Supplementary Material of “Explicit Mutual Information Maximization for Self-Supervised Learning”

I. Related Work

While there exists a large number of SSL methods developed in the last years, here we mainly review recent contrastive-based methods closely related to ours. Besides, while our method is based on MI optimization, we particularly review the works related to MI.

A. SIAMESE NETWORKS BASED SELF-SUPERVISED REPRESENTATION LEARNING

Siamese networks [1] have become a prevalent structure in recent SSL models and achieved great performance [2, 3, 4, 5, 6]. Siamese structure-based methods utilize two weight-sharing network encoders to process distinct views of the same input image, which facilitates comparing and contrasting entities. Typically, these models are designed to maximize the similarity between two different augmentations of the same image, while employing various regulations to avoid the collapse problem of converging to a trivial constant solution. Contrastive learning is an effective approach to avoid undesired trivial solutions. Techniques such as SimCLR, MoCo, PIRL leverage this approach by contrasting between positive and negative pairs, e.g., pulling positive pairs closer while pushing negative pairs farther apart [2, 3, 7, 8]. Another approach SwAV [6] utilizes online clustering to prevent trivial solutions, which clusters features to prototypes while enforcing consistency between cluster assignments of different views of the same image. Without using negative pairs for explicit contrasting, asymmetric structure and momentum encoder have been considered for preventing collapsing [3, 4, 9, 10, 11, 12]. For example, BYOL uses a Siamese network with one branch being a momentum encoder and directly predicts the output representation of one branch from another [4]. Then, it has been recognized in [9] that the momentum encoder in BYOL is unnecessary for preventing collapsing, rather a stop-gradient operation is crucial for avoiding collapsing. Furthermore, the Barlow Twins method [5] shows that, without using negative pairs and asymmetry structure, the collapse problem can be naturally avoided by feature-wise contrastive learning. It maximizes the similarity between the embeddings of distorted versions of the same sample, while minimizing the redundancy between the features of the embeddings. Moreover, VICReg [13] explicitly avoids the collapse problem using a regularization term on feature variance and combining it with redundancy reduction and covariance regularization to form a variance-invariance-covariance regularization formulation.

B. MUTUAL INFORMATION MAXIMIZATION FOR SELF-SUPERVISED LEARNING

MI is a fundamental quantity for measuring the dependence between random variables based on Shannon entropy. While calculating MI has traditionally been challenging, neural network-based methods have been recently developed to estimate MI for high-dimensional random variables [14, 15]. For SSL, many early methods [2, 3, 16, 17] use a contrastive loss function called InfoNCE or its variants. These methods commonly employ an NCE estimation of MI based on discriminating between positive and negative pairs [15]. These methods are linked to MI maximization as the NCE loss is a lower bound of MI [18]. As the NCE estimator of MI is low-variance but high-bias, a large batch size is required at test time for accurate MI estimation when the MI is large. Moreover, in [19], the minimal coding length in lossy coding is used as a surrogate to construct a maximum entropy coding objective for SSL. Additionally, the recent work [20] has shown that the VICReg method [13] in fact implements an approximation of MI maximization criterion. While these methods can be viewed as approximate implementations of the MI maximization criterion, we consider explicit MI maximization for SSL.

II. Proof of Theorem 1

We first recall the result on the invariance property of mutual information. Specifically, if $Y = F(Z)$ and $Y' = G(Z')$ are homeomorphisms, then $I(Z; Z') = I(Y; Y')$ [21]. Denote $\tilde{Y} = [Y^T, Y'^T]^T$, if $Y = F(Z)$ and $Y' = G(Z')$ are Gaussian distributed, i.e. $Y \sim \mathcal{N}(Y; \mu_Y, C_{YY})$ and $Y' \sim \mathcal{N}(Y'; \mu_{Y'}, C_{Y'Y'})$, we have $\tilde{Y} \sim \mathcal{N}(\tilde{Y}; \mu_{\tilde{Y}}, C_{\tilde{Y}\tilde{Y}})$ with

$$C_{\tilde{Y}\tilde{Y}} = \left[\begin{array}{cc} C_{YY} & C_{YY'} \\ C_{Y'Y} & C_{Y'Y'} \end{array} \right].$$

Then, the mutual information $I(Y;Y') = H(Y) + H(Y') - H(Y,Y')$ is given by

$$I(Y;Y') = \frac{1}{2}\log\frac{\det(C_{YY})\det(C_{Y'Y'})}{\det(C_{\tilde{Y}\tilde{Y}})},$$

where $H(Y)$ and $H(Y')$ are the marginal entropy of Y and Y', respectively, $H(Y,Y')$ is the joint entropy of Y and Y'. This together with the invariance property of mutual information, i.e. $I(Z;Z') = I(Y;Y')$ under homeomorphisms condition, results in Theorem 1.

III. Proof of Theorem 2

Theorem 1 implies that we can compute the MI only based on second-order statistics even if the distributions of Z and Z' are not Gaussian. We investigate the MI under the generalized Gaussian distribution (GGD) as defined in (3) of the main paper. The GGD offers a flexible parametric form that can adapt to a wide range of distributions by varying the shape parameter β in (3), from super-Gaussian when $\beta < 1$ to sub-Gaussian when $\beta > 1$, including the Gamma, Laplacian and Gaussian distributions as special cases. Figure 1 provides an illustration of univariate GGD with different values.

Let $\tilde{Z} = \left[Z^T, Z'^T\right]^T \in \mathbb{R}^{2d}$, and from (3) in main paper, the joint distribution $p_{Z,Z'}(z,z')$ is $\tilde{Z} \sim \mathcal{GN}\left(\tilde{Z}; \mu_{\tilde{Z}}, \Sigma_{\tilde{Z}\tilde{Z}}, \beta\right)$, where $\mu_{\tilde{Z}}$ is the mean, and $\Sigma_{\tilde{Z}\tilde{Z}}$ is the dispersion matrix. The MI between Z and Z' is given by

$$\begin{aligned} & I\left(Z, Z'\right) \\ & = \iint p_{Z,Z'}\left(z, z'\right) \log \frac{p_{Z,Z'}\left(z, z'\right)}{p_Z(z) p_{Z'}\left(z'\right)} dz dz' \\ & = E\left[\log p_{Z,Z'}\left(z, z'\right)\right] - E\left[\log p_Z(z)\right] - E\left[\log p_{Z'}\left(z'\right)\right]. \end{aligned} \tag{12}$$

Then, it follows that

$$\begin{aligned} & E\left[\log p_{Z,Z'}\left(z, z'\right)\right] \\ & = \log \frac{\Phi(\beta, 2n)}{\left[\det\left(\Sigma_{\tilde{Z}\tilde{Z}}\right)\right]^{1/2}} - \frac{1}{2} E\left\{\left[\left(\tilde{Z} - \mu_{\tilde{Z}}\right)^T \Sigma_{\tilde{Z}\tilde{Z}}^{-1}\left(\tilde{Z} - \mu_{\tilde{Z}}\right)\right]^\beta\right\}, \end{aligned}$$

where the expectation over the parameter space $\mathbb{R}^n$ of a function $\varphi\left((\tilde{Z} - \mu_{\tilde{Z}})^T \Sigma_{\tilde{Z}\tilde{Z}}^{-1}\left(\tilde{Z} - \mu_{\tilde{Z}}\right)\right) = \varphi\left(\bar{Z}^T \bar{Z}\right) \equiv \varphi(w)$ $\left(\text{with } w > 0 \text{ for } \tilde{Z} - \mu_{\tilde{Z}} \neq 0\right)$ is essentially type 1 Dirichlet integral, which can be converted into integral over $\mathbb{R}^+$ [22]. Specifically, let $\varphi(w) = w^\beta$, then

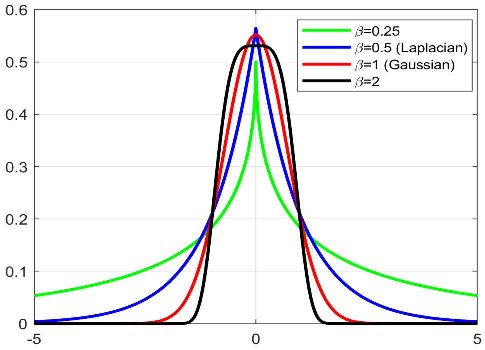

Fig. 1. Univariate generalized Gaussian distribution with different values of the shape parameter.

$$
\begin{aligned}
&E\left\{\left[\left(\tilde{Z}-\mu_{\tilde{Z}}\right)^{T}\Sigma_{\tilde{Z}\tilde{Z}}^{-1}\left(\tilde{Z}-\mu_{\tilde{Z}}\right)\right]^{\beta}\right\}\\
&=\frac{\Phi(\beta,2n)}{\left[\det\left(\Sigma_{\tilde{Z}\tilde{Z}}\right)\right]^{1/2}}\int_{\mathbb{R}^{2n}}\left[\left(\tilde{Z}-\mu_{\tilde{Z}}\right)^{T}\Sigma_{\tilde{Z}\tilde{Z}}^{-1}\left(\tilde{Z}-\mu_{\tilde{Z}}\right)\right]^{\beta}\\
&\qquad\times\exp\left(-\frac{1}{2}\left[\left(\tilde{Z}-\mu_{\tilde{Z}}\right)^{T}\Sigma_{\tilde{Z}\tilde{Z}}^{-1}\left(\tilde{Z}-\mu_{\tilde{Z}}\right)\right]^{\beta}\right)d\tilde{z}\\
&\overset{(a)}{=}\frac{\beta}{2^{n/\beta}\Gamma(n/\beta)}\int_{\mathbb{R}^{+}}\varphi(w)w^{n-1}\exp\left(-\frac{1}{2}w^{\beta}\right)dw \qquad ,\\
&=\frac{\beta}{2^{n/\beta}\Gamma(n/\beta)}\frac{2^{2+n/\beta}\Gamma\left((\beta+n)/\beta\right)}{2\beta}\\
&=\frac{2\Gamma((\beta+n)/\beta)}{\Gamma(n/\beta)}\\
&=\frac{2n}{\beta}
\end{aligned}
$$

where (a) is due to the fact that the density function of the positive variable $w=\left(\tilde{Z}-\mu_{\tilde{Z}}\right)^{T}\Sigma_{\tilde{Z}\tilde{Z}}^{-1}\left(\tilde{Z}-\mu_{\tilde{Z}}\right)$ is given by [22]

$$
p(w;\beta)=\frac{\beta}{\Gamma\left(\frac{n}{\beta}\right)2^{\frac{n}{\beta}}}w^{n-1}\exp\left(-\frac{1}{2}w^{\beta}\right).
$$

Therefore,

$$
E\left[\log p_{Z,Z'}\left(z,z'\right)\right]=\log\frac{\Phi(\beta,2n)}{\left[\det\left(\Sigma_{\tilde{Z}\tilde{Z}}\right)\right]^{1/2}}-\frac{2n}{\beta}.
$$

Similarly,

$$
\begin{aligned}
E\left[\log p_{Z}(z)\right]&=\log\frac{\Phi(\beta,n)}{\left[\det\left(\Sigma_{ZZ}\right)\right]^{1/2}}-\frac{n}{\beta},\\
E\left[\log p_{Z'}\left(z'\right)\right]&=\log\frac{\Phi(\beta,n)}{\left[\det\left(\Sigma_{Z'Z'}\right)\right]^{1/2}}-\frac{n}{\beta}.
\end{aligned}
$$

Substituting these expectations into (7) in main paper leads to

$$
\begin{aligned}
&I\left(Z,Z'\right)\\
&=\frac{1}{2}\log\frac{\det\left(\Sigma_{ZZ}\right)\det\left(\Sigma_{Z'Z'}\right)}{\det\left(\Sigma_{\tilde{Z}\tilde{Z}}\right)}+\log\frac{\Phi(\beta,2n)}{[\Phi(\beta,n)]^{2}}\\
&=\frac{1}{2}\log\frac{\det\left(\Sigma_{ZZ}\right)\det\left(\Sigma_{Z'Z'}\right)}{\det\left(\Sigma_{\tilde{Z}\tilde{Z}}\right)},
\end{aligned}
\tag{13}
$$

where we used $\Phi(\beta,n)=\frac{\beta\Gamma(n/2)}{2^{n/(2\beta)}\pi^{n/2}\Gamma(n/(2\beta))}$ and the following relation

$$
\frac{\Phi(\beta,2n)}{[\Phi(\beta,n)]^{2}}=\frac{\beta\Gamma(n)}{\Gamma(n/\beta)}\frac{\left[\Gamma\left(\frac{n}{2\beta}\right)\right]^{2}}{\left[\beta\Gamma\left(\frac{n}{2}\right)\right]^{2}}=\frac{1}{\beta}\frac{2^{n-\frac{1}{2}}}{2^{\frac{n}{\beta}-\frac{1}{2}}}\frac{\Gamma\left(\frac{n}{2}+\frac{1}{2}\right)}{\Gamma\left(\frac{n}{2\beta}+\frac{1}{2}\right)}\frac{\Gamma\left(\frac{n}{2\beta}\right)}{\Gamma\left(\frac{n}{2}\right)}=1.
$$

Then, using the relation between the dispersion matrix and covariance matrix

$$
\Sigma_{\tilde{X}\tilde{X}}=\frac{n\Gamma(n/(2\beta))}{2^{1/\beta}\Gamma((n+2)/(2\beta))}C_{\tilde{X}\tilde{X}},
$$

it follows that

$$
I\left(Z;Z'\right)=\frac{1}{2}\log\frac{\det\left(\Sigma_{ZZ}\right)\det\left(\Sigma_{Z'Z'}\right)}{\det\left(\Sigma_{\tilde{Z}\tilde{Z}}\right)}=\frac{1}{2}\log\frac{\det\left(C_{ZZ}\right)\det\left(C_{Z'Z'}\right)}{\det\left(C_{\tilde{Z}\tilde{Z}}\right)}.
$$

IV. Ablation Study

A. LOSS FUNCTION

We investigate the effectiveness of each term of the loss function. Specifically, we remove one of $\log\det C_{ZZ}$ term (w/o $\log\det C_{ZZ}$) and the $\log\det C_{Z'Z'}$ term (w/o $\log\det C_{Z'Z'}$) or both terms (w/o both) from the loss function. Additionally, we replace the $\log\det(C_{ZZ} - C_{Z'Z'})$ term with $\|Z - Z'\|^2$ since both terms aim to align the representations Z and Z'. Furthermore, we simplify the rescaling operation from $\tilde{M} = \frac{M - \mu_\lambda I}{\alpha} + I$ to $\tilde{M} = \frac{M}{\alpha} + I$ (w/o μ_λ). Results in Table I show that removing either $\log\det C_{ZZ}$ or $\log\det C_{Z'Z'}$ leads to performance decrease, yet the training still succeed. However, removing both leads to training failure. The reason behind this is straightforward. By minimizing the term $\log\det(C_{ZZ} - C_{Z'Z'})$, we aim to align the representations Z and Z'. The terms $\log\det C_{ZZ}$ and $\log\det C_{Z'Z'}$ play a crucial role in ensuring that these representations are informative enough to prevent representation collapse. When one of these terms is removed, the remaining term is expected to partially fulfill this, but becomes less effective.

TABLE I
Ablation on loss function. The experiment follows the same setup as in Table I in main paper, with Top-1 accuracy is reported.

Method	CIFAR-100	ImageNet-100
Original	70.5	81.1
w/o $\log\det C_{ZZ}$	66.6	76.5
w/o $\log\det C_{Z'Z'}$	67.7	78.8
w/o both	3.55	4.01
Using $\|Z - Z'\|^2$	69.8	79.6
w/o μ_λ in rescaling	70.6	80.3

It can be seen from Table I that replacing the $\log\det(C_{ZZ} - C_{Z'Z'})$ term with $\|Z - Z'\|^2$ decreases the performance. Although both $\log\det(C_{ZZ} - C_{Z'Z'})$ and $\|Z - Z'\|^2$ encourage consistency between Z and Z', their mathematical properties differ significantly. The term $\log\det(C_{ZZ} - C_{Z'Z'})$ encourages a holistic consistency in the structural properties of the feature spaces. Meanwhile, our derived loss function obviates the need to tune the balance ratio between the terms. Moreover, the results show that removing μ_λ term in the Taylor approximation, i.e. adding a fixed I_d to the three terms in the loss function does not affect the performance on CIFAR-100 but decreases the performance on ImageNet-100.

B. PROJECTOR HIDDEN DIMENSION AND PROJECTOR OUTPUT DIMENSION

We evaluate the effect of the projector's hidden dimension and projector output dimension in Table II. For both our method and Barlow Twins, there's a tendency that the increase of projector hidden dimension generally improves the performance on both the CIFAR-100 and ImageNet-100 datasets. Compared with CIFAR-100, both our method and Barlow Twins need a larger hidden dimension to achieve high performance on the more complex ImageNet-100 dataset. Using a momentum encoder, our method can achieve better performance and becomes more robust to projector hidden dimension. Moreover, similar to the results on the hidden dimension, both our method and Barlow Twins exhibit a trend that increasing the projector output dimension generally improves performance. The performance of Barlow Twins is particularly sensitive to projector output dimension, whereas our method performs well even with a very small projector output dimension of 256.

TABLE II
Impact of projector hidden/output dimension on accuracy.

	CIFAR100			ImageNet100		
Proj. hidden dim	Ours	Ours-M	Barlow Twins	Ours	Ours-M	Barlow Twins
2048	70.5	70.4	70.9	81.1	81.7	80.4
1024	70.8	70.4	70.2	80.2	81.5	79.3
512	69.1	70.1	69.6	79.5	81.4	78.3
256	67.9	70.0	68.0	78.7	80.6	76.9
Proj. output dim	Ours	Ours-M	Barlow Twins	Ours	Ours-M	Barlow Twins
2048	70.5	70.4	70.9	81.1	81.7	80.4
1024	70.3	70.4	69.7	80.6	81.1	79.6
512	70.6	70.5	66.5	80.4	81.7	77.4
256	70.5	71.1	62.1	80.3	81.2	73.6

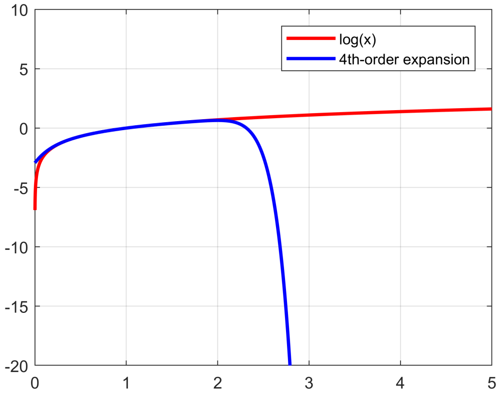

Fig. 2. Illustration of a fourth-order approximation of the log function in (10) in main paper.

TABLE III

Ablation study on the update interval and moving average coefficient ρ for eigenvalues tracking used for the rescaling operation.

update interval	ρ	CIFAR-100
1000	0	69.45
1000	0.1	70.22
1000	0.99	68.69
100	0.99	70.54
1	0.99	69.17
1	0	69.37

V. Experiment Implementation Details

For experiments on ImageNet-1K, we use a batch size of 1020 on 3 A100 GPUs for 100, 400, and 800 epochs. Training is conducted using 16-bit precision (FP16) and 4 batches of gradient accumulation to stabilize model updating and accelerate the training process. We use the LARS optimizer with a base learning rate of 0.8 for the backbone pretraining and 0.2 for the classifier training. The learning rate is scaled by lr = base_lr × batch_size/256 × num_gpu. We use a weight decay of 1.5E-6 for backbone parameters. The linear classifier is trained on top of frozen backbone. We follow the default setting as in Solo-learn benchmark [23] for the rest of the training hyper-parameters.

Recall that in implementing the loss function (9) from the main paper, the three log-determinant terms are expanded as

$$\begin{aligned}\log\det(M) &= \sum_{i=1}^{n}\log\lambda_i(M)\\ &= \sum_{i=1}^{n}\sum_{k=1}^{\infty}(-1)^{k+1}\frac{(\lambda_i(M)-1)^k}{k}\\ &= \sum_{k=1}^{\infty}(-1)^{k+1}\frac{\operatorname{tr}\left((M-I)^k\right)}{k} = \operatorname{tr}\left(\sum_{k=1}^{\infty}(-1)^{k+1}\frac{(M-I)^k}{k}\right),\end{aligned}\tag{10}$$

retaining only a p-th order approximation is kept, e.g., $p=4$ in the experiments of this work. As shown in Figure 2, a fourth-order approximation of the log function in (10) is sufficiently accurate around the value of 1.

For our method with a momentum encoder, we follow the setting of [4, 9] and use a two-layer predictor with hidden dimension 1024 for all datasets. For ImageNet-100, we set the base learning rate as 0.2 for backbone pretraining and 0.3 for the classifier. We set the weight decay of backbone parameters as 0.0001. For CIFAR-100, we set the base learning rate for backbone pretraining as 0.3 and the classifier as 0.2. The weight decay is set as 6E-5. For the rescaling operation ($\tilde{M}=\frac{M-\mu_\lambda I}{\alpha}+I$), we track the eigenvalues of C_{ZZ} with an update interval of 100 batches with a moving average coefficient ρ of 0.99. Table III shows the ablation study on the update interval and moving average coefficient ρ. Generally, a larger update interval should be used with a smaller moving average coefficient, and vice versa. This is reasonable as the two parameters together control the speed of the eigenvalue tracking. Overall, our method is insensitive to these two hyperparameters. Table 1 in main paper and Figure 3 depict the ablation study on the parameter β used for the rescaling operation. We adhere to the experimental setup described in Table 1 and

TABLE IV
Ablation study on the parameter β used for the rescaling operation .

β	CIFAR-100
7	69.85
6	70.10
5	70.17
4	70.09
3	69.46
1	NaN

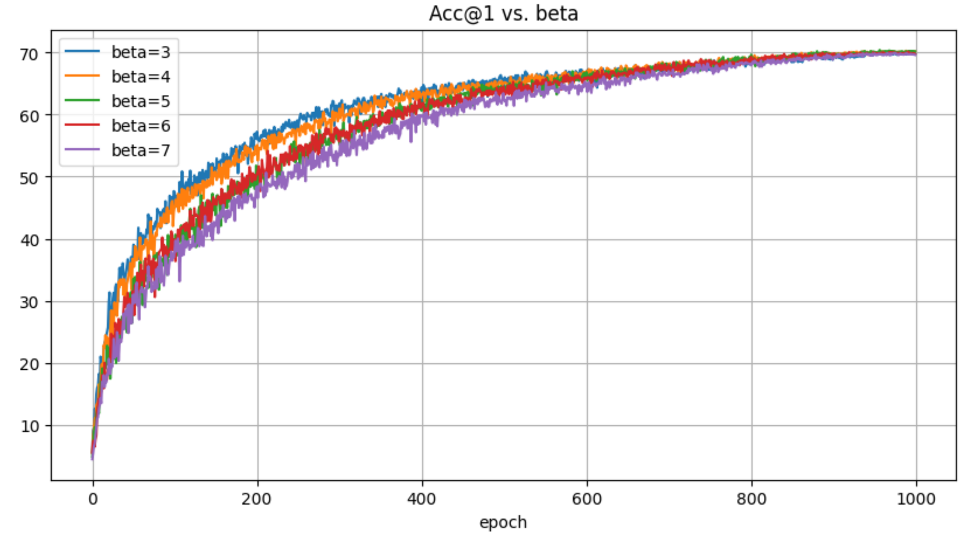

Fig. 3. The convergence curves of our method on CIFAR-100 for different values of the parameter β used for the rescaling operation.

report the Top-1 accuracy on CIFAR-100. As shown in Table IV and Figure 3, a smaller β results in faster convergence during training. However, excessively small values may lead to training failure. We set $\beta = 5$ for all the experiments.

References

[1] J. Bromley, I. Guyon, Y. LeCun et al., "Signature verification using a" siamese" time delay neural network," NeurIPS, vol. 6, 1993.

[2] T. Chen, S. Kornblith, M. Norouzi et al., "A simple framework for contrastive learning of visual representations," in International conference on machine learning. PMLR, 2020, pp. 1597–1607.

[3] K. He, H. Fan, Y. Wu et al., "Momentum Contrast for Unsupervised Visual Representation Learning," in CVPR, Jun. 2020, pp. 9726–9735.

[4] J.-B. Grill, F. Strub, F. Altché et al., "Bootstrap Your Own Latent - A New Approach to Self-Supervised Learning," in Advances in Neural Information Processing Systems, 2020.

[5] J. Zbontar, L. Jing, I. Misra et al., "Barlow Twins: Self-Supervised Learning via Redundancy Reduction," in International Conference on Machine Learning, Jul. 2021, pp. 12 310–12 320.

[6] M. Caron, I. Misra, J. Mairal et al., "Unsupervised Learning of Visual Features by Contrasting Cluster Assignments," in Advances in Neural Information Processing Systems, vol. 33, 2020, pp. 9912–9924.

[7] I. Misra and L. van der Maaten, "Self-Supervised Learning of Pretext-Invariant Representations," Dec. 2019.

[8] Z. Wu, Y. Xiong, S. Yu et al., "Unsupervised Feature Learning via Non-Parametric Instance-level Discrimination," May 2018.

[9] X. Chen and K. He, "Exploring simple siamese representation learning," in CVPR, 2021, pp. 15 750–15 758.

[10] M. Zheng, S. You, F. Wang et al., "Ressl: Relational self-supervised learning with weak augmentation," 2021.

[11] C. Feng and I. Patras, "Adaptive Soft Contrastive Learning," in International Conference on Pattern Recognition (ICPR), Aug. 2022, pp. 2721–2727.

[12] F. Wang, T. Kong, R. Zhang et al., "Self-Supervised Learning by Estimating Twin Class Distributions," Dec. 2021.

[13] A. Bardes, J. Ponce, and Y. LeCun, “Vicreg: Variance-invariance-covariance regularization for self-supervised learning,” in The Tenth International Conference on Learning Representations, ICLR, 2022.

[14] M. I. Belghazi, A. Baratin, S. Rajeshwar et al., “Mutual information neural estimation,” in International Conference on Machine Learning, 2018, pp. 531–540.

[15] M. Gutmann and A. Hyvärinen, “Noise-contrastive estimation: A new estimation principle for unnormalized statistical models,” in Thirteenth International Conference on Artificial Intelligence and Statistics, 2010, pp. 297–304.

[16] Z. Wu, Y. Xiong, S. X. Yu et al., “Unsupervised feature learning via non-parametric instance discrimination,” in IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 3733–3742.

[17] A. v. d. Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” arXiv preprint arXiv:1807.03748, 2018.

[18] B. Poole, S. Ozair, A. Van Den Oord et al., “On variational bounds of mutual information,” in International Conference on Machine Learning, 2019, pp. 5171–5180.

[19] X. Liu, Z. Wang, Y.-L. Li et al., “Self-supervised learning via maximum entropy coding,” NeurIPS, vol. 35, pp. 34 091–34 105, 2022.

[20] R. Shwartz-Ziv, R. Balestriero, K. Kawaguchi et al., “An information-theoretic perspective on variance-invariance-covariance regularization,” NeurIPS, 2023.

[21] A. Kraskov, H. Stögbauer, and P. Grassberger, “Estimating mutual information,” Physical Review E, vol. 69, no. 6, p. 066138, 2004.

[22] G. Verdoolaege and P. Scheunders, “On the Geometry of Multivariate Generalized Gaussian Models,” Journal of Mathematical Imaging and Vision, vol. 43, no. 3, pp. 180–193, Jul. 2012.

[23] V. G. T. da Costa, E. Fini, M. Nabi et al., “Solo-learn: A Library of Self-supervised Methods for Visual Representation Learning,” Journal of Machine Learning Research, vol. 23, no. 56, pp. 1–6, 2022.