AUGMENTATION CRITERION

In this section, we motivate a simple maximum-entropy augmentation criterion that can be used to improve already-trained SSL embeddings with only a handful of epochs of continued pre-training.

Unlike all the other methods of which we are aware, we focus only on properties of the one- and two-dimensional marginal distributions, and speculate that by focusing on properties that are more reliably estimated with moderate sample sizes, we might be able to obtain a more useful criterion.

To motivate an effective maximum-entropy criterion, we start with an observation that the following facts about distributions over the unit cube are mathematically equivalent (Cover and Thomas, 1991):

1. The joint distribution has maximum joint entropy.
2. The joint distribution is uniform.
3. The one-dimensional marginal distributions are maximum entropy (i.e., uniform) *and* the components are mutually independent.

We use the third characterization to design our loss function. This characterization for formulating a self-supervised learning criterion requires (i) an effective approach to estimating the entropy of one-dimensional marginal distributions and (ii) a method for encouraging mutual independence.

To obtain a good estimate of the marginal entropies, we leverage the m-spacings estimator (c.f. Section 2.2). Unfortunately, mutual independence of the components is a property of the joint distribution, and we believe that it is too high-dimensional to achieve directly. Instead, we consider a *necessary, but not sufficient, condition* for mutual independence: decorrelation of all pairs of embedding dimensions. Criteria that serve this purpose have been used in both VICReg and other SSL methods to attempt to move embeddings towards independent features (Bardes et al., 2021; Mialon et al., 2022) but not, to our knowledge, in conjunction with the idea of maximizing marginal entropies.

Unfortunately, enforcing decorrelation of all pairs of embedding dimensions does not guarantee mutual statistical independence. Hence, maximizing marginal entropies while decorrelating embedding dimensions is not sufficient to guarantee maximum entropy of the joint distribution. We nevertheless press on and ask:

> What kinds of distributions have maximal marginal entropy *and* are decorrelated but do not have maximum joint entropy?

Consider Figure 2. Part (a) shows a two-dimensional uniform distribution, which maximizes the joint entropy and minimizes our loss function. Part (b) is what we call the "X" distribution, which also has uniform marginals and diagonal covariance (i.e., no correlations between components). In principle, either of these distributions could emerge under the criterion described above. Part (c) shows a 2-d marginal of VICReg, which is clearly non-uniform. Surprisingly, our loss, which enforces uniformity only of 1-d marginals, also produces nearly uniform 2-d marginals as shown in (d), instead of alternatives like the "X" distribution. One possible explanation could be that the inductive bias of such deep networks might make it difficult to produce non-smooth distributions like the "X" distribution.

3.1 Specifying a maximum-entropy augmentation criterion

In this section, we formalize the specific criterion from the discussion above. To define this criterion, we first transform embedding samples $Z \in \mathbb{R}^d$ to lie in a compact space, and consider the transformed embedding random variable $\check{Z} \in [0,1]^d$ instead, for applying our criterion. We defer the discussion of this transformation to Section 3.2. Finally, given an arbitrary SSL method pre-trained using loss function $\mathcal{L}^{\text{SSL}}(\theta)$, we define the constrained optimization problem

$$\min_\theta \mathcal{L}^{\text{SSL}}(\theta) \qquad \text{subject to} \quad \mathcal{L}^{\text{Entropy}}(\theta) \geq C_1, \\ \text{and} \quad \mathcal{L}^{\text{Covariance}}(\theta) \leq C_2. \quad (5)$$

In practice, we express this objective equivalently as

$$\mathcal{L}(\theta) = \mathcal{L}^{\text{SSL}}(\theta) - \beta \mathcal{L}^{\text{Entropy}}(\theta) + \gamma \mathcal{L}^{\text{Covariance}}(\theta), \quad (6)$$

where $\beta, \gamma \in \mathbb{R}$ are hyperparameters. For transformed embeddings $\check{Z}_\theta$ and $\check{Z}'_\theta$ of views X and X', we have

$$\mathcal{L}^{\text{Entropy}}(\theta) \quad (7) \\ = \frac{1}{d} \sum_{j=1}^{d} \Big(\widehat{\mathcal{H}}_j(\check{Z}_\theta^1, \cdots, \check{Z}_\theta^n) + \widehat{\mathcal{H}}_j(\check{Z}_\theta'^1, \cdots, \check{Z}_\theta'^n) \Big).$$

And, letting $\check{\check{Z}}_\theta = \check{Z}_\theta - \frac{1}{n}\sum_{i=1}^{n} \check{Z}_\theta^i$ and $\check{\check{Z}}'_\theta = \check{Z}'_\theta - \frac{1}{n}\sum_{i=1}^{n} \check{Z}_\theta'^i$ for X and X', we have

$$\mathcal{L}^{\text{Covariance}}(\theta) \quad (8) \\ = \frac{1}{nd}\big(\|(K_\theta - \text{diag}(K_\theta))\|_F^2 + \|(K'_\theta - \text{diag}(K'_\theta))\|_F^2 \big),$$

where $||\cdot||_F$ is the Frobenius norm, and we defined $K_\theta = \check{\check{Z}}_\theta^\top \check{\check{Z}}_\theta$ and $K'_\theta = \check{\check{Z}}_\theta'^\top \check{\check{Z}}'_\theta$.

We estimate the marginal entropies $\widehat{\mathcal{H}}_j$ for each embedding dimension j using the m-spacings estimator (c.f. Section 2.2) and average them in the final loss. We estimate the sample covariance for every pair of embedding dimensions j, k and $j \neq k$ using the same estimator as VICReg (c.f. Section 2.1.1, Appendix C). Under this formulation, maximizing $\mathcal{L}^{\text{Entropy}}(\theta)$ maximizes the marginal entropies, and minimizing $\mathcal{L}^{\text{Covariance}}(\theta)$ corresponds to minimizing the squared off-diagonal entries of the sample covariance computed from the embedding.

3.2 Transformation to a compact space

Maximizing entropy on a non-compact space such as $\mathbb{R}^d$ is not meaningful, since the data can simply be spread out without bound. That is, our methods are meaningfully applied only on compact spaces. We discuss the maximization of entropy on two compact spaces: the unit hypercube and the surface of the unit hypersphere. We begin with the hypercube.

For SSL methods that produce embeddings in $\mathbb{R}^d$ and do not normalize their final embeddings (e.g., VICReg), we construct a transformation $\Psi : \mathbb{R} \to [0,1]$, and apply it to every embedding component $\check{Z}_j = \Psi(Z_j)$, such that the transformed embedding $\check{Z} = [\check{Z}_1, \cdots, \check{Z}_d]$ lies in a unit hypercube of d dimensions, with an implicit joint distribution $p(\check{z}_1, ..., \check{z}_d)$ over the hypercube. We simply let Ψ be the sigmoid transformation, $\Psi(Z_j) = 1/(1 + \exp(-Z_j))$ and apply our loss function to this transformed embedding.

However, methods that do normalize their final embeddings to be on the hypersphere (e.g., SwAV, SimSiam) present a unique challenge. In particular, if we produce uniform marginal embeddings and then normalize, the resulting distribution on the hypersphere will be far from uniform. In particular, mass will be much greater in directions corresponding to the corners of the hypercube, since the projections there will accumulate density along the longer diagonal directions of the hypercube. How then can we construct Ψ such that maximizing the entropy of the compact embeddings $\check{Z}$ translates to a uniform distribution on the hypersphere when the original embeddings Z are normalized?

To answer this question, we use a simple result that is often used to draw samples uniformly from the surface of a hypersphere (Muller, 1959): If we construct an embedding vector Z whose components Z_j are independent zero-mean, unit-variance Gaussians $Z_j \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$, then the normalized embedding vector $\tilde{Z} = Z/\left\|Z\right\|_2$ maps uniformly onto the surface of the unit hypersphere $\mathcal{S}^{d-1}$. In practice, we apply this result by letting Ψ be the cumulative density function (CDF) of the zero-mean, unit-variance Gaussian, $\Psi(Z_j) = 0.5(1 + \text{erf}(x/\sqrt{2}))$, and then applying our entropy maximization criterion to the transformed embeddings to produce a uniform distribution. This is possible due to the *probability integral transform*, in which a continuous random variable is mapped through its own CDF to become uniformly distributed. This implies that the distribution over transformed variables $p(\check{z}_j) \overset{\mathrm{d}}{=} \mathcal{U}[0,1]$ if and only if the distribution over original embeddings $p(z_j) \overset{\mathrm{d}}{=} \mathcal{N}(0,1)$. Our criterion thus ensures that the components of the embedding distribution prior to normalization are normal with zero mean and unit variance. In addition, our term to minimize correlation helps to minimize dependencies among the unit variance marginals.

Using these two methods of transforming to compact spaces, our criterion can be applied to SSL methods irrespective of their normalization strategy.

4 RELATED WORK

Here we mainly review prior approaches to improving self-supervised embeddings using explicit information maximization objectives, and how they relate to ours.

Log determinant maximization. Ozsoy et al. (2022) propose *CorInfoMax*, that maximizes mutual information between similar views in latent space, while preventing dimensional collapse by spreading embeddings in this space. They use the log-determinant mutual information as a second-order approximation of the mutual information, and maximize the log-determinant of the covariance matrix as a measure of the spread of latent vectors (under a Gaussian model assumption). Liu et al. (2022) also posit that the most generalizable embeddings should have the maximum possible entropy in order to avoid bias from pretext tasks. They maximize a related objective—minimum coding length—as a computationally tractable surrogate for entropy of high dimensional embeddings, and offer a simplified version that is less computationally demanding, but suffers from the same distributional assumptions as before. Shwartz-Ziv et al. (2023) show that, under the assumption that the input data is a mixture of Gaussians, VICReg maximizes an upper bound to the embedding entropy, through an approximation of the log determinant of the covariance matrix. We show that our criterion outperforms these methods without making distributional assumptions while also relying solely on first and second order statistics.

Manifold capacity. In another recent method, *Maximum Manifold Capacity Representations*, Yerxa et al. (2023) proposed that image representations should lie on compact sub-manifolds in the representation space that are well separated from each other. Their method favors low-rank manifolds constructed by averaging embeddings from multiple views of an image, and then maximizing the nuclear norm of the embedding matrix rather than the log-determinant of the covariance matrix like the previous approaches. Schaeffer et al. (2024) show that this method also maximizes a lower bound on the mutual information between input views. In practice, we find that this method cannot be used as an add-on criteria in the continued pretraining setting like ours, lowering the performance of base SSL methods.

Noise as targets. In *Noise as Targets* (NAT), Bojanowski and Joulin (2017) propose to map input samples to a fixed set of embeddings uniformly sampled from the surface of a hypersphere. Though this approach has similarities with our objective, an important difference is that NAT strives to match an empirical *sample* from a uniform distribution, whereas our loss function enforces properties of a true uniform distribution. In *Alignment and Uniformity on the Hypersphere* (AUH), Wang and Isola (2020) propose a way to distribute embeddings by minimizing the energy configuration of points using pairwise Gaussian potentials. They show that the only distribution from which their objective admits samples, in the limit, is the uniform distribution. Zheng et al. (2022) generalize their method further by minimizing the maximum mean discrepancy (MMD) between the embedding distribution and uniform distribution using rotation invariant kernels instead of the RBF kernel.

Our *maximum-entropy* criterion (E2MC) also mimics certain properties of a fully uniform distribution on a compact embedding space. The key difference between these methods and ours is that we enforce properties of the one- and two-dimensional marginals of the distribution, rather than operating directly on properties of the joint distribution. We also find in our experiments that AUH fails to produce improvements as large as ours in the challenging continued pretraining setting. We hypothesize that since AUH evaluates energy potentials of the high-dimensional joint distribution rather than of single-dimensional marginal distributions, it is less sample efficient. Our method also does not rely on training from scratch unlike all of the above methods, and is able to adapt existing SOTA embeddings for better downstream performance. Works like EMP-SSL (Tong et al., 2023) show this is increasingly important in the realm of extant resource intensive SSL methods.

High-dimensional estimators of differential entropy. Recent advances have also been made in developing high-dimensional estimators of differential entropy for regularizing neural network training (Pichler et al., 2022; Nilsson et al., 2024). While these methods propose an upper bound entropy estimator, it is still subject to the data inefficiency of plug-in estimators in even moderately high dimensions (Goldfeld et al., 2019). It is unclear therefore, how their approach would scale to dimensions as large as 8,192 (the highest dimensionality addressed in our paper). These methods also require fitting a neural network with additional parameters and training with large batch sizes, whereas ours does not.

Table 1: **Evaluation on ImageNet.** We report Top-1-Accuracy (%) on ImageNet validation set using classifiers trained on SSL embeddings, before and after continued pre-training. Best result in each category is marked bold if a clear winner exists, along with standard errors over three random trials. We run statistical significance tests in Appendix D.1. Results marked with † are taken from the original papers, and * are reproduced results that differ from reported results in the original paper despite best attempts. Note: No numbers were reported for SimSiam in the semi-supervised setting, and hence omitted.

		Linear Evaluation			Semi-supervised learning	
Method	Epoch	1% labels	10% labels	100% labels	1% labels	10% labels
VICReg base (Bardes et al., 2021)	1,000	53.50 ±0.11	66.57 ±0.02	73.20†	54.53*±0.12	67.97*±0.03
VICReg continued	1,010	53.51 ±0.07	66.57 ±0.06	73.16 ±0.02	–	–
VICReg + E2MC (**ours**)	1,010	**54.54** ±0.05	**66.82** ±0.05	**73.45** ±0.07	**55.05** ±0.08	**68.12** ±0.04
SwAV base (Caron et al., 2020)	400	52.34 ±0.07	67.61 ±0.02	74.30†	52.57 ±0.15	69.25 ±0.05
SwAV continued	410	52.31 ±0.07	67.56 ±0.05	74.31 ±0.02	–	–
SwAV + E2MC (**ours**)	410	**53.40** ±0.01	**67.73** ±0.03	**74.44** ±0.03	52.70 ±0.54	69.24 ±0.02
SwAV base (Caron et al., 2020)	800	53.70 ±0.05	68.86 ±0.03	75.30†	53.89†±0.13	70.22†±0.05
SwAV continued	810	53.69 ±0.05	68.87 ±0.04	75.32 ±0.01	–	–
SwAV + E2MC (**ours**)	810	**55.27** ±0.07	**68.98** ±0.02	**75.41** ±0.02	53.94 ±0.30	70.32 ±0.05
SimSiam base (Chen and He, 2021)	100	43.71 ±0.04	60.15 ±0.02	68.37*	–	–
SimSiam continued	110	43.78 ±0.05	60.23 ±0.08	68.45 ±0.08	–	–
SimSiam + E2MC (**ours**)	110	43.78 ±0.06	60.23 ±0.07	**68.52** ±0.05	–	–

5 EMPIRICAL EVALUATION

Our experimental setup uses a three-stage approach (c.f. Figure 1 for an overview):

1. **Selecting** a base SSL method with publicly available checkpoints,
2. **Continued pre-training** on the base dataset using the base SSL method augmented with our criterion, and
3. **Evaluating** the representations learned by the backbone network using classifiers trained on downstream datasets.

While our approach can be used with *any* joint-embedding SSL method, we focus on three popular methods—VICReg, SwAV, and SimSiam—to demonstrate the versatility of our criterion. These methods do not require negative examples, work well with small batch sizes, and have official checkpoints and code that can be modified to incorporate our criterion.

In the continued pre-training stage, we train on the same dataset that was used for pre-training the base SSL method (ImageNet; Russakovsky et al., 2015), but with a fixed reduced learning rate and batch size. We train for *exactly* 10 additional epochs using the prescribed criterion, and report results using this updated model. All other hyperparameters associated with the base SSL method, including data augmentations, optimizers, and loss coefficients, are kept identical. This allows us to treat the base SSL method as a black box, and leaves us with only a few hyperparameters to tune for our method, namely the coefficients β and γ for our proposed loss in Equation (6). See Appendix B for implementation details and pseudocode (Algorithm 1).

5.1 Evaluation on ImageNet

We evaluate methods on ImageNet (Russakovsky et al., 2015) using the final representations (2048-d) from the ResNet-50 backbone (He et al., 2016) in two ways: (i) *linear* evaluation by training a classifier on top of the frozen backbone, and (ii) *semi-supervised* evaluation by finetuning the whole backbone with a classifier on a subset of available labels. Table 1 shows the top-1 accuracy on the ImageNet validation set of classifiers trained using different subsets of available labels (predefined by Chen et al., 2020). In the linear evaluation scenario, our method "[SSL]+E2MC" surpasses almost all baselines from just 10 epochs of continued pre-training, with larger gains in label deficient (1% subset) settings. For SwAV, the maximum-entropy *updated* 400-epoch model in the 1% setting has comparable performance with the 800-epoch base model (53.4 vs 53.7 respectively), which suggests our method has fast convergence obviating the need for longer pre-training. In the semi-supervised learning scenario, our method outperforms the VICReg baseline, and is on par with SwAV. Note that this setting is more challenging to show improvements on, as fine-tuning the backbone can change the final embeddings significantly, and subvert any benefits from learning maximum-entropy embeddings. On SimSiam, our method shows little improvement over the baseline, which could be because the SimSiam checkpoint is *less* trained compared to other methods (only 100 epochs), and benefits from our method may only emerge late in the pre-training phase where base models already have near-optimal performance.

Table 2: **Transfer Learning.** We report Top-1-Accuracy (%) of linear classifier trained on iNat18, and mAP of linear SVM trained on VOC07 datasets. Best results shown in bold. Results marked with † are taken from the original papers, and * are reproduced results that differ from reported results in the original paper despite best attempts.

Method	Epoch	iNat18	VOC07
VICReg base	1,000	47.00†	86.60†
VICReg + E2MC	1,010	**47.18** ±0.11	**86.80**
SwAV base	400	46.00	**88.38**
SwAV + E2MC	410	**46.71** ±0.17	88.24
SwAV base	800	49.08*	88.56*
SwAV + E2MC	810	**49.72** ±0.20	**88.69**
SimSiam base	100	38.75	**84.62**
SimSiam + E2MC	110	**38.99** ±0.20	84.54

Table 3: **Effect of different continued pre-training criteria.** We report Top-1-Accuracy (%) of linear classifier trained on ImageNet, using SwAV base (800 epochs) (Caron et al., 2020) and SwAV + *other criteria* (810 epochs) including [(a)] Bardes et al. (2021), [(b)] Yerxa et al. (2023), [(c)] Wang and Isola (2020), and [(d)] ours. Best results are shown in bold, and second best are underlined. Details of these alternative criteria can be found in Appendix B.

Method	1% labels	10% labels	100% labels
SwAV base	53.70 ±0.05	68.86 ±0.03	75.30†
SwAV continued	53.69 ±0.05	68.87 ±0.04	75.32 ±0.01
SwAV + VCReg[(a)]	$\underline{54.02}$ ±0.05	$\underline{68.88}$ ±0.03	$\underline{75.36}$ ±0.02
SwAV + MMCR[(b)]	53.30 ±0.02	68.77 ±0.04	75.27 ±0.01
SwAV + AUH[(c)]	53.84 ±0.07	68.90 ±0.04	75.33 ±0.01
SwAV + E2MC[(d)]	**55.27** ±0.07	**68.98** ±0.02	**75.41** ±0.02

5.2 Transfer Learning on Other Datasets

Following Goyal et al. (2019) and Misra and Maaten (2020), we show how our updated representations transfer to downstream tasks on other datasets. Table 2 shows top-1 accuracy from linear classification on the challenging iNat18 dataset (Van Horn et al., 2018) (with over 8000 fine-grained classes), and mAP for multi-label object classification on VOC07 (Everingham et al., 2010). Again, we show consistent improvements for iNat18, and better or comparable performance to base models on VOC07.

5.3 Ablation Study: Effect of different continued pre-training criteria

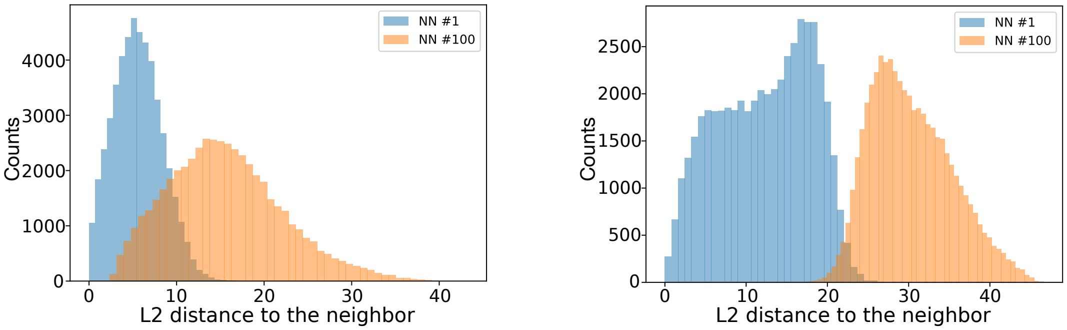
To disambiguate performance gains of continued pre-training using our method from continued pre-training using other criteria, we train the base SSL method for the same number of epochs as our method, but with the other criteria.

One important setting is to simply train the base method longer using only its original loss function(s) ("[SSL] continued"). As seen in Table 1, continued training with base criteria produces no additional gains for any of the SSL methods under consideration.

Another important setting is where we replace our maximum-entropy criteria with similar criteria proposed by others. Table 3 shows the results of doing continued pre-training on SwAV with other criteria. We chose the SwAV model for these experiments as it has several features that are amenable to the other criteria we experiment with. VCReg (Bardes et al., 2021) uses covariance minimization similar to our criterion, but variance regularization produces inferior gains compared to entropy maximization. The uniformity loss proposed in AUH (Wang and Isola, 2020) also fails to produce improvements as large as our method, or even VCReg, and we hypothesize that their method is less sample efficient because it optimizes properties of the high dimensional joint distribution, instead of lower dimensional criteria such as ours. Finally, MMCR (Yerxa et al., 2023) uses an alternative information criterion compared to these methods, and we show that it degrades the performance of the base model instead of improving it. This suggests that not all information criteria are created equal, and some such as MMCR, though good standalone methods, do not play as nicely with popular SSL criteria.

5.4 Ablation study: Embedding separability under different criteria

One of the advantages of a maximum-entropy embedding is that data is well separated in the embedding space thereby preserving discriminability for downstream tasks. Inspired by Sablayrolles et al. (2018), in Figure 3, we show the histogram of distances to the nearest and 100th nearest neighbors for all points in the ImageNet validation set, computed using VICReg embeddings before (left) and after (right) continued pre-training using our maximum-entropy criterion. We note that for VICReg, the two histograms have significant overlap signifying that the 100th nearest neighbor for a point is often closer than the 1st nearest neighbor for another point, suggesting low separability that affects performance on downstream task. Continued pre-training using our criterion significantly reduces this overlap ensuring greater discriminability for downstream tasks.



(a)

(b)

Figure 3: Histograms of distances from a query point to its nearest neighbor (blue) and 100^{th} nearest neighbor (orange) for (a) VICReg and (b) VICReg + E2MC. For VICReg, the histograms have significant overlap indicating less separability between points in embedding space. We show that continued pre-training with E2MC reduces the overlap and ensures greater separability between points for downstream tasks.

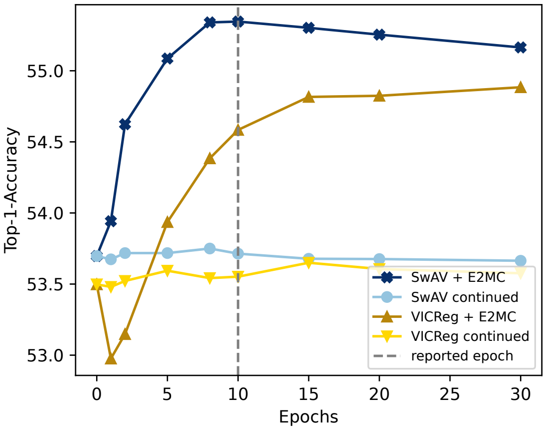


Figure 4: Top-1-Accuracy of a linear classifier trained using 1% ImageNet labels at different epochs of continued pre-training. Continued pre-training with our criteria (E2MC) outperforms other baselines, and performance beyond the reported ten epochs either improves marginally or degrades depending on the method.

5.5 Comparative analysis: Number of continued pre-training Epochs

In this experiment, we show results from intermediate epochs of continued pre-training for a total of 30 epochs (i.e., 3× longer training than the reported results in Table 1). In Figure 4, we show the performance of SwAV (800 epochs) and VICReg (1000 epochs) base models at epoch 0, and how they evolve during continued pre-training with our E2MC criterion compared to their base criterion. It is evident that simply training longer using the base criterion (continued) provides no added benefit. We also notice a rapid improvement in performance until epoch 10 after which the performance only improves marginally (e.g., for VICReg) or starts degrading (e.g., for SwAV), and therefore recommend 10 epochs of continued pre-training as a good trade-off between performance and training time.

6 DISCUSSION

ResNet-50 models pre-trained with SSL methods are the workhorse of computer vision applications across industry. Despite significant efforts and resources devoted to the development of effective pre-training methods, the downstream performance of pre-trained ResNet-50 models had plateaued. We have presented a simple method to maximize the entropy of pre-trained embeddings that consistently improves the performance of already-highly-optimized publicly available pre-trained ResNet-50 models with only a handful of training epochs. In downstream tasks where only a small number of training samples is available, our method can be used to leverage the full potential of state-of-the-art models with rapid continued pretraining on as few as one GPU within a day. We also show that other methods for maximizing entropy do not converge as quickly as our method in this challenging setting, likely because they rely on high dimensional statistics or make fundamental assumptions about the underlying data distribution. Moreover, by showing improvements on a variety of SSL methods, we hope that our general-purpose criterion can be extended to larger transformer-based models in the future, and squeeze any remaining gains out of them.

7 CONCLUSION

We proposed a simple add-on criterion for self-supervised learning motivated by information-theoretic principles and applicable to a wide variety of SSL methods. We demonstrated empirically that the proposed criterion has desirable properties and that—with only a handful of epochs of continued pre-training—it is possible to achieve consistent and, in some cases, significant improvements in downstream-task performance across a selection of computer vision tasks.

Acknowledgments

The authors would like to thank Dmitry Petrov, Fabien Delattre, and Edmond Cunningham for helpful discussions and feedback. We would also like to thank the anonymous reviewers for their invaluable comments and suggestions to improve our work. Part of this work utilized resources from Unity, a collaborative, multi-institutional high-performance computing cluster managed by UMass Amherst Research Computing and Data.

References

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

Jan Beirlant, Edward J Dudewicz, László Györfi, Edward C Van der Meulen, et al. Nonparametric entropy estimation: An overview. *International Journal of Mathematical and Statistical Sciences*, 6(1): 17–39, 1997.

Piotr Bojanowski and Armand Joulin. Unsupervised learning by predicting noise. In *International Conference on Machine Learning*, pages 517–526. PMLR, 2017.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a" siamese" time delay neural network. *Advances in Neural Information Processing Systems*, 6, 1993.

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.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pages 1597–1607. PMLR, 2020.

Xinlei Chen and Kaiming He. Exploring simple Siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory.* Wiley, New York, 1991.

Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The Pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88:303–338, 2010.

Ziv Goldfeld, Kristjan Greenewald, Jonathan Weed, and Yury Polyanskiy. Optimality of the plug-in estimator for differential entropy estimation under gaussian convolutions. In *2019 IEEE International Symposium on Information Theory (ISIT)*, pages 892–896. IEEE, 2019.

Phillip Good. *Permutation tests: a practical guide to resampling methods for testing hypotheses.* Springer Science & Business Media, 2013.

Priya Goyal, Dhruv Mahajan, Abhinav Gupta, and Ishan Misra. Scaling and benchmarking self-supervised visual representation learning. In *Proceedings of the IEEE/CVF International Conference on computer vision*, pages 6391–6400, 2019.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020.

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*, pages 770–778, 2016.

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.

Erik Learned-Miller and John W. Fisher III. ICA using spacings estimates of entropy. *The Journal of Machine Learning Research*, 4:1271–1295, 2003.

Xin Liu, Zhongdao Wang, Ya-Li Li, and Shengjin Wang. Self-supervised learning via maximum entropy coding. *Advances in Neural Information Processing Systems*, 35:34091–34105, 2022.

David McAllester and Karl Stratos. Formal limitations on the measurement of mutual information. In *International Conference on Artificial Intelligence and Statistics*, pages 875–884. PMLR, 2020.

Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947.

Grégoire Mialon, Randall Balestriero, and Yann LeCun. Variance covariance regularization enforces pairwise independence in self-supervised representations. *arXiv preprint arXiv:2209.14905*, 2022.

Erik G Miller. A new class of entropy estimators for multi-dimensional densities. In *2003 IEEE Inter-*

national Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03)., volume 3, pages III–297. IEEE, 2003.

Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6707–6717, 2020.

Mervin E. Muller. A note on a method for generating points uniformly on n-dimensional spheres. *Communications of the ACM*, 2(4):19–20, 1959.

Viktor Nilsson, Anirban Samaddar, Sandeep Madireddy, and Pierre Nyquist. Remedi: Corrective transformations for improved neural entropy estimation. In *International Conference on Machine Learning*, pages 38207–38236. PMLR, 2024.

Serdar Ozsoy, Shadi Hamdan, Sercan Arik, Deniz Yuret, and Alper Erdogan. Self-supervised learning with an information maximization criterion. *Advances in Neural Information Processing Systems*, 35:35240–35253, 2022.

Georg Pichler, Pierre Jean A Colombo, Malik Boudiaf, Günther Koliander, and Pablo Piantanida. A differential entropy estimator for training neural networks. In *International Conference on Machine Learning*, pages 17691–17715. PMLR, 2022.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115:211–252, 2015.

Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, and Hervé Jégou. Spreading vectors for similarity search. *arXiv preprint arXiv:1806.03198*, 2018.

Rylan Schaeffer, Victor Lecomte, Dhruv Bhandarkar Pai, Andres Carranza, Berivan Isik, Alyssa Unell, Mikail Khona, Thomas Yerxa, Yann LeCun, SueYeon Chung, et al. Towards an improved understanding and utilization of maximum manifold capacity representations. *arXiv preprint arXiv:2406.09366*, 2024.

Ravid Shwartz-Ziv, Randall Balestriero, Kenji Kawaguchi, Tim GJ Rudner, and Yann LeCun. An information-theoretic perspective on variance-invariance-covariance regularization. *arXiv preprint arXiv:2303.00633*, 2023.

Shengbang Tong, Yubei Chen, Yi Ma, and Yann Lecun. Emp-ssl: Towards self-supervised learning in one training epoch. *arXiv preprint arXiv:2304.03977*, 2023.

Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The iNaturalist species classification and detection dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8769–8778, 2018.

Oldrich Vasicek. A test for normality based on sample entropy. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 38(1):54–59, 1976.

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

Thomas Yerxa, Yilun Kuang, Eero Simoncelli, and SueYeon Chung. Learning efficient coding of natural images with maximum manifold capacity representations. *Advances in Neural Information Processing Systems*, 36:24103–24128, 2023.

Léon Zheng, Gilles Puy, Elisa Riccietti, Patrick Pérez, and Rémi Gribonval. Self-supervised learning with rotation-invariant kernels. *arXiv preprint arXiv:2208.00789*, 2022.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [No]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [No, but we provide pytorch-pseudocode for the main method in Algorithm 1.]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Not Applicable]
   (b) Complete proofs of all theoretical results. [Not Applicable]
   (c) Clear explanations of any assumptions. [Not Applicable]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results

(either in the supplemental material or as a URL). [Yes, we provide pseudocode in Algorithm 1, and full code is available at `https://github.com/deepc94/e2mc`]

(b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes, they can be found in Appendix B.]

(c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes, standard errors over three trials and limited tests of statistical significance are reported.]

(d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes, in Appendix B]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:

   (a) Citations of the creator If your work uses existing assets. [Yes]

   (b) The license information of the assets, if applicable. [Not Applicable]

   (c) New assets either in the supplemental material or as a URL, if applicable. [No]

   (d) Information about consent from data providers/curators. [Not Applicable]

   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

   (a) The full text of instructions given to participants and screenshots. [Not Applicable]

   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]

   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Appendix

Appendix A Algorithm for Continued Pre-Training with E2MC

Algorithm 1 PyTorch pseudocode for our max-entropy augmentation criterion

```
# sample training loop
for x in loader:
    # two random views of x
    x_a, x_b = augment(x)

    # compute embeddings
    z_a = f(x_a) # N x D
    z_b = f(x_b) # N x D

    # base SSL loss
    base_loss = ssl_loss(z_a, z_b)

    # our criterion
    ent_z_a, cov_z_a = max_ent_criterion(z_a)
    ent_z_b, cov_z_b = max_ent_criterion(z_b)
    ent_loss = ent_z_a + ent_z_b
    cov_loss = cov_z_a + cov_z_b

    # final loss
    loss = base_loss - beta * ent_loss + gamma * cov_loss

    # optimization step
    loss.backward()
    optimizer.step()

def max_ent_criterion(x, type):
    if type == ’hypercube’:
        # apply the sigmoid transformation
        x_hyper = torch.sigmoid(x)
    elif type == ’hypersphere’:
        # apply the CDF of 0-mean, 1 variance gaussian
        x_hyper = 0.5 * (1 + torch.erf(x / math.sqrt(2)))
    ent_loss = m_spacings_estimator(x_hyper)
    cov_loss = sample_cov_estimator(x_hyper)
    return (ent_loss, cov_loss)

def m_spacings_estimator(x):
    n = x.shape[0] # batch size
    m = round(math.sqrt(n)) # window size
    eps = 1e-7 # small constant to avoid underflow
    x, _ = torch.sort(x, dim=0) # order statistics
    x = x[m:] - x[:n - m] # m-spaced differences
    x = x * (n + 1) / m
    marginal_ents = torch.log(x + eps).sum(dim=0) / (n - m)
    return marginal_ents.mean()

def sample_cov_estimator(x):
    n, d = x.shape
    x = x - x.mean(dim=0) # mean subtraction
    cov_x = (x.T @ x) / (n - 1) # sample covariance matrix
    cov_loss = off_diagonal(cov_x).pow(2).sum().div(d)
    return cov_loss
```

Appendix B Full Implementation Details

B.1 Continued Pre-training

Continued pre-training involves starting from a pre-trained checkpoint of a base SSL method and training for *exactly* 10 epochs with an additional criteria. This stage uses 2×NVIDIA Quadro RTX8000 GPUs with 48GB VRAM for continued pre-training of each model. Training times for 10 epochs of continued pre-training are 10 hrs for VICReg, 13 hrs for SwAV, and 14 hrs for SimSiam. Due to the extremely limited number of these GPUs, we could not train models from scratch, or fine-tune with bigger batch sizes, or run extensive grid searches for hyperparameters. We will now detail the different criteria used and their associated hyperparameters for various base SSL methods.

B.1.1 VICReg

We start from the 1000-epoch checkpoint for VICReg (Bardes et al., 2021) with ResNet-50 backbone, and 3-layer MLP (8192-8192-8192) as projector architecture. For continued pretraining, our criterion is applied to the final projector embeddings Z mapped through a sigmoid transformation. We use the default coefficients for the VICReg loss function (Equation (1)) $\lambda = \mu = 25,\ \nu = 1$, and coefficients used for our loss in Eqn. (6) are $\beta = 1000$, $\gamma = 100$. We do continued pre-training for 10 epochs, with a learning rate of 0.003 (i.e., 0.01× the base learning rate used to train VICReg), batch size of 512, and all other hyperparameters left unchanged from the original method. The same settings are used for continued training ablation using the base loss only.

B.1.2 SwAV

We run experiments using both 400-epoch and 800-epoch checkpoints released for SwAV (Caron et al., 2020) with multicrop, using resnet50-backbone and 2-layer MLP (2048-128) as projector architecture. For continued pretraining, our criterion is applied to the projector embeddings Z before the cluster assignment layer and before normalization after mapping through the CDF function. We use default hyperparameters for SwAV loss (Equation (2)) namely $\tau = 0.1$, and apply our loss only to the embeddings of full-resolution crops (two views) and not the low resolution multiple crops for computational efficiency, with coefficients $\beta = 1$ and $\gamma = 25$. See Appendix D.2 for an ablation study on the coefficients. We do continued pre-training for 10 epochs, with a learning rate of 0.001 (i.e., 0.01× the base learning rate used to train SwAV), batch size of 512, and all other hyperparameters left unchanged from the original method. The same settings are used for continued training ablation without our loss.

We will now describe the ablation studies that train SwAV models further using an alternative criterion and associated hyperparameters.

Variance-Covariance Regularization (Bardes et al., 2021). We use the variance and covariance regularization losses from the VICReg objective in Equation (1), to minimize the following loss

$$\mathcal{L}(\theta) = \mathcal{L}^{\text{SSL}}(\theta) + \mu \mathcal{L}^{\text{Variance}}(\theta) + \nu \mathcal{L}^{\text{Covariance}}(\theta) \tag{B.1}$$

We set $\mu = 0.1$ and $\nu = 0.001$ and this loss is applied only on the embeddings of the two full resolution crops to be consistent with our setting. These hyperparameters were determined experimentally by searching over $\mu = [0.01, 0.1, 1, 25]$ and $\nu = [0.001, 0.005, 0.01, 0.1, 1, 25]$ on the validation set using linear classifier trained on 1% ImageNet labels.

Uniformity loss from AUH (Wang and Isola, 2020). We use the following loss:

$$\mathcal{L}(\theta) = \mathcal{L}^{\text{SSL}}(\theta) + \lambda \mathcal{L}^{\text{Uniform}}(\theta) \tag{B.2}$$

where,

$$\begin{aligned} \mathcal{L}^{\text{Uniform}}(\theta) \triangleq & \log\ \mathbb{E}_{[Z^p, Z^q \overset{\text{i.i.d.}}{\sim} p(z)]}\Big[G_t(Z^p_\theta, Z^q_\theta)\Big] \\ & + \log\ \mathbb{E}_{[Z'^p, Z'^q \overset{\text{i.i.d.}}{\sim} p(z')]}\Big[G_t(Z'^p_\theta, Z'^q_\theta)\Big], \quad t > 0 \end{aligned} \tag{B.3}$$

where $p, q \in \{1, \cdots, n\}$, and G_t is defined as the average pairwise gaussian potential between two embedding vectors:

$$G_t(Z^p_\theta, Z^q_\theta) = \exp\left(-t\|Z^p - Z^q\|_2^2\right) \tag{B.4}$$

In practice this is applied to all embedding pairs from each of the full resolution crops (each of the two views) to be consistent with our setting. The hyperparameters used are $\lambda = 0.5$ and $t = 2$, and continued pre-training is done for 10 epochs with the same training hyperparameters. The hyperparameters for this method are guided by the original paper and experimentally verified on 1%-ImageNet split as before.

Maximum Manifold Capacity Representations (MMCR; Yerxa et al. (2023)) Consider multiview embeddings $Z^{(v)} \in \mathbb{R}^{d\times k}$ for input views $v \in \{1, \cdots, V\}$. The centroid embedding C is then considered an average embedding across the views $C = \frac{1}{V}\sum_{v=1}^{V} Z^{(v)}$. We use the following loss

$$\begin{gathered}\mathcal{L}(\theta) = \mathcal{L}^{\mathrm{SSL}}(\theta) - \lambda\|C\|_* \\ \text{where} \quad \|C\|_* = \sum\nolimits_{r=1}^{rank(C)} \sigma_r(C),\end{gathered} \tag{B.5}$$

where $\|\cdot\|_*$ is defined as the nuclear norm of a matrix, and $\sigma_r(C)$ is the r-th singular value of C. This loss is applied over all views, i.e., multi-resolution crops of SwAV with coefficient $\lambda = 0.005$, and $V = 8$. The coefficient is determined by grid search over $\lambda = [0.001, 0.005, 0.1, 0.5, 1, 2]$ verified using the same 1%-ImageNet split as before.

B.1.3 SimSiam

We start from the 100-epoch checkpoints released for SimSiam (Chen and He, 2021), using resnet50-backbone and 3-layer MLP (2048-2048-2048) as projector, and 2-layer MLP (2048-512) as predictor architecture. For continued pretraining, our criterion is applied to the projector embeddings Z on both branches before the predictor, mapped through a CDF function. The projector embeddings thus serve as uniformly distributed points on the hypersphere that the predictor has to map to. We use default coefficients for SimSiam loss, and apply our loss with coefficients $\beta = 0.001$ and $\gamma = 0.01$. We do continued pre-training for 10 epochs, with a learning rate of 0.001 (i.e., 0.01× the base learning rate used to train SimSiam), batch size of 512, and all other hyperparameters left unchanged from the original method. The same settings are used for continued training ablation without our loss.

B.2 Evaluation

In this stage, the final representations from ResNet-50 backbone are used to train classifiers on different datasets in order to evaluate the quality of the representations. Training hardware includes 4×NVIDIA RTX 2080TI GPUs with 11GB VRAM for each training.

B.2.1 Linear evaluation on ImageNet.

Following standard practice, we train linear classifiers using frozen ResNet-50 representations on 1% (12,811 images), 10% (128,117 images), and 100% (1,281,176 images) of ImageNet labels (using predefined splits from (Chen et al., 2020)) for 100 epochs, and report the top-1 accuracy on the validation set containing 50,000 images and 1,000 classes.

For VICReg, we use the SGD optimizer with learning rate 0.02 and cosine decay, batch size of 256, and a weight decay of 10^{-4} for 1% and 10% splits, and 10^{-6} for 100% split respectively.

For SwAV, we use the SGD optimizer with learning rate 0.3 and cosine decay, batch size of 256, and a weight decay of 10^{-6} for all splits.

For Simsiam, we use the LARS optimizer with weight decay 0 and cosine decay for learning rate as follows. For the 1% split, we use learning rate 2.0 and batch size 256. For the 10% split, we use learning rate 0.2 and batch size 2048. For the 100% split, we use learning rate 0.1 and batch size 2048.

B.2.2 Semi-supervised learning on ImageNet.

We perform semi-supervised evaluation by finetuning the whole backbone with a classifier on a subset of available labels.

For VICReg, we use the SGD optimizer with batch size 256, cosine learning rate schedule, and no weight decay, and train for 20 epochs using learning rate 0.03 for the backbone and 0.08 for the linear classifier in the 1% labels setting, and learning rate 0.01 for the encoder and 0.1 for the linear classifier in the 10% labels setting. Unfortunately, we're unable to reproduce the numbers reported in the paper (54.8% and 69.5% respectively) exactly using these prescribed settings, and report our closest reproduced values in Table 1.

For SwAV, use the SGD optimizer with batch size 256, step decay of 0.2 at epochs 12 and 16 for a total of 20 epochs, and no weight decay using learning rate 0.02 for the backbone and 5 for the linear classifier in the 1% labels setting, and learning rate 0.01 for the encoder and 0.2 for the linear classifier in the 10% labels setting.

For SimSiam, the semi-supervised experiments were not conducted in the original paper, and therefore we skip this in our experiments.

B.2.3 Transfer learning performance on other datasets.

Following (Misra and Maaten, 2020; Goyal et al., 2019), we show how representations updated using our method on ImageNet dataset generalize to downstream linear classification on other datasets such as iNaturalist 2018 (Van Horn et al., 2018) and Pascal VOC 2007 (Everingham et al., 2010).

For iNat18 (437,513 images and 8,142 classes), we use `res5` features from the ResNet-50 backbone (before average pooling layer) subsampled to 8192-d using an average pooling layer of size (6, 6) and stride 1, followed by a batch normalization layer. A linear classifier is then trained on top of these representations using the SGD optimizer with batch size 256, weight decay 10^{-4}, momentum 0.9, and learning rate 0.01 reduced by a factor of 10 at epochs 24, 48, and 72, for a total of 84 epochs. These hyperparameters are used consistently across all methods, and we find that for SwAV, we obtain better performance (49.72) than reported in the original paper (48.6).

For VOC07 (5,011 images and 20 classes), we train linear SVMs on top of final average pooled representations (2048-d) from ResNet-50 backbone using the VISSL library (Goyal et al., 2019) and report the mean Average Precision (mAP) of multi-label object classification on the validation set. In this setting, we were unable to reproduce numbers reported in the papers exactly due to missing hyperparameter details and default values in the library not working well. The numbers reported in the paper are using the following C values: $[0.000001, 0.000003, 0.00001, 0.00003, 0.0001, 0.0003, 0.001, 0.003, 0.01, 0.03, 0.1, 0.3, 1, 2, 5, 10,$ $15, 20, 50, 100, 200, 500, 1000]$. We get close to the performance reported in original papers, and mark our results where there are significant differences, such as SwAV (88.56 vs. the reported 88.9).

Appendix C Maximum Entropy Augmentation Criteria: Further Details

C.1 Sample Covariance

We estimate the squared off-diagonal covariance matrix entries using the sample covariance.

Let $\check{\check{Z}}_\theta = \check{Z}_\theta - \frac{1}{n}\sum_{i=1}^n \check{Z}_\theta^i$ for $\check{\check{Z}}_\theta, \check{Z}_\theta \in \mathbb{R}^{n \times d}$. Now, consider the sample covariance estimator

$$\widehat{\mathrm{Cov}}_{jk}(\check{Z}_\theta^1, ..., \check{Z}_\theta^n) = \frac{1}{n-1}\sum_{i=1}^n \check{\check{Z}}_{\theta j}^i \check{\check{Z}}_{\theta k}^i \tag{C.6}$$

for the jth and kth dimension of $\check{Z}_\theta$.

Letting $K_\theta = \check{\check{Z}}_\theta^\top \check{\check{Z}}_\theta$, we define

$$\mathcal{L}^{\text{Covariance}}(\theta) = \frac{1}{nd}\|(K_\theta - \mathrm{diag}(K_\theta))\|_F^2 = \frac{1}{d}\sum_{j=1,k=1}^d \mathbb{I}[k \neq j]\widehat{\mathrm{Cov}}_{jk}(\check{Z}_\theta^1, ..., \check{Z}_\theta^n)^2, \tag{C.7}$$

where $\|\cdot\|_F^2$ is the squared Frobenius norm.

Appendix D Further Empirical Results

D.1 Statistical Significance of Our Results

Table D.1: Results for statistical tests conducted on the ImageNet validation set using both linear evaluation and semi-supervised learning protocol. The results show the test statistic for Paired Permutation test and McNemar's test along with associated p-values. Improvements achieved by E2MC over the baseline are statistically significant.

Evaluation protocol	VICReg (1000 epochs)	VICReg + E2MC (1010 epochs)	Observed Mean Δ (μ_{E2MC} - μ_{base})	Permutation test p-value	Effect size (Cohen's d)	McNemar's test $\chi^2 = \frac{(n_{01}-n_{10})^2}{n_{01}+n_{10}}$	McNemar's p-value
Linear, 1% labels	53.50 ± 0.11	54.54 ± 0.05	0.010	< 0.001	0.05	108.79	< 0.001
Linear, 10% labels	66.57 ± 0.02	66.82 ± 0.05	0.002	0.02	0.01	5.25	0.02
Linear, 100% labels	73.20	73.45 ± 0.07	0.002	0.003	0.01	8.46	0.003
Semi-sup, 1% labels	54.53 ± 0.12	55.05 ± 0.08	0.006	< 0.001	0.03	47.38	< 0.001
Semi-sup, 10% labels	67.97 ± 0.03	68.12 ± 0.04	0.002	< 0.001	0.02	12.12	< 0.001

In this section, we evaluate the statistical significance of the improvements obtained by our method E2MC over the VICReg (1000 epochs) baseline, as reported in Table 1. To do this, we conduct a paired permutation test (Good, 2013) and McNemar's test (McNemar, 1947) on the ImageNet validation set, consisting of 50k samples. These tests were chosen to analyze differences in correctness of predictions between the two models, and prediction disagreements respectively, as we have paired data for both models. For running these tests, we picked a single model each for the baseline and for our method with performance closest to the reported mean accuracy on the ImageNet 1%-labels setting.

The p-values for both paired permutation test and McNemar's test (Table D.1) confirm that improvements on ImageNet obtained by our method under all settings are statistically significant, as we reject the null hypothesis with p-values well under 0.05 (the chosen significance level). Note that while the effect size (Cohen's d) in all these tests is small, our results are still impactful because achieving significant performance improvements on ImageNet with few epochs of continued pre-training is challenging, as most SSL models are already highly optimized.

D.2 Ablation Study: Contribution of Different Components in Our Objective Criterion

Table D.2: Contribution of entropy and covariance components controlled by their coefficients β and γ resp. We report Top-1-Accuracy (%) of linear classifier trained on ImageNet, using SwAV + E2MC method (810 epochs).

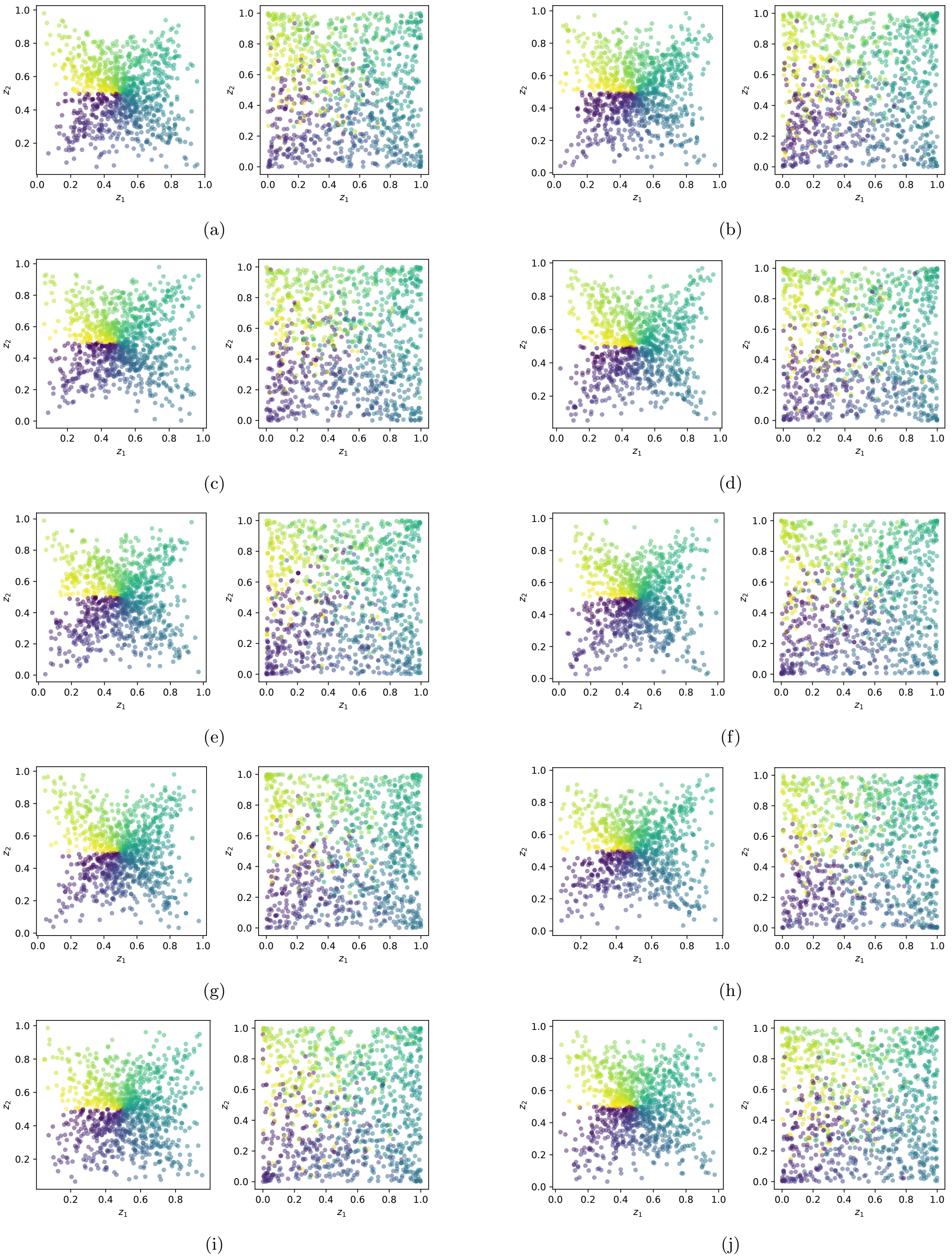
β	γ	1% labels	10% labels	Notes
0	0	53.69	68.87	Baseline (SwAV contd.) - no entropy, no covariance
1	0	53.85 (↑ 0.16)	68.81 (↓ 0.06)	Only entropy
0	1	53.63 (↓ 0.06)	68.83 (↓ 0.04)	Only covariance
1	25	**55.27 (↑ 1.58)**	**68.98 (↑ 0.11)**	Reported (SwAV + E2MC) - entropy < covariance
1	0.1	53.92 (↑ 0.23)	68.81 (↓ 0.06)	Entropy + some covariance
0.1	1	54.09 (↑ 0.40)	68.89 (↑ 0.02)	Covariance + some entropy
1	1	54.29 (↑ 0.60)	68.90 (↑ 0.03)	Entropy == Covariance
1	10	54.99 (↑ 1.30)	68.91 (↑ 0.04)	Entropy < Covariance
10	1	51.53 (↓ 2.16)	68.01 (↓ 0.86)	Entropy > Covariance
10	25	55.12 (↑ 1.43)	69.11 (↑ 0.24)	
0.1	25	53.88 (↑ 0.19)	68.86 (↓ 0.01)	Entropy ≪ Covariance
0	25	53.66 (↓ 0.03)	68.83 (↓ 0.04)	No entropy, high covariance

Here, we analyze the top-1 accuracy on the ImageNet validation set (using linear probes trained on 1% and 10% of the training labels) for continued pre-training of SwAV (800 epochs) using our proposed objective, albeit with different coefficients for the marginal entropy component and pairwise covariance component (β and γ respectively in Equation (6)). This shows the effect of removing or *reducing* the contribution of either of the two components (marginal entropy and covariance) in our proposed objective.

Our hypothesis is that for E2MC to work as intended, we need both of these components in the loss to avoid learning degenerate embedding distributions. The results shown in Table D.2 confirm that removing either one of these components ($\beta = 1, \gamma = 0$ or $\beta = 0, \gamma = 1$) either hurts performance compared to the baseline ($\beta = 0, \gamma = 0$), or achieves smaller performance improvements compared to the best possible model ($\beta = 1, \gamma = 25$). Moreover, these results also show how performance changes as a function of varying contributions from the entropy and covariance losses, leading us to conclude that our reported results are using near-optimal hyperparameters.

D.3 Sample Distributions over Two Dimensional Marginals

Figure D.1 shows more examples of sample distributions over a randomly selected pair of embedding dimensions (marginals) from VICReg *before* and *after* continued pre-training with our maximum-entropy criterion. Note how our method virtually always produces uniformly distributed marginals over any random pair even though this is not explicitly enforced by our loss. A fixed set of colors was assigned to the data points when plotting the "before" embeddings, and one can follow how the points were distributed by the application of our criterion by noting the relative distribution of colors in the "after" embeddings.



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

(i)

(j)

Figure D.1: Plots (a)-(j) show sample distributions from VICReg (left) and VICReg + E2MC (right) over a random pair of compact embedding dimensions, for a fixed set of data points.