Self-Supervised Learning with an Information Maximization Criterion

Anonymous Author(s)
Affiliation
Address
email

Abstract

Self-supervised learning allows AI systems to learn effective representations from large amounts of data using tasks that do not require costly labeling. A common self-supervision task is to match the latent representations that come from the distortions of the same input. When training a model for this task, the central problem is to avoid collapse, i.e., the model producing identical representations for all inputs. Contrastive and regularization-based learning methods have been proposed to solve the collapse problem. In this article, we argue that a straightforward application of information maximization among alternative latent representations of the same input naturally solves the collapse problem and achieves competitive empirical results. We propose a self-supervised learning method, CorInfoMax, that uses a second-order statistics-based mutual information measure that reflects the level of correlation among its arguments. Maximizing this correlative information measure between alternative representations of the same input serves two purposes: (1) it avoids the collapse problem by generating feature vectors with non-degenerate covariances; (2) it establishes relevance among alternative representations by increasing the linear dependence among them. A first-order approximation of the proposed information maximization objective simplifies to a Euclidean distance-based objective function regularized by the log-determinant of the feature covariance matrix. The regularization term acts as a natural barrier against feature space degeneracy. Consequently, beyond avoiding complete output collapse to a single point, the proposed approach also prevents dimensional collapse by encouraging the spread of information across the whole feature space. As a result, the correlative information maximization principle offers dispersed but organized representations spanning the entire latent space eliminating the need for negative samples or feature decorrelation.

1 Introduction

Self-supervised learning (SSL) is a relatively new and successful paradigm in artificial intelligence [1–6]. SSL can significantly reduce the need for expensively-obtained labeled data. Furthermore, the internal representations learned by SSL approaches are more suitable for task generalization in transfer learning applications [7]. Recently, SSL methods have been demonstrated to achieve comparable performance to their supervised counterparts [8, 9].

The primary approach for SSL is to define a pretext objective based on unlabeled data, whose optimization leads to rich and useful representations for downstream applications. For example, for visual processing, a common way to construct a pretext task is to obtain similar features for two different augmentations of the same input, which exploits the expected invariance of representations to particular transformations of the input. However, the corresponding optimization setting should avoid the (total) "collapse problem," which is described as generating the same feature output for all

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

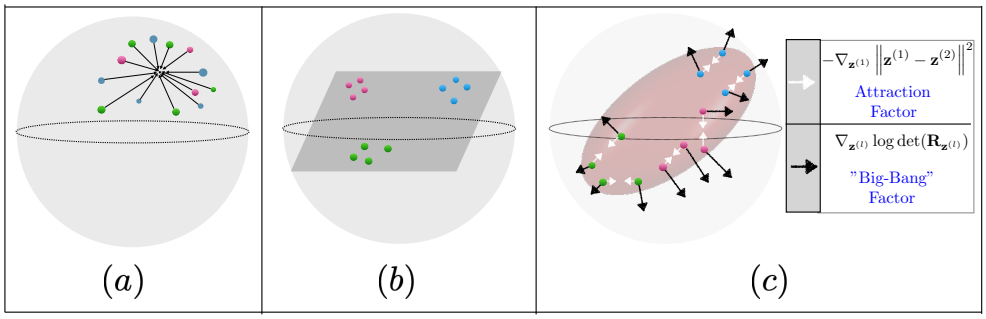

Figure 1: Depictions in latent space - (a) "Total Output Collapse": all latent vectors converge to the same point, (b) "Dimensional Output Collapse": latent vectors are restricted to a strict subspace of the latent space, (c) Gradient dynamics of the CorInfoMax based on (10): the ellipsoid surface reflects the average spread pattern of latent vectors.

inputs. Figure 1,(a) illustrates a toy view of the total collapse problem, where all latent vector outputs converge to a single point during training. We can also define its less severe version, which is referred to as "dimensional collapse," where all latent vectors lie in a strict subspace of the whole latent space as illustrated by Figure 1,(b).

This article proposes an SSL approach based on the maximization of a form of mutual information among the alternative latent representations obtained from the same input. The mutual information maximization approach would serve two fundamental purposes for SSL:

(i). (*Similarity*) It would enforce the representations obtained for the same input source to be related to or dependent on each other,

(ii). (*No Collapse*) Since the entropy of each alternative representation is lower bounded by the mutual information, maximizing mutual information ensures that latent representations have non-degenerate distributions avoiding collapse.

Therefore, the information maximization approach provides a natural solution to the problems targeted by the existing SSL approaches, which are briefly surveyed in Section 2.2.

The conventional choice of mutual information measure is Shannon Mutual Information (SMI) [10]. Within the SSL context, maximizing SMI between the representations of the same input is a challenging task whose implementation would require relatively large sample sizes [11, 12]. However, having a large batch size may not be favorable as (i) it may have an adverse impact on generalization performance [13, 14], (ii) it can significantly increase the memory requirement. Furthermore, estimating and optimizing SMI may significantly burden the computational budget. Finally, SMI maximization may induce a nonlinear dependence between the representations corresponding to the same input, where the resultant partitioning of the latent space may not be suitable as an input to a simple linear classifier.

To address these issues, we can consider a second-order-statistics-based mutual information measure, referred to as log-determinant mutual information (LDMI) [15, 16]. The joint entropy measure corresponding to LDMI is the log-determinant of the diagonally perturbed covariance matrix. The zero perturbation case boils down to the Shannon differential entropy for Gaussian distributed vectors. Unlike SMI, which reflects general dependence between its arguments, LDMI reflects linear dependence and is computationally more efficient.

In this article, we propose a novel SSL approach, correlative information maximization (CorInfoMax), derived from the LDMI measure. The loss function for CorInfoMax is obtained through the first-order approximation of LDMI and restricting the linear dependence to the identity map. The resulting optimization objective is a Euclidean distance-based loss function regularized by the log-determinant of the latent vector covariance matrix. This regularization term encourages the spreading of the feature vectors and acts as a natural barrier against the dimensional degeneracy and collapse in the feature space. Our numerical experiments confirm that CorInfoMax learns effective representations that perform well in downstream tasks.

The following is the organization of the article: Section 2, provides a discussion of the relevant literature related to the proposed approach. We introduce the log-determinant entropy and mutual information measures by highlighting LDMI as a measure of linear dependence in Section 3. In the same section, we derive the objective function of CorInfoMax as a variation on the LDMI measure. Section 4 introduces the proposed correlative information maximization-based self-supervised learning method. Section 5 provides the numerical experiments illustrating the performance of the proposed approach. The appendix provided in the supplementary document contains the details related to these experiments. Finally, Section 6 presents the discussion and conclusions.

2 Related Work

2.1 Determinant Maximization for Unsupervised Learning

The determinant maximization criterion utilized in our framework has been used as an effective algorithmic tool in unsupervised matrix factorization methods such as nonnegative matrix factorization (NMF) [17, 18], simplex-structured matrix factorization (SSMF) [19], sparse component analysis (SCA) [20], bounded component analysis (BCA) [21] and polytopic matrix factorization (PMF) [22].

In the generative models of these frameworks, the input data is assumed to be linear transformations of some latent vectors. Furthermore, these latent vectors are assumed to be sufficiently scattered in their domain. Maximizing the determinant of the latent covariance matrix spreads the latent vector estimates to capture this presumed scattering of the generative model samples. Similarly, the log-determinant of the latent vector covariance matrix in the CorInfoMax objective causes the spreading of latent vectors in their ambient space and, therefore, avoids collapse.

According to [16], the determinant maximization criterion used in these matrix factorization frameworks is essentially equivalent to correlative information maximization between input data and its latent factors based on the LDMI objective, where the latent vectors are constrained to lie in their domain determined by the generative model.

In the CorInfoMax approach, we utilize determinant maximization for correlative information maximization among latent space vectors rather than between inputs and their latent space representations.

2.2 Handling Collapse for SSL

Collapse is a central concern for SSL, and we can categorize the existing approaches according to how they handle this issue. For example, contrastive methods, such as SimCLR [8] and MoCo [23], define an objective that pushes features for different inputs (negative samples) away from each other while keeping the representations corresponding to the same input (positive samples) close to each other. The number and the choice of the negative samples appear as critical factors for the performance and scalability of contrastive methods. Another category of SSL is the distillation methods, such as BYOL [4] and SimSiam [24], which avoid collapse through the asymmetry of alternative encoder branches and algorithmic tuning [25]. Other SSL approaches such as DeepCluster [26], SeLa [27] and SvAW [9] enforce a cluster structure in the feature space to prevent constant output. As a different line of algorithms, decorrelated batch normalization (DBN) [28] Barlow-Twins [5], Whitening MSE (W-MSE) [29] and VICReg [6], use feature decorrelation [30] as a means to avoid information collapse. The effective whitening mechanisms used in these methods aim for the isotropic spread of information inside the feature space which also prevents "dimensional collapse" illustrated in Figure 1.(b). Finally, [31] proposed graph representations for the embeddings of the positive samples and the corresponding spectral decomposition algorithm.

The proposed CorInfoMax approach is most related to the decorrelation-based methods discussed above, mainly due to correlation-based measures. However, unlike the decorrelation methods, CorInfoMax does not constrain latent vectors to be uncorrelated. Instead, it avoids covariance matrix degeneracy by using its log-determinant as a regularizer loss function. Furthermore, the information maximization principle is more direct and explicit for the CorInfoMax algorithm.

2.3 Information Maximization for Unsupervised Learning

The maximization of SMI has been proposed as an unsupervised learning mechanism in different but related contexts. As one of the earliest approaches, Linsker proposed maximum information

transfer from input data to its latent representation and showed that it is equivalent to maximizing the determinant of the output covariance under the Gaussian distribution assumption [32]. Around the same time frame, Becker & Hinton [33]put forward a representation learning approach based on the maximization of (an approximation of) the SMI between the alternative latent vectors obtained from the same image. The most well-known application is the Independent Component Analysis (ICA) Infomax algorithm [34] for separating independent sources from their linear combinations. The ICA-Infomax algorithm targets to maximize the mutual information between mixtures and source estimates while imposing statistical independence among outputs. The Deep Infomax approach [35] extends this idea to unsupervised feature learning by maximizing the mutual information between the input and output while matching a prior distribution for the representations.

It is essential to underline that the proposed method clearly distinguishes itself from the Deep Infomax in [35]: Our objective is not to maximize the mutual information between inputs and outputs of a deep network. Instead, we maximize the mutual information content of the alternative latent representations of the same input. From this point of view, our approach is closer to what is aimed at by [33]. However, we use a different (correlative) information measure, which is computationally more efficient and induces linear dependence among alternative latent representations of the same input, which may be more desirable considering the goal of generating features to be used by a linear classifier.

3 Log-Determinant Entropy and Mutual Information

The most natural and conventional means to measure uncertainty of a real-valued random vector $\mathbf{x}$ is to use the joint Shannon differential entropy defined by [10]

$$h(\mathbf{x}) \quad = \quad -\int\limits_{\mathrm{dom}(f_{\mathbf{x}})} \log(f_{\mathbf{x}}(x)) f_{\mathbf{x}}(x) \mathrm{d}x = -E(\log(f_{\mathbf{x}}(\mathbf{x}))),$$

where $f_{\mathbf{x}}$ is the joint probability density function (pdf) of the components of $\mathbf{x}$. The corresponding Shannon mutual information (SMI) between the random vectors $\mathbf{x}$ and $\mathbf{y}$ is defined as

$$I(\mathbf{x};\mathbf{y}) \quad = \quad h(\mathbf{x}) - h(\mathbf{x}\,|\,\mathbf{y}) \tag{1}$$

where $h(\mathbf{x}\,|\,\mathbf{y}) = -E(\log(f_{\mathbf{x}\,|\,\mathbf{y}}(\mathbf{x}\,|\,\mathbf{y})))$ is the conditional entropy. Shannon mutual information is a measure of dependence between its arguments.

For a given r-dimensional random vector $\mathbf{x}$ with the pdf $f_{\mathbf{x}}(\mathbf{x})$, log-determinant (LD) entropy is defined as [15, 16]

$$h_{LD}^{(\varepsilon)}(\mathbf{x}) = \frac{1}{2}\log\det(\mathbf{R}_{\mathbf{x}} + \varepsilon\mathbf{I}) + \frac{r}{2}\log(2\pi e), \tag{2}$$

where $\mathbf{R}_{\mathbf{x}}$ is the auto-covariance matrix of $\mathbf{x}$, and ε is a small nonnegative parameter defined for the diagonal perturbation on $\mathbf{R}_{\mathbf{x}}$. We note that $h_{LD}^{(\varepsilon)}(\mathbf{x})$ is equivalent to the Shannon entropy when $\mathbf{x}$ is a Gaussian vector with covariance $\mathbf{R}_{\mathbf{x}}$, and ε is equal to zero. However, we should underline that it is a standalone uncertainty measure solely based on the second-order statistics which reflects linear dependence.

The conditional LD-entropy defined as $h_{LD}^{(\varepsilon)}(\mathbf{x}\,|\,\mathbf{y}) = h_{LD}^{(\varepsilon)}(\begin{bmatrix} \mathbf{x}^T & \mathbf{y}^T \end{bmatrix}^T) - h_{LD}^{(\varepsilon)}(\mathbf{y})$, can be expressed in the following form:

$$h_{LD}^{(\varepsilon)}(\mathbf{x}|\mathbf{y}) = \log\det(\mathbf{R}_{\mathbf{x}} - \mathbf{R}_{\mathbf{xy}}(\mathbf{R}_{\mathbf{y}} + \varepsilon\mathbf{I})^{-1}\mathbf{R}_{\mathbf{xy}}^T + \varepsilon\mathbf{I}) + \frac{r}{2}\log(2\pi e), \tag{3}$$

where $\mathbf{R}_{\mathbf{xy}}$ is the cross-covariance matrix of $\mathbf{x}$ and $\mathbf{y}$. We note that the argument of the $\log\det$ function in (3) is the auto-covariance of the error in linearly estimating $\mathbf{x}$ from $\mathbf{y}$ with respect to the minimum mean square estimation (MMSE) criterion, for $\varepsilon \to 0$. Therefore, we can view $h_{LD}^{(\varepsilon)}(\mathbf{x}|\mathbf{y})$ as a measure of remaining uncertainty after linearly estimating $\mathbf{x}$ from $\mathbf{y}$ using the MMSE criterion.

The LD-mutual information (LDMI) measure is defined based on (2) and (3) as:

$$\begin{aligned} I_{LD}^{(\varepsilon)}(\mathbf{x};\mathbf{y}) \quad &= \quad h_{LD}^{(\varepsilon)}(\mathbf{x}) - h_{LD}^{(\varepsilon)}(\mathbf{x}\,|\,\mathbf{y}) \\ &= \quad \frac{1}{2}\log\det(\mathbf{R}_{\mathbf{x}} + \varepsilon\mathbf{I}) - \frac{1}{2}\log\det(\mathbf{R}_{\mathbf{x}} - \mathbf{R}_{\mathbf{xy}}(\mathbf{R}_{\mathbf{y}} + \varepsilon\mathbf{I})^{-1}\mathbf{R}_{\mathbf{xy}}^T + \varepsilon\mathbf{I}). \end{aligned} \tag{4}$$

By taking the average of (4) with its symmetric version obtained by exchanging $\mathbf{x}$ and $\mathbf{y}$, we can obtain an alternative but equivalent expression for the LDMI:

$$\begin{aligned} I_{LD}^{(\varepsilon)}(\mathbf{x};\mathbf{y}) &= \frac{1}{4}\log\det(\mathbf{R}_{\mathbf{x}}+\varepsilon\mathbf{I}) + \frac{1}{4}\log\det(\mathbf{R}_{\mathbf{y}}+\varepsilon\mathbf{I}) \\ &\quad -\frac{1}{4}\log\det(\mathbf{R}_{\mathbf{x}} - \mathbf{R}_{\mathbf{xy}}(\mathbf{R}_{\mathbf{y}}+\varepsilon\mathbf{I})^{-1}\mathbf{R}_{\mathbf{xy}}^T + \varepsilon\mathbf{I}) \\ &\quad -\frac{1}{4}\log\det(\mathbf{R}_{\mathbf{y}} - \mathbf{R}_{\mathbf{yx}}(\mathbf{R}_{\mathbf{x}}+\varepsilon\mathbf{I})^{-1}\mathbf{R}_{\mathbf{yx}}^T + \varepsilon\mathbf{I}). \end{aligned} \tag{5}$$

The following lemma asserts that $I_{LD}^{(\varepsilon)}(\mathbf{x},\mathbf{y})$ is an information measure reflecting the correlation, or "linear dependence" between two vectors [16]:

Lemma 1 *Let $\mathbf{x},\mathbf{y}$ be random vectors with the auto-covariance matrices $\mathbf{R}_{\mathbf{x}} > 0$ and $\mathbf{R}_{\mathbf{y}}$ respectively, and the cross-covariance matrix $\mathbf{R}_{\mathbf{xy}}$. Then*

- $I_{LD}^{(\varepsilon)}(\mathbf{x};\mathbf{y}) \geq 0$,
- $I_{LD}^{(\varepsilon)}(\mathbf{x};\mathbf{y}) = 0$ *if and only if* $\mathbf{R}_{\mathbf{xy}} = \mathbf{0}$, *i.e.,* $\mathbf{x}$ *and* $\mathbf{y}$ *are uncorrelated.*

There are three potential advantages of using LDMI in (4) over SMI in (1) for SSL:

i. The first apparent advantage of LDMI is that it is only based on second-order statistics. On the other hand, SMI is a statistic based on the joint PDFs of the argument vectors, and its accurate estimation has high sample complexity. In contrast, it is more practical to obtain (the estimates of) the auto-covariance and the cross-covariance matrices, as discussed in Section 4.

ii. The second advantage is related to the use of information maximization principle in connection with self-supervised training. The maximization of SMI of two vectors induces general, potentially nonlinear dependence between them, whereas the maximization of LDMI increases correlation, or linear dependence, between them. Therefore, LDMI-based information maximization is expected to lead to the organization of the feature space being more favorable for data-efficient and low-complexity linear or shallow supervised classifiers as targeted in self-supervised learning applications.

iii. The third advantage is related to the interpretability of the resulting LDMI-based maximization problems. The corresponding objective functions involve the log-determinant of the projector-space covariance, whose maximization clearly avoids feature collapse, a significant concern in non-contrastive self-supervised methods.

3.1 CorInfoMax as a Criterion Based on LDMI

This section derives the optimization objective for the proposed CorInfoMax framework as a variation on the LDMI measure. The primary motivation for obtaining an LDMI variant is to increase the similarity between the alternative latent representations of the same input by enforcing the linear identity transformation between them.

Let $\mathbf{z}^{(1)}$ and $\mathbf{z}^{(2)}$ represent alternative latent representations corresponding to the same input. One potential SSL approach would be to pursue direct maximization of $I_{LD}^{(\varepsilon)}(\mathbf{z}^{(1)};\mathbf{z}^{(2)})$ in (5). As discussed earlier, this would maximize the correlation between $\mathbf{z}^{(1)}$ and $\mathbf{z}^{(2)}$, therefore, inducing a linear dependence between them. For the SSL application, we would prefer the identity mapping to an arbitrary linear relationship, such that the alternative latent representations corresponding to the same input concentrate in the same neighborhood of the latent space. Therefore, we modify the LDMI expression in (5) to impose this constraint. For this purpose, we apply the first-order Taylor series approximation, $\log\det(\mathbf{C}+\mathbf{D}) \approx \log\det(\mathbf{C}) + \mathrm{Tr}(\mathbf{D}^T\mathbf{C}^{-1})$, on the third term in the right side of (5), which provides

$$\begin{aligned} &\log\det(\mathbf{R}_{\mathbf{z}^{(1)}} - \mathbf{R}_{\mathbf{z}^{(1)}\mathbf{z}^{(2)}}(\mathbf{R}_{\mathbf{z}^{(2)}}+\varepsilon\mathbf{I})^{-1}\mathbf{R}_{\mathbf{z}^{(1)}\mathbf{z}^{(2)}}^T + \varepsilon\mathbf{I}) \\ &\approx \frac{1}{\varepsilon}\mathrm{Tr}(\mathbf{R}_{\mathbf{z}^{(1)}} - \mathbf{R}_{\mathbf{z}^{(1)}\mathbf{z}^{(2)}}\mathbf{R}_{\mathbf{z}^{(2)}}^{-1}\mathbf{R}_{\mathbf{z}^{(1)}\mathbf{z}^{(2)}}^T) + \log\det(\varepsilon\mathbf{I}) \\ &= \frac{1}{\varepsilon}\min_{\mathbf{A}_1,\mathbf{b}_1} E(\|\mathbf{z}^{(1)} - (\mathbf{A}_1\mathbf{z}^{(2)} + \mathbf{b}_1)\|_2^2) + P\log(\varepsilon). \end{aligned} \tag{6}$$

Therefore, the expression in (6) corresponds to the mean square error of the best linear(affine) MMSE estimator of $\mathbf{z}^{(1)}$ from $\mathbf{z}^{(2)}$, multiplied by ε^{-1}. Similarly, if we apply the same approximation to the fourth term on the right side of (5), we obtain

$$\begin{aligned}&\log\det(\mathbf{R}_{\mathbf{z}^{(2)}} - \mathbf{R}^T_{\mathbf{z}^{(1)}\mathbf{z}^{(2)}}(\mathbf{R}_{\mathbf{z}^{(1)}} + \varepsilon\mathbf{I})^{-1}\mathbf{R}_{\mathbf{z}^{(1)}\mathbf{z}^{(2)}} + \varepsilon\mathbf{I})\\ &\approx \frac{1}{\varepsilon}\min_{\mathbf{A}_2,\mathbf{b}_2} E(\|\mathbf{z}^{(2)} - (\mathbf{A}_2\mathbf{z}^{(1)} + \mathbf{b}_2)\|_2^2) + P\log(\varepsilon). \qquad (7)\end{aligned}$$

In order to induce the identity mapping between $\mathbf{z}^{(1)}$ and $\mathbf{z}^{(2)}$, we constrain $\mathbf{A}_1 = \mathbf{A}_2 = \mathbf{I}$, and $\mathbf{b}_1 = \mathbf{b}_2 = \mathbf{0}$, which transforms both (6) and ((7) into $\varepsilon^{-1}E(\|\mathbf{z}^{(1)} - \mathbf{z}^{(2)}\|_2^2) + \text{const}$. Therefore, scaling the expression in (5) by 4 and using this modification, we obtain

$$J(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}) = \log\det(\mathbf{R}_{\mathbf{z}^{(1)}} + \varepsilon\mathbf{I}) + \log\det(\mathbf{R}_{\mathbf{z}^{(2)}} + \varepsilon\mathbf{I}) - 2\varepsilon^{-1}E(\|\mathbf{z}^{(1)} - \mathbf{z}^{(2)}\|_2^2), \qquad (8)$$

where we ignored the constant terms. We refer to (8) as the stochastic CorInfoMax objective function. In Section 4, we propose a self supervised learning method based on the optimization of this objective function.

4 LDMI based Self-Supervised Learning

This section introduces the proposed correlative information maximization (CorInfoMax) approach for SSL. We start by describing the presumed setting for the pretext task, which is matching the latent representations of different augmentations of the same input in Section 4.1. Section 4.2 is the main section where we propose the correlative information maximization algorithm for SSL. Finally, we discuss the implementation complexity of the proposed approach in Section 4.3.

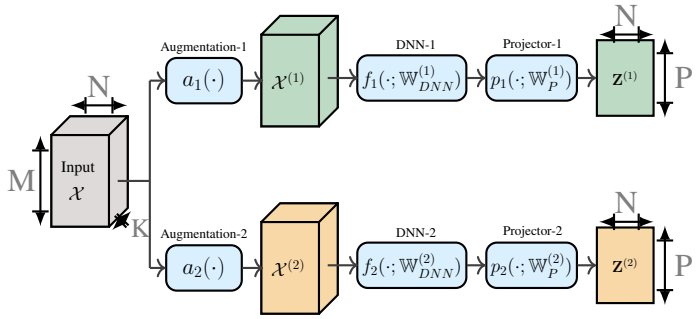

Figure 2: Self-supervised learning setup: we consider two parallel encoder branches corresponding to two different augmentations of the same input X. Augmented views are fed into Siamese networks f followed by projector g, basically a 3-layer MLP. N stands for batch size, $M \times K$ for image dimension, P for feature dimension of projector output.

4.1 Self-Supervised Learning Setting

We start by describing the presumed self-supervised learning setup, which is illustrated in Figure 2:

- *The input* is a sequence of tensors $\{\mathcal{X}[l] \in \mathbb{R}^{M\times K\times N}, l \in \mathbb{Z}\}$, where each index sample corresponds to a batch of N images with dimensions $M \times K$.
- As discussed in Section 1, the pretext task that we consider is matching the latent representations of the two augmentations of the same input. Therefore, we first apply *the augmentation functions* $a_1(\cdot)$ and $a_2(\cdot)$ to obtain the transformed versions of the input sequence, i.e., $\{\mathcal{X}^{(q)}[l] = a_q(\mathcal{X}[l]) \in \mathbb{R}^{M\times K\times N}, l \in \mathbb{Z}\}, \quad q \in \{1,2\}$. We will illustrate particular choices of augmentations in the examples provided in Section 5. We note that the proposed setting can be potentially extended to multi-modal schemes where the augmentations are defined for multiple modalities of the same input, such as visual and sound.
- We represent *Siamese networks* to be trained by self-supervised learning with the mappings $f_q(\mathcal{U};\mathbb{W}^{(q)}_{DNN})$ for $q = 1,2$, where $\mathcal{U} \in \mathbb{R}^{M\times K\times N}$ is the input, $\mathbb{W}^{(q)}_{DNN}$ represents the trainable parameters of the qth network.

- The outputs of the DNN for different augmented inputs are represented by $\mathbf{Y}^{(q)}[l] = f_q(\mathbb{X}^{(q)}[l]; \mathbb{W}_{DNN}^{(q)}) \in \mathbb{R}^{F\times N}, \quad q \in \{1,2\}, l \in \mathbb{Z}$, where F is the output feature dimension.
- To be used in self-supervised training, we project the outputs of the enconders using *projector networks* $p_q(\mathbf{Y}; \mathbb{W}_P^{(q)})$, $q = 1, 2$, where $\mathbf{Y}$ is its input, and $\mathbb{W}_P^{(q)}$ is the set of trainable parameters of the qth projector.
- The outputs of the projector network for different augmentations are represented by $\mathbf{Z}^{(q)}[l] = p_q(\mathbf{Y}^{(q)}[l]; \mathbb{W}_P^{(q)}) \in \mathbb{R}^{P\times N}, \quad q \in \{1,2\}, l \in \mathbb{Z}$, where P is the projector head dimension.

For the numerical experiments in the article, we consider the weight-sharing setup where both encoders use the same network weights. We provide more details in Appendix A.

4.2 Correlative Mutual Information Maximization

We define the CorInfoMax approach for SSL through the optimization problem

$$\underset{\mathbb{W}_{DNN},\, \mathbb{W}_P}{\text{maximize}} \quad \hat{J}(\mathbf{z}^{(1)}; \mathbf{z}^{(2)})[l] \tag{9a}$$

where $\hat{J}^{(\varepsilon)}$ is the sample based estimate of (8) at the lth-batch which can be written as

$$\hat{J}(\mathbf{z}^{(1)}, \mathbf{z}^{(2)})[l] = \log\det(\hat{\mathbf{R}}_{\mathbf{z}^{(1)}}[l] + \varepsilon\mathbf{I}) + \log\det(\hat{\mathbf{R}}_{\mathbf{z}^{(2)}}[l] + \varepsilon\mathbf{I}) - \frac{2}{\varepsilon N}\|\mathbf{Z}^{(1)}[l] - \mathbf{Z}^{(2)}[l]\|_F^2, \tag{10}$$

where the rightmost term stands for the sample based estimate of the term $2\varepsilon^{-1}E(\|\mathbf{z}^{(1)} - \mathbf{z}^{(2)}\|_2^2)$ in (8), and $\hat{\mathbf{R}}_{\mathbf{z}^{(1)}}[l]$, $\hat{\mathbf{R}}_{\mathbf{z}^{(2)}}[l]$ and $\hat{\mathbf{R}}_{\mathbf{z}^{(1)}\mathbf{z}^{(2)}}[l]$ are the auto-covariance matrix and cross-covariance matrix estimates for the projector heads at the lth batch. If the batch size is large enough, these estimates can only be obtained from the current batch samples. However, due to hardware limitations and considering the fact that intermediate batch sizes offer better accuracy [13, 14], sufficiently-large batch sizes may not be possible to obtain reliable covariance estimates. Therefore, we adopt recursive covariance matrix estimation across batches [36]. The corresponding covariance update expressions take the form

$$\hat{\mathbf{R}}_{\mathbf{z}^{(q)}}[l] \quad = \quad \lambda\hat{\mathbf{R}}_{\mathbf{z}^{(q)}}[l-1] + (1-\lambda)\frac{1}{N}\tilde{\mathbf{Z}}^{(q)}[l]\tilde{\mathbf{Z}}^{(q)}[l]^T . q \in \{1,2\}, \tag{11}$$

where $0 \leq \lambda < 1$ is the forgetting factor, and $\tilde{\mathbf{Z}}^{(q)}[l]$ represents the batch of mean-centralized projector outputs defined as $\tilde{\mathbf{Z}}^{(q)}[l] = \mathbf{Z}^{(q)}[l] - \boldsymbol{\mu}^{(q)}[l]\mathbf{1}_N^T$, where $\{\boldsymbol{\mu}^{(q)}[l], q \in \{1,2\}\}$ represents the mean estimates for the projector outputs, updated by $\boldsymbol{\mu}^{(q)}[l] = \lambda\boldsymbol{\mu}^{(q)}[l-1] + (1-\lambda)\frac{1}{N}\mathbf{Z}^{(q)}\mathbf{1}_N$, for $q \in \{1,2\}$. We can write the update expression for the cross-covariance matrix as $\hat{\mathbf{R}}_{\mathbf{z}^{(1)}\mathbf{z}^{(2)}}[l] = \lambda\hat{\mathbf{R}}_{\mathbf{z}^{(1)}\mathbf{z}^{(2)}}[l-1] + (1-\lambda)\frac{1}{N}\tilde{\mathbf{Z}}^{(1)}[l]\tilde{\mathbf{Z}}^{(2)}[l]^T$.

It is informative to inspect the terms in the objective function $\hat{J}(\mathbf{z}^{(1)}, \mathbf{z}^{(2)})$ in (10):

- The minimization of $\frac{2\varepsilon^{-1}}{N}\|\mathbf{Z}^{(1)}[l] - \mathbf{Z}^{(2)}[l]\|_F^2$, acts as a force to pull the representations of alternative augmentations towards each other, which we refer to as the *attraction factor*.
- The maximization of $\log\det(\hat{\mathbf{R}}_{\mathbf{z}^{(1)}}[l] + \varepsilon\mathbf{I})$ acts as a dispersion force causing non-degenerate (for $\varepsilon \approx 0$) expansion of projection vectors in the P-dimensional space, which we informally refer to as *big-bang factor*. Therefore, the covariance determinant acts as a regularization or barrier function avoiding collapse in the latent space. Consequently, it provides a convincing replacement for negative samples in contrastive methods to prevent feature collapse.

Figure 1,(c) illustrates the learning dynamics of the CorInfoMax approach based on (10) with a toy picture. In this figure, the ellipsoid is a representative surface for the level set of the quadratic function $(\mathbf{z} - \boldsymbol{\mu}^{(q)})^T\mathbf{R}_{\mathbf{z}^{(q)}}^{-1}(\mathbf{z} - \boldsymbol{\mu}^{(q)})$, which reflects the spreading pattern of the latent vectors around their mean. The black arrows represent the gradient of the $\log\det(\mathbf{R}_{\mathbf{z}^{(q)}})$ regularization factor, which acts as a force to push latent vectors away from the center of the ellipsoid ($\boldsymbol{\mu}^{(q)}$) and therefore, corresponds to the expansion force of the "big-bang" factor. In the same figure, the white arrows correspond to the gradients corresponding to the attraction factor, i.e., the Euclidian distance between two positive pairs. In summary, we can view the learning dynamics of CorInfomax SSL as an expansion in the latent space while the representations of positive samples are attracted to each other.

4.3 Computational complexity

The main difference between our and related SSL methods is the extra log determinant computation. The log determinant is typically computed using a decomposition method such as LU decomposition and its gradient requires a matrix inversion. Both determinant and inversion have the same complexity as matrix multiplication [37], i.e. $O(n^\alpha)$ with $2 < \alpha \leq 3$ for multiplying two $n \times n$ matrices. In our experiments with GPUs, we observe an almost flat runtime cost up to $n = 1024$, which implies that the overhead due to the $\log\det$ cost is insignificant in practice. We find the extra cost of log determinant and its gradient negligible compared to the rest of the model computation, whose runtime is dominated by the encoder, as confirmed by the runtime experiments reported in Appendix D.

5 Experiments

5.1 Implementation Details

Datasets: We perform experiments on CIFAR-10 and CIFAR-100 [38], Tiny ImageNet [39], and ImageNet-100 datasets. For ImageNet-100, we use the same subset of ImageNet [40] as related work [41-43].

Training Procedure: The experiments consist of two consecutive stages: pretraining and linear evaluation. We first perform the unsupervised pretraining of the encoder network f by applying the proposed CorInfoMax method described in Section 4.2 on the training dataset. After completing pretraining, we perform the linear evaluation, a standardized protocol to evaluate the quality of the learned representations [4, 8, 44]. For the linear evaluation stage, we first perform a supervised training of the linear classifier using the representations obtained from the encoder network f with frozen coefficients on the same training dataset. Then we obtain the test accuracy results for the trained linear classifier based on the validation dataset.

Computational Resources: For ImageNet-100, we pretrain our model on 8 A100 Cloud GPUs, with a batch size of 1024. We use a batch size of 512 for CIFAR datasets, while we use a batch size of 1024 for Tiny-ImageNet; both are trained using a single A100 Cloud GPU. Linear evaluations are performed using the same type and amount of computational resources, with a batch size of 256.

5.1.1 Input Augmentations

During the *pretraining stage*, two augmented versions of each input image are generated as shown in Figure 2. During this process, each image is cropped with random size, resized to the original resolution, followed by random applications of horizontal mirroring, color jittering, grayscale conversion, Gaussian blurring, and solarization. Since we use the same augmentation parameters as BYOL [4] and VicReg [6]: each augmentation branch uses the same probability values for these randomized operations except Gaussian blurring and solarization, which use different probabilities.

During the training period of the *linear evaluation* stage, a single augmentation of each input image is produced by random cropping and resizing followed by a random horizontal flip. For the test period of linear evaluation, we use resize and center crop augmentations, similar to [5, 6]. We provide more details about the augmentations in Appendix B.

5.1.2 Network Architecture

Encoder Network: For CIFAR datasets, we use a modified form of ResNet-18 architecture [45] similar to [8, 24, 31]. We use the standard ResNet-18 and ResNet-50 architectures ([45]) for ImageNet-100 and Tiny ImageNet. In both cases, the last fully connected layer is removed. Therefore, the encoder output size is 512 and 2048 for ResNet-18 and ResNet-50, respectively. The encoder network f shares weights between augmented branches (see Appendix A).

Projector Network: The output of the encoder network is fed into the projector network, as in Figure 2. The projector network p is a 3-layer MLP, with ReLU activation functions for the hidden layers and linear activation functions for the output layer. The projector dimensions are 2048-2048-64 for CIFAR datasets, and 4096-4096-128 for Tiny-ImageNet and ImageNet-100. Contrary to the common practice [5, 6], we are not using batch normalization [46] in the projection network p. Finally, we perform L_2-normalization over the projector output.

Linear Classifier: During the linear evaluation period, we employ a standard linear classifier whose input is the weight-frozen encoder network's output.

5.1.3 Optimization

For pretraining, our model is trained 1000 epochs with a batch size of 512 for CIFAR datasets, and 800 epochs with a batch size of 1024 for Tiny-ImageNet. In ImageNet-100 experiments, we run 400 epochs for ResNet-18 and run 200 epochs for ResNet-50 with a batch size of 1024 for both.

For optimization, we use the SGD optimizer with the momentum of 0.9 and the weight decay of $1e-4$. The learning rate is adjusted by batch size using the linear scaling rule [47], where the initial learning rate is 0.5 for batch size of 512 and follows cosine decay with a linear warmup schedule.

We use the objective function in (10) as our objective function, where we replace $\frac{2\varepsilon^{-1}}{N}$ with α, the attraction coefficient, which we consider as a separate hyper-parameter. In our experiments, we chose $\alpha = 500$, the forgetting factor $\lambda = 0.01$, and the diagonal perturbation $\varepsilon = 1\text{e}{-8}$. Details with coefficients of our loss function are provided in Appendix C.

For linear evaluation, the linear classifier is trained for 100 epochs with a batch size of 256 for all the training datasets. We used the SGD optimizer with a momentum value of 0.9 and no weight decay. The learning rate follows a cosine decay schedule with initial and minimum learning rates of 0.2 and $2\text{e}{-3}$, respectively.

5.2 Results

We evaluate the learned representations from the CorInfoMax pretraining by following the linear evaluation protocol explained in Section 5.1. The results in Table 1 show that CorInfoMax achieves state-of-the-art performances in linear classification after pretraining. Due to the limited size of the validation sets (5-10K), differences less than $\approx 0.5\%$ are not statistically significant.

Table 1: Top-1 accuracies (%) under linear evaluation on CIFAR-10, CIFAR-100, Tiny-ImageNet and ImageNet-100 datasets. Results for SimCLR [8], SimSiam [24], and Spectral Constrastive [31] are reported from [31]. Results for BYOL[4] and W-MSE[29] are reported from [29]. In addition, results for Imagenet-100 in ResNet-50 for BYOL [4] and SimSiam [24] are reported from [43, 48], respectively. The solo-learn [49] row consists of the best results among the 12 methods reimplemented and hyperparameter tuned in that work: MoCo-V2 [50] for CIFAR-10 and Barlow-Twins [5] for CIFAR-100 and ImageNet-100. We bold all top results that are statistically indistinguishable.

Method	CIFAR-10	CIFAR-100	Tiny-ImageNet	ImageNet-100	
	ResNet-18	ResNet-18	ResNet-50	ResNet-18	ResNet-50
SimCLR	91.80	66.83	48.12	-	-
SimSiam	91.40	65.87	46.76	-	81.6
Spectral Con.	92.07	66.18	49.86	-	-
BYOL	91.73	66.60	-	-	78.76
W-MSE	91.55	66.10	-	-	-
solo-learn	**92.94**	70.90	-	**80.38**	-
CorInfoMax	**92.86**	**71.41**	**53.72**	**80.48**	**82.64**

6 Discussion and Conclusion

We proposed a novel SSL framework based on the correlative information maximization, CorInfoMax, which provides a natural solution for obtaining organized representations that are scattered in the latent space and avoiding collapse. Our experiments confirm the competitive performance in downstream tasks provided by the CorInfoMax approach.

We can list the sensitivity of the performance to the hyperparameter and augmentation selections as one of the main limitations, which is common to SSL approaches in general. As confirmed by the experiments, the additional cost of the $\log\det$ regularizer function has no significant impact on the speed of pretraining.

References

[1] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167, 2008.

[2] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430, 2015.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[4] 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.

[5] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR, 2021.

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

[7] Linus Ericsson, Henry Gouk, and Timothy M Hospedales. How well do self-supervised models transfer? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5414–5423, 2021.

[8] 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.

[9] 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.

[10] MTCAJ Thomas and A Thomas Joy. *Elements of information theory*. Wiley-Interscience, 2006.

[11] Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *International Conference on Machine Learning*, pages 5171–5180. PMLR, 2019.

[12] Michael Tschannen, Josip Djolonga, Paul K Rubenstein, Sylvain Gelly, and Mario Lucic. On mutual information maximization for representation learning. In *International Conference on Learning Representations*, 2019.

[13] Sam McCandlish, Jared Kaplan, Dario Amodei, and OpenAI Dota Team. An empirical model of large-batch training. *arXiv preprint arXiv:1812.06162*, 2018.

[14] Dominic Masters and Carlo Luschi. Revisiting small batch training for deep neural networks. *arXiv preprint arXiv:1804.07612*, 2018.

[15] Zhanghao Zhouyin and Ding Liu. Understanding neural networks with logarithm determinant entropy estimator. *arXiv preprint arXiv:1401.3420*, 2021.

[16] Alper T Erdogan. An information maximization based blind source separation approach for dependent and independent sources. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022.

[17] Xiao Fu, Kejun Huang, Nicholas D Sidiropoulos, and Wing-Kin Ma. Nonnegative matrix factorization for signal and data analytics: Identifiability, algorithms, and applications. *IEEE Signal Process. Mag.*, 36(2):59–80, March 2019.

[18] Xiao Fu, Kejun Huang, Bo Yang, Wing-Kin Ma, and Nicholas D Sidiropoulos. Robust volume minimization-based matrix factorization for remote sensing and document clustering. *IEEE Transactions on Signal Processing*, 64(23):6254–6268, August 2016.

[19] Tsung-Han Chan, Wing-Kin Ma, ArulMurugan Ambikapathi, and Chong-Yung Chi. A simplex volume maximization framework for hyperspectral endmember extraction. *IEEE Transactions on Geoscience and Remote Sensing*, 49(11):4177–4193, May 2011.

[20] Eren Babatas and Alper T Erdogan. An algorithmic framework for sparse bounded component analysis. *IEEE Transactions on Signal Processing*, 66(19):5194–5205, August 2018.

[21] Huseyin A Inan and Alper T Erdogan. A convolutive bounded component analysis framework for potentially nonstationary independent and/or dependent sources. *IEEE Transactions on Signal Processing*, 63(1):18–30, November 2014.

[22] Gokcan Tatli and Alper T. Erdogan. Polytopic matrix factorization: Determinant maximization based criterion and identifiability. *IEEE Transactions on Signal Processing*, 69:5431–5447, 2021. doi: 10.1109/TSP.2021.3112918.

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

[24] 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.

[25] Yuandong Tian, Xinlei Chen, and Surya Ganguli. Understanding self-supervised learning dynamics without contrastive pairs. In *International Conference on Machine Learning*, pages 10268–10278. PMLR, 2021.

[26] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pages 132–149, 2018.

[27] Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. *arXiv preprint arXiv:1911.05371*, 2019.

[28] Lei Huang, Dawei Yang, Bo Lang, and Jia Deng. Decorrelated batch normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 791–800, 2018.

[29] Aleksandr Ermolov, Aliaksandr Siarohin, Enver Sangineto, and Nicu Sebe. Whitening for self-supervised representation learning. In *International Conference on Machine Learning*, pages 3015–3024. PMLR, 2021.

[30] Tianyu Hua, Wenxiao Wang, Zihui Xue, Sucheng Ren, Yue Wang, and Hang Zhao. On feature decorrelation in self-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9598–9608, 2021.

[31] Jeff Z HaoChen, Colin Wei, Adrien Gaidon, and Tengyu Ma. Provable guarantees for self-supervised deep learning with spectral contrastive loss. *Advances in Neural Information Processing Systems*, 34, 2021.

[32] Ralph Linsker. Self-organization in a perceptual network. *Computer*, 21(3):105–117, 1988.

[33] Suzanna Becker and Geoffrey E Hinton. Self-organizing neural network that discovers surfaces in random-dot stereograms. *Nature*, 355(6356):161–163, 1992.

[34] Anthony J Bell and Terrence J Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural computation*, 7(6):1129–1159, 1995.

[35] 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.

[36] Justinian Rosca, Deniz Erdogmus, José C Príncipe, and Simon Haykin. *Independent Component Analysis and Blind Signal Separation: 6th International Conference, ICA 2006, Charleston, SC, USA, March 5-8, 2006, Proceedings*, volume 3889. Springer, 2006.

[37] James R Bunch and John E Hopcroft. Triangular factorization and inversion by fast matrix multiplication. *Mathematics of Computation*, 28(125):231–236, 1974.

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

[39] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

[40] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

[41] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019.

[42] Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. Hard negative mixing for contrastive learning. *Advances in Neural Information Processing Systems*, 33:21798–21809, 2020.

[43] Songwei Ge, Shlok Mishra, Chun-Liang Li, Haohan Wang, and David Jacobs. Robust contrastive learning using negative samples with diminished semantics. *Advances in Neural Information Processing Systems*, 34, 2021.

[44] Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. Revisiting self-supervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1920–1929, 2019.

[45] 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.

[46] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[47] 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.

[48] Hankook Lee, Kibok Lee, Kimin Lee, Honglak Lee, and Jinwoo Shin. Improving transferability of representations via augmentation-aware self-supervision. *Advances in Neural Information Processing Systems*, 34, 2021.

[49] Victor Guilherme Turrisi da Costa, Enrico Fini, Moin Nabi, Nicu Sebe, and Elisa Ricci. Solo-learn: A library of self-supervised methods for visual representation learning. *Journal of Machine Learning Research*, 23(56):1–6, 2022.

[50] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] As claimed in the abstract and introduction, in the article we provide a novel self supervised learning framework based on the information maximization principle and provide numerical experiment results reflecting its performance.

(b) Did you describe the limitations of your work? [Yes] We stated sensitivity to hyperparameter selections as a limitation in Section 6.

(c) Did you discuss any potential negative societal impacts of your work? [N/A] We believe that our work does not have any potential negative societal impacts.

(d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] We have read the ethics review guidelines and our paper conforms to them.

2. If you are including theoretical results...

(a) Did you state the full set of assumptions of all theoretical results? [Yes] Theoretical discussions in Section 3 and Section 4 clearly provide the mathematical settings used in our derivations.

(b) Did you include complete proofs of all theoretical results? [Yes] We show the details of the derivations of our proposed objective in Section 3.

3. If you ran experiments...

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We include the Python codes of our experiments in supplementary materials and the details of the experiment results in Section 5 and in Appendix.

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] We provided all the details about training, its phases, hyperparameter selections in Section 5 and in Appendix.

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] In Appendix, we provide information on the confidence interval of our experimental results.

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We reported the details about the computation resources and computation times in Section 5 and in Appendix.

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

(a) If your work uses existing assets, did you cite the creators? [Yes] We cite references for all the data sets, experiment results and network models in Section 5.

(b) Did you mention the license of the assets? [N/A]

(c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We include our code as supplementary material.

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]