Deciphering the Projection Head: Representation Evaluation Self-supervised Learning

Jiajun Ma[1,2] , **Tianyang Hu**[3] , **Wenjia Wang**[1,2]
[1]Hong Kong University of Science and Technology
[2]Hong Kong University of Science and Technology (Guangzhou)
[3]Huawei Noah's Ark Lab
jmabh@connect.ust.hk, hutianyang.up@outlook.com, wenjiawang@ust.hk

Abstract

Self-supervised learning (SSL) aims to learn the intrinsic features of data without labels. Despite the diverse SSL architectures, the projection head always plays an important role in improving downstream task performance. In this study, we systematically investigate the role of the projection head in SSL. We find that the projection head targets the uniformity aspect, which maps samples into uniform distribution and enables the encoder to focus on extracting semantic features. Drawing on this insight, we propose a Representation Evaluation Design (RED) in SSL models in which a shortcut connection between the representation and the projection vectors is built. Our extensive experiments with different architectures (including SimCLR, MoCo-V2, and SimSiam) on various datasets demonstrate that the RED-SSL consistently outperforms their baseline counterparts in downstream tasks. Furthermore, the RED-SSL learned representations exhibit superior robustness to previously unseen augmentations and out-of-distribution data.

1 Introduction

Extracting meaningful representations from a large amount of unlabeled data is a crucial task in self-supervised learning. With the rapid progress in the SSL [Chen *et al.*, 2020a; He *et al.*, 2020; Wang *et al.*, 2021; Caron *et al.*, 2020; Grill *et al.*, 2020; Chen and He, 2021; Caron *et al.*, 2021; He *et al.*, 2022; Chen *et al.*,], simple classifiers learned from pre-trained representations can achieve comparable performance to those from supervised learning. Despite its empirical success, the underlying mechanisms of SSL still require further exploration. Many efforts have been devoted to studying the loss function [Sohn, 2016; Oord *et al.*, 2018; Wang and Isola, 2020; Li *et al.*, 2020; Khosla *et al.*, 2020; Wang and Liu, 2021; Hu *et al.*, 2022], and the construction of positive pairs [Arora *et al.*, 2019; Wen and Li, 2021; Wang and Isola, 2020; Tian *et al.*, 2021; Wang and Liu, 2021; Wang *et al.*, 2022a], while less are paid on the investigation of the model architectural components. Typically, the SSL architecture includes two components: an encoder and a projection head. The encoder is usually a discriminative model like ResNet [He *et al.*, 2016] or ViT [Dosovitskiy *et al.*, 2020] that aims to extract semantic features; the projection head is a multi-layer perceptron used in pre-training. After the pre-training, the projection head is discarded, and the encoder outputs (representation vectors) are used for the downstream tasks. The inclusion of projection head during the pre-training significantly improves the SSL method in downstream performance [Chen *et al.*, 2020a], leading to its widespread adoption in various SSL architectures [Chen *et al.*, 2020a; He *et al.*, 2020; Chen *et al.*, 2020b; Caron *et al.*, 2020; Grill *et al.*, 2020; Zbontar *et al.*, 2021; Chen and He, 2021; He *et al.*, 2022]. However, the mechanism behind the contribution of the projection head during pre-training has not been fully understood.

Through an in-depth analysis of the pre-training objective values across the intermediate layers of SSL, we discover that the projection head mainly promotes the uniformity objective, i.e., mapping the samples closer to the uniform distribution. In contrast, the encoder mainly serves to enhance alignment, i.e., minimizing the distance between similar samples, such as a training data point and its augmentations. We demonstrate that this phenomenon widely exists in SSL methods, including contrastive methods such as SimCLR and MoCo-V2, and non-contrastive methods such as SimSiam. While augmentation invariance (alignment) may relate more to extracting semantic features [Purushwalkam and Gupta, 2020; Von Kügelgen *et al.*, 2021; Wang *et al.*, 2022a; Wang *et al.*, 2022c], the projection head can effectively prevent the SSL model maps the inputs to the identical constant output, known as trivial constant solution [Jing *et al.*, 2021; Li *et al.*, 2022].

Building on our understanding of the projection head, we propose a novel approach called Representation Evaluation Design (RED) for SSL methods. RED involves creating a shortcut connection between the representation and projection vectors, allowing the benefits (lower entropy, general augmentation, and downstream guidance) of the representation vector to bypass the projection head and guide the training directly. As a result, RED-SSL strengthens the augmentation invariance of representation (extracting semantic information [Von Kügelgen *et al.*, 2021; Purushwalkam and Gupta, 2020; Wang *et al.*, 2022a]) without impairing the uniformity of the projection head. Through comprehensive comparison experiments between the baseline SSL methods (SimCLR, MoCo-V2, SimSiam) and the RED-version (RED-SimCLR, RED-MoCo-V2, and RED-SimSiam) in various datasets (CIFAR-

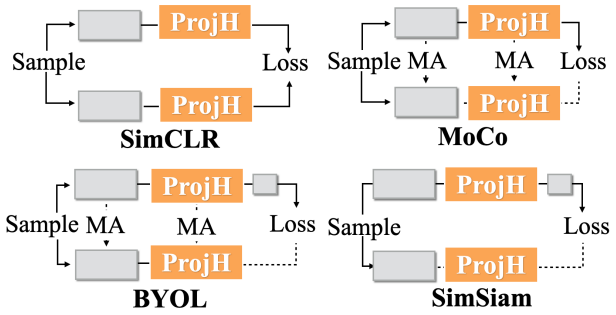


Figure 1: Demonstration of SSL network architectures. "MA" stands for moving average operation, "ProjH" for the projection head. The dashed line stands for the stop gradient operation.

10, CIFAR-100 [Krizhevsky, 2009], ImageNet [Deng *et al.*, 2009]), we observe a consistent performance improvement in the downstream classification tasks using k-nearest neighbor (kNN) [Wu *et al.*, 2018] and linear classifier. Furthermore, the RED-SSL learned representations exhibit *stronger robustness* to augmentation in the downstream evaluation.

Our main contributions are summarized as follows.

- We uncover that the projection head mainly serves as a uniform projector, regardless of whether the uniformity appears explicitly in the loss function of SSL. Thus, the projection head enables the encoder to focus on boosting alignment without degenerating to a collapsed constant. It explains the combination of the encoder and projection head outperforms the individual encoder.
- We reveal that the encoder outputs (representation vectors) exhibit superiority in terms of augmentation robustness, lower entropy, and better downstream task performance than the outputs of the projection head (projection vectors). It explains why the representation vectors are used in the downstream tasks.
- We propose a representation evaluation design (RED) that bridges the representation information and the SSL objective functions. Extensive experiments on different SSL methods, various datasets, and different classifiers demonstrate that our proposed design can consistently improve the downstream task performance of the baseline models and is more robust to unseen augmentations and out-of-distribution data.

2 Related Works

The projection head in SSL. The projection head design was initially introduced in SimCLR [Chen *et al.*, 2020a], which differentiates the pre-training and the downstream phase as separate objectives for the projection vector z and representation vector r, respectively. This projection head design is widely adopted by the later proposed methods [He *et al.*, 2020; Chen *et al.*, 2020b; Grill *et al.*, 2020; Caron *et al.*, 2020; Zbontar *et al.*, 2021; Chen and He, 2021], where some architectures are displayed in Figure 1. [Chen *et al.*, 2020a] indicates that the downstream classification does not increase monotonically as the number of layers of projection head increases. [Wang *et al.*, 2022b] indicated that the projection head is the key to enhancing the transferability by measuring the downstream supervised accuracy between different network layers. However, for unlabeled SSL, assessing it solely based on supervised performance may raise doubts. In contrast, we examine the SSL objective values among the layers, which illustrates the benefits provided by the projection head more directly. [Gupta *et al.*, 2022] regarded the projection head as a low-rank mapping such that the trained vectors can be more style-invariant and generalize better. We think that this does not capture the complete picture. For example, in the ablation study conducted on SimCLR [Chen *et al.*, 2020a], altering the dimensionality of the non-linear projection head does not significantly impact the downstream performance. Additionally, in SimSiam [Chen and He, 2021], reducing the projection dimension adversely affects the downstream performance, indicating that low-rank projection substantially deteriorates the performance of SimSiam. In contrast, our interpretation of the projection head applies to various SSL architectures (including SimCLR, MoCo-V2, and SimSiam) and inspires designs that universally improve SSL performance.

Analysis of SSL. The theoretical understanding of SSL has been the focus of many works [Arora *et al.*, 2019; Tosh *et al.*, 2021; HaoChen *et al.*, 2021; Wen and Li, 2021; Ji *et al.*, 2021; Tian, 2022]. The analysis of the SSL loss and the architecture is relatively fewer. [Wang and Isola, 2020] split the InfoNCE loss as alignment and uniformity, and demonstrate that uniformity stands for uniformly distributing on the hypersphere based on Gaussian potential kernel. Compared with [Wang and Isola, 2020], our study integrates alignment and uniformity concepts into the analysis of SSL architectures. This integration enables a deeper exploration of SSL network design. Importantly, our analysis goes beyond contrastive SSL approaches and also includes non-contrastive SSL methods, providing a more comprehensive understanding. [Hu *et al.*, 2022] relates contrastive SSL to the stochastic neighbor embedding(SNE). [Wang *et al.*, 2022a] considers augmentation encourages the different intra-class samples to be overlapped, and thus positive alignment could attract the intra-class samples together. [Wang and Liu, 2021] states that uniformity guides the contrastive model to learn separable features, and proper temperature gives tolerance to semantically similar samples. [Wen and Li, 2022] reveals that the identity-initialized prediction head prevents BYOL from the training collapse. [Saunshi *et al.*, 2022] point out that the inductive biases within the contrastive function class contribute to the downstream success.

3 Projection Head Is a Uniformity Projector

This section investigates the interplay between the encoder and the projection head during the pre-training process. Specifically, we investigate the alignment & uniformity values within the SSL architectures. We reveal that the projection head prioritizes the uniformity objective, and the encoder focuses on the alignment objective in the pre-training.

The InfoNCE loss [Sohn, 2016; Oord *et al.*, 2018; Chen *et*

al., 2020a; Wang and Isola, 2020; Yeh *et al.*, 2022] is

$$\begin{aligned}\sum_{i=1}^{n} l_i &= -\sum_{i=1}^{n} \log \frac{\exp(z_i^{(1)} z_i^{(2)}/\tau)}{\sum_{j\in\{1,n\},k,l\in\{1,2\}} \exp(z_i^{(k)} z_j^{(l)}/\tau)} \\ &= \sum_{i=1}^{n} (-z_i^{(1)} z_i^{(2)}/\tau) + \sum_{i=1}^{n} \log \sum_{j=1}^{n} \exp(z_i^{(k)} z_j^{(l)}/\tau) \\ &= \text{alignment loss} + \text{uniformity loss}.\end{aligned} \tag{1}$$

We can define

$$\begin{aligned}\text{vector } x \text{ alignment value} &= \sum_{i=1}^{n} (-x_i^{(1)} x_i^{(2)}/\tau) \\ \text{vector } x \text{ uniformity value} &= \sum_{i=1}^{n} \log \sum_{j=1}^{n} \exp(x_i^{(k)} x_j^{(l)}/\tau)\end{aligned} \tag{2}$$

where n is the batch size, z_i is the projection vector of a sample i. In (1), $z_i^{(1)}, z_i^{(2)}$ are referred to as positive pairs, since they are from the same sample under different augmentations, and $z_i^{(k)}, z_j^{(l)}$ are regarded as negative pairs. In the InfoNCE loss Formula1, the first term corresponds to the positive pair alignment loss: $\sum_{i=1}^{n}(-z_i^{(1)} z_i^{(2)}/\tau)$, which is minimized if the projection z is invariant to the training augmentation. The second term represents uniformity loss: $\sum_{i=1}^{n} \log \sum_{j=1}^{n} \exp(z_i^{(k)} z_j^{(l)}/\tau)$, which is minimized if the projection z is distributed uniformly on the hypersphere [Wang and Isola, 2020]. Inspired by the alignment and uniformity formula in 1, we define the alignment and uniformity values of given vector x in (2), respectively. In our subsequent analysis, we compute the alignment and uniformity values for intermediate layer output vectors in various SSL networks.

Figures 2 depict the alignment and uniformity values using Formula (2) for the intermediate layers of the SSL models: (a) MoCo-V2; (b) SimSiam. (SimCLR analysis is placed in Appendix B.1) All are trained for 200 epochs in CIFAR-10, and the encoder is ResNet-18. In the encoder part of Figures 2, the uniformity values fluctuate without showing a consistent decrease, while the alignment values decrease across diverse augmentation types (except for Gaussian augmentation, which lacks semantic meaningfulness). Moving into the projection head, there is a noticeable decrease in uniformity values, accompanied by an increase in alignment values. This phenomenon in the projection head suggests that the samples are mapped closer to a uniform distribution on the hyper-sphere, albeit at the expense of losing some augmentation invariance.

Table 1 presents the explicit changes in alignment and uniformity values (computed using Formula (2)) within the encoder and projection head. In the encoder part, alignment values decrease while uniformity values increase. Within the projection head, the decrease in uniformity values coincides with an increase in alignment values, indicating its role as a uniformity projector. In fact, with the alignment and uniformity values dropping within the encoder and projection head separately, the encoder output (representation layer) owns the lowest alignment value and possesses semantically meaningful features. This SSL inner investigation helps explain why introducing the projection head can improve the downstream performance of the encoder output (representation).

		UNIFORMITY	ALIGNMENT
SIMCLR	ENC	+0.035	-0.356
	PRH	-0.057	+0.266
MOCO-V2	ENC	+0.049	-0.481
	PRH	-0.028	+0.151
SIMSIAM	ENC	+0.019	-0.401
	PRH	-0.025	+0.143

Table 1: Uniformity and alignment changes. For abbreviation, Enc stands for Encoder part of SSL model, PrH for Projection Head. The numbers in the Uniformity column and EnC row (based on Formula(2)) are obtained by: (the uniformity value of the last layer of encoder output) − (the uniformity value of the first layer of encoder output); The numbers in the Uniformity column and PrH row (based on Formula(2)) are obtained by:(the uniformity value of the last layer of projection head output) − (the uniformity value of the first layer of projection head output). The Alignment column is obtained similarly.

In SimSiam pre-training, uniformity terms are not included in the training objective. Nevertheless, similar observations found in contrastive SSL methods like SimCLR and MoCo-v2 can still be made. The significant decrease in uniformity values within the projection head, as shown in the upper subfigure of SimSiam (b) in Figure 2 and the SimSiam row in Table 1, indicates that the projection head implicitly promotes uniformity and prevents the collapse of SimSiam pre-training.

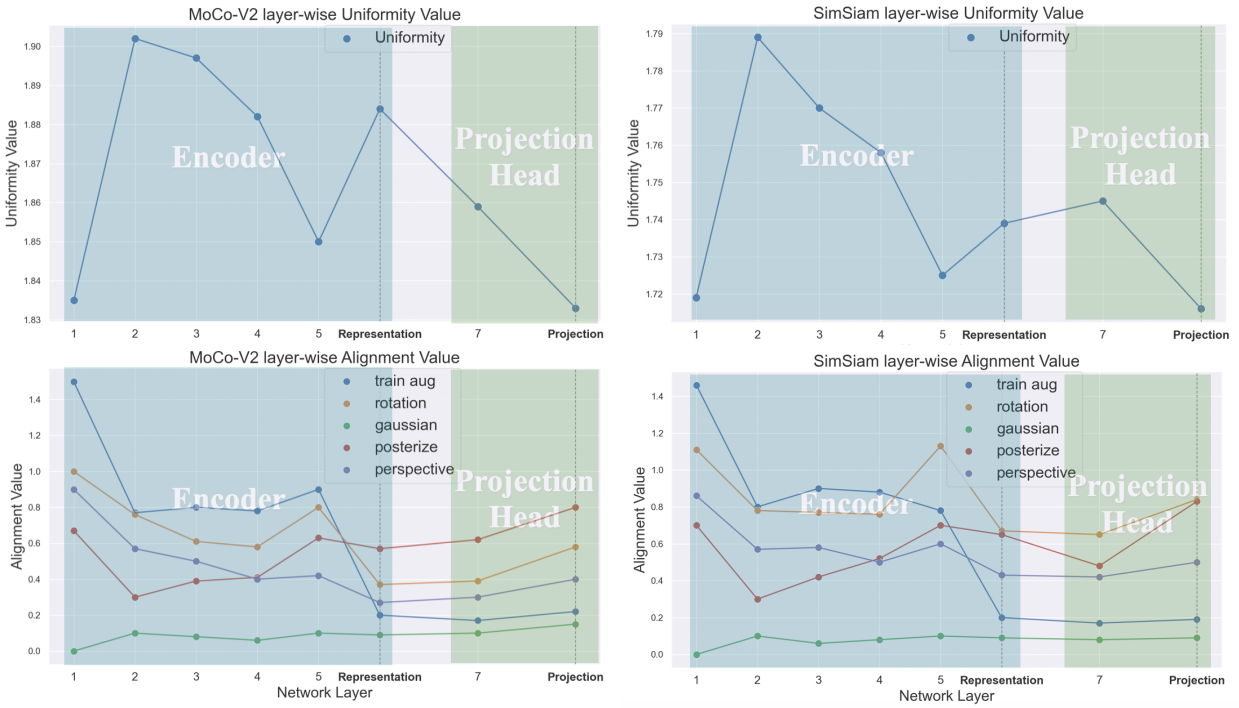
After uncovering the projection head is a uniformity projector through SSL architectural analysis, we aim to understand the effect on the encoder output layer (representation vectors) in the following analysis.

	REPR		PROJ	
	TRAIN	TEST	TRAIN	TEST
TRAIN AUGMENTATION	0.823	0.802	0.894	0.857
ANGLE ROTATE	0.559	0.558	0.479	0.470
GAUSSIAN BLUR	0.951	0.947	0.965	0.958
CENTER CROP	0.219	0.221	0.134	0.130
RANDOM POSTERIZE	0.509	0.509	0.422	0.413
RANDOM PERSPECTIVE	0.730	0.728	0.704	0.698

Table 2: Augmentation robustness. The cosine similarity between positive sample pairs under different types of augmentations. The "train augmentation" row stands for the augmentation types used during the pre-training. The "Repr" column stands for Representation and "Proj" stands for projection vectors.

4 Representation Vector Analysis

Having revealed the projection head's role as a uniformity projector through SSL architectural analysis, we conduct a detailed analysis of the encoder output layer (representation vectors) in this section. Specifically, we demonstrate that the representation vectors exhibit superiority in augmentation robustness, lower entropy, and better downstream task performance compared to the projection vectors.



((a)) ((b))

Figure 2: Layer-wise uniformity and alignment value analysis. Conducting the alignment and uniformity values calculation using Formula (2) for each intermediate layer in the SSL architectures. (a) MoCo-V2 (b) SimSiam.

4.1 Robustness to Unseen Augmentations

Augmentation invariance plays a crucial role in guiding the extraction of semantic features in SSL [Von Kügelgen *et al.*, 2021; Purushwalkam and Gupta, 2020; Wang *et al.*, 2022a] and contributes to the achievements of non-contrastive models [Grill *et al.*, 2020; Chen and He, 2021; He *et al.*, 2022]. However, in the pre-training process, only a limited number of augmentation types are typically applied, while the invariance to unseen augmentation types can also be advantageous in extracting additional semantic information.

To study the robustness to unseen augmentations of the representation and projection vectors, we compare the cosine similarity of the positive pairs based on the representation and the projection vectors. Table 2 records the cosine similarity under different augmentations, calculated with representation and projection vectors, respectively. The "train augmentation" row in Table 2 refers to the augmentation types used during the training (random combination of RandomResizedCrop, HorizontalFlip, ColorJitter, and RandomGrayscale, specified in [Chen *et al.*, 2020a]). The projection vectors exhibit a greater cosine similarity for the training augmentation, but for all other unseen augmentation types, the cosine similarity of the representation vectors is significantly higher than that of the projection vectors. The exception to this is Gaussian blur, which is not a semantically meaningful augmentation. This superior augmentation robustness indicates that the representation vectors are capable of extracting meaningful semantic information beyond the predetermined augmentation types.

4.2 Entropy Analysis

Uniformity is crucial in preventing a model from collapsing into a trivial constant state [Wang and Liu, 2021]. However, promoting uniformity can lead to increased entropy, as shown in Proposition 1. In this section, we demonstrate that the entropy of representation vectors (encoder output) is lower than that of projection vectors (projection head output). This is because the projection head prioritizes the uniformity objective, while the encoder focuses on the alignment objective.

Proposition 1. *Encouraging uniformity is equivalent to reducing the KL divergence towards the uniform distribution, and approaching closer to the uniform distribution results in an entropy increase.*

The proof of Proposition 1 is based on the derivation in (A.2) and (A.3) of Appendix A. Intuitively, since the uniform distribution is the distribution that has the largest entropy for bounded variables, and the projection head encourages uniformity, the output distribution of the projection head has a higher entropy than its original distribution. To provide a quantitative comparison, we use the discrete entropy estimator [Beirlant *et al.*, 1997; Lombardi and Pant, 2016] defined as the following Formula, b refers to the m non-overlapping sub-regions that constitute all samples. Defined as: $\hat{H} = \frac{1}{m}\sum_{b=1}^{m}\sum_{\text{label}=1}^{k} -\hat{p}_{\{b,\text{label}\}}\log\hat{p}_{\{b,\text{label}\}}$, where: $\hat{p}_{b,\{\text{label}=\text{i}\}}$ = (labeled i sample counts in region b) / (sample counts in region b).

Table 3 compares the estimated entropy calculated using

representation and projection vectors from SimCLR, MoCo-V2, and SimSiam, all trained for 200 epochs on the CIFAR-10 dataset using ResNet-18 as the encoder. The results verify that the estimated entropy of representation vectors is consistently lower than that of projection vectors, indicating that the encoder implicitly transfers the burden of uniformity to the projection head. Consequently, the representation vectors enjoy a smaller entropy and contain a wealth of information useful for downstream tasks. Our quantitative analysis of entropy provides a formal and comprehensive illustration, offering a contrast to the visually-oriented T-SNE scatter plots presented in [Wang *et al.*, 2022b]. Figure B.3 in Appendix B.2 presents a visual evaluation of the entropy.

ENTROPY	REPRESENTATION	PROJECTION
SIMCLR	1.7476	1.850
MOCO-V2	1.5131	2.0315
SIMSIAM	1.8611	1.8940

Table 3: Compare the entropy of representation and projection vectors in SimCLR, MoCo-V2, and SimSiam, all of which were trained for 200 epochs using ResNet-18 as the encoder on the CIFAR-10 dataset.

4.3 Relationship to Downstream Task

With the projection head targeting the uniformity objective, SSL models allow the representation vectors to have higher alignment and therefore, contain more semantic information and are more closely related to downstream performance than projection vectors, as demonstrated in this subsection.

We calculate the mean of the pair-wise cosine similarity of representation r and projection vectors z in Formula 3

$$s_i^{(r)} = \frac{1}{n}\sum_{j!=i}^{j=n} r_i r_j, s_i^{(z)} = \frac{1}{n}\sum_{j!=i}^{j=n} z_i z_j, \tag{3}$$

where r_i and z_i refer to representation and projection vectors of sample i. Define y_i as the downstream misclassification of sample i. Specifically, $y_i = 1$ represents misclassification while $y_i = 0$ represents correct classification in the downstream task. The correlation between the sample average of pairwise cosine similarity $s^{(r)}, s^{(z)}$ and misclassification result y is denoted as $\rho(s^{(r)}, y)$ and $\rho(s^{(z)}, y)$. Intuitively, a sample with higher similarity to others is more difficult to differentiate and is thus more likely to be misclassified.

Table 4 displays the correlations $\rho(s^{(r)}, y)$, $\rho(s^{(z)}, y)$ for the representation and projection vectors, respectively. The results indicate that representation vectors are more closely related to downstream task performance, as they exhibit a higher correlation. To further verify that the quality of the representation vectors can influence the downstream task performance more directly, we compute the error rate split for the representation and projection vectors. Specifically, we split the samples into two groups according to the pair-wise cosine similarity of representation vectors r, as follows:

$$\mathcal{G}_1^{(r)} = \{i : s_i^{(r)} > \text{median of } s^{(r)}\}, \mathcal{G}_2^{(r)} = \{1, ..., n\} \backslash \mathcal{G}_1^{(r)}.$$

CIFAR-10	REPRESENTATION	PROJECTION
CORRELATION	0.1482	0.021
ERROR RATE SPLIT	14.1% | 6.7%	11.1% | 9.7%
CIFAR-100	REPRESENTATION	PROJECTION
CORRELATION	0.1746	0.0642
ERROR RATE SPLIT	46.1%|32.2%	42.1%|36.2%

Table 4: Downstream error rate. The correlations $\rho(s^{(r)}, y)$, $\rho(s^{(z)}, y)$ for the representation and projection vectors. The error rate, defined in (4), stands for the two groups of error rate comparison, divided by 50% percentile of the sample average of pair-wise cosine similarity s. (representation-based and projection-based error rates comparison are: $e_1^{(r)}|e_2^{(r)}$ and $e_1^{(z)}|e_2^{(z)}$.

The error rate for each group is defined as

$$e_j^{(r)} = \frac{1}{\text{card}(\mathcal{G}_j^{(r)})}\sum_{i\in\mathcal{G}_j^{(r)}} y_i, \quad j = 1, 2. \tag{4}$$

The split and the error rates of the projection vectors z: $\mathcal{G}^{(z)}, e^{(z)}$, can be obtained similarly. Intuitively, the error rate of $\mathcal{G}_1^{(r)}$ (or $\mathcal{G}_1^{(z)}$) is larger than that of $\mathcal{G}_2^{(r)}$ (or $\mathcal{G}_2^{(z)}$), since the samples in the former group are harder to be differentiated. The gap between the error rates of the two groups should be large if the cosine similarity accurately represents the difficulty of the downstream task. Table 4 exhibits that the gap in error rates for representation vectors is much larger than that for projection vectors, suggesting that the sample pair-wise similarity based on representation vectors is more closely related to downstream task performance.

5 Representation Evaluation Design in SSL

In Section 4, we demonstrate that representation exhibits superior robustness to universal augmentations, lower entropy, and stronger relevance to downstream tasks compared to projection vectors. These merits motivate us to propose Representation Evaluation Design (RED) in the SSL pre-training, which is applicable for both the contrastive and non-contrastive SSL.

Our motivation is to enhance the performance of SSL by targeting samples that are difficult to differentiate from the others. Building on our discovery that representation vectors r offer benefits including better augmentation invariance, lower entropy, and stronger downstream relevance than the projection vectors, we propose the representation evaluation weights w in (5) for usage in SSL pre-training loss.

$$w_i = (\text{percentile}_j(\exp(r_i r_j/\eta), k\%))^{-1}, \tag{5}$$

In (5), r is the representation vector, η is the representation temperature parameter, and k is the percentile parameter. The weight w measures the reciprocal of the percentile of cosine similarity of each sample to others within a training batch. With re-weighting w_i included in SSL (RED-SSL), the objective functions of RED-Contrastive and RED-Non-Contrastive

are the following (where τ is the temperature hyperparameter):

$$\begin{aligned} L^{\text{RED-Contrastive}} &= \sum_{i=1}^{n} -\log \frac{w_i \exp(z_i^{(1)} z_i^{(2)}/\tau)}{\sum_{j=1}^{N} \exp(z_i z_j/\tau)} \\ &= -\sum_{i=1}^{n} \log w_i - \sum_{i=1}^{n} \log \frac{\exp(z_i^{(1)} z_i^{(2)}/\tau)}{\sum_{j=1}^{N} \exp(z_i z_j/\tau)}, \end{aligned} \tag{6}$$

$$\begin{aligned} L^{\text{RED-Non-Contrastive}} &= \sum_{i=1}^{n} -\log w_i \exp(z_i^{(1)} z_i^{(2)}/\tau) \\ &= -\sum_{i=1}^{n} \log w_i - \sum_{i=1}^{n} z_i^{(1)} z_i^{(2)}/\tau, \end{aligned} \tag{7}$$

Incorporating w into the SSL loss in (6) and (7) assigns a higher loss to the sample with greater similarities to others in the batch, since this sample tends to be more challenging for the model in downstream tasks. Additionally, as representation vectors are more related to the downstream performance, the representation-based w actively guides the training to boost downstream accuracy. Furthermore, we leverage the superior augmentation robustness of representation vectors to adjust the alignment term of projection vectors, thereby equipping features with stronger augmentation robustness in RED-SSL.

With the logarithm expansion of RED-SSL losses in Formula (6) and (7), RED-SSL essentially establishes a shortcut connection between the representation based w and projection-based SSL loss, illustrated in Figures 3. During the *batch* stochastic gradient descent, the gradients of $\log w_i$ can bypass the projection head and reach the representation vector directly. This approach also implicitly helps to avoid vanishing gradients of the representation vector.

Remark 1. *batch optimization is important in RED-SSL. In* (6) *and* (7)*, RED-SSL essentially adds w term with the SSL loss and does not reweigh samples within the batch. Instead, the w of RED updates the model training in a batch-wise manner in batch optimization.*

It should be noted that w_i in (5) is defined by the $k\%$ percentile of the exponential pairwise product: $\text{percentile}_j(\exp(r_i r_j/\eta), k\%)$ instead of the average: $\frac{1}{n}\sum_j \exp(r_i r_j/\eta)$. The percentile approach allows us to focus on more similar samples without considering other samples that are already well separated, resulting in a better assessment of sample-wise similarity within a training batch. In contrast, the average operation considers all samples equally in the calculation. Table 5 compares the downstream classification accuracy of MoCo-V2 trained with the baseline contrastive loss (InfoNCE) versus the percentile, and average-based RED-contrastive loss. The model was trained for 200 epochs on the CIFAR-10/100 using ResNet-18 as the encoder.

LOSS TYPE	CIFAR-10	CIFAR-100
BASELINE	83.0%	62.1%
PERCENTILE BASED w	86.4%	63.5%
AVERAGE BASED w	85.1%	62.6%

Table 5: Percentile and average design comparison. Linear evaluation of MoCo-V2 based on percentile-based w and average-based w.

6 Experiments

In this section, we compare the existing SSL models: SimCLR, MoCo-V2, and SimSiam, with our representation evaluation design SSL (RED-SSL): RED-SimCLR, RED-MoCo-V2, and RED-SimSiam, respectively. Our results demonstrate that RED consistently improves downstream performance across diverse architectures, datasets, and out-of-distribution (OOD) settings. Similar to the superior robustness of shortcut connections to perturbations [Cazenavette *et al.*, 2021; Reshniak and Webster, 2020], we show that RED-SSL models also gain more robustness against unseen augmentations. We conduct an ablation study on the two hyperparameters within the re-weighting term w in RED, namely the percentile parameter $k\%$ and the representation temperature η. The ablation results exhibit that RED can steadily boost downstream performance. However, due to space limitations, we include the ablation study in Appendix B.6.

	CIFAR-10 KNN|LINEAR	CIFAR-100 KNN|LINEAR
SIMCLR	81.4%|83.0%	52.1%|56.3%
RED-SIMCLR	84.6%|86.4%	56.6%|58.6%
MOCO-V2	83.2%|85.1%	55.7%|62.1%
RED-MOCO-V2	85.4%|87.2%	58.5%|63.5%
SIMSIAM	81.8%|82.9%	50.7%|52.0%
RED-SIMSIAM	84.6%|85.7%	51.7%|52.7%

Table 6: Downstream comparison between the SimCLR, MoCo-V2, SimSiam, and their Representation evaluation (RED-) counterparts.

	IMAGENET 1000 KNN|LINEAR
MOCO-V2	44.9%|67.5%
RED-MOCO-V2	54.4%|68.0%
SIMSIAM	52.3%|68.1%
RED-SIMSIAM	53.4%|68.3%

Table 7: Downstream classification comparison in ImageNet1000.

6.1 Downstream Task Performance

We evaluated the classification accuracy of SimCLR, MoCo-V2, SimSiam, and their corresponding representation evaluation design models, RED-SimCLR, RED-MoCo-V2, and RED-SimSiam, on downstream classification tasks. Table 6 presents the classification accuracy of different SSL models on CIFAR-10 and CIFAR-100 data using k-nearest neighbors (kNN) and linear classifiers as downstream classifiers [Wu *et al.*, 2018]. Additionally, we compared the downstream classification performance on ImageNet1000 in Table 7 and mixed-Gaussian simulated data in Appendix C. We list the training settings in AppendixB.3. Our results demonstrate that RED consistently improves the downstream task performance of popular SSL models, regardless of whether they are contrastive (e.g., SimCLR, MoCo-V2) or not (e.g., SimSiam).

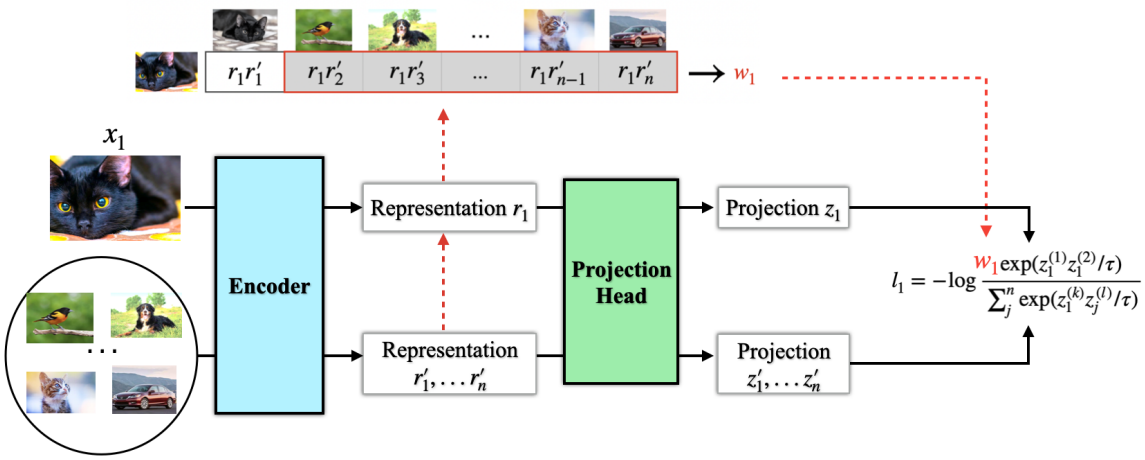


Figure 3: RED-SimCLR demonstration.

6.2 Robustness to (Unseen) Augmentations

In Section 4.1, we discussed how representation vectors are more robust to unseen augmentations. By introducing re-weights w_i that establish a shortcut connection between the representation and projection vectors, we expect RED-SSL models to be more robust to diverse (unseen) augmentations and verify it in this subsection.

We used MoCo-V2 as the base model and compared its performance with RED-MoCo-V2. After training both models, we applied (unseen) augmentations to the test samples and evaluated the kNN classification accuracy of the augmented test data. The results are reported in Table 8 and training settings in Appendix B.4. Compared to the original MoCo-V2, RED-MoCo-V2 consistently exhibits stronger robustness to diverse augmentations, whether they are specified during training (train augmentation) or unseen. These results demonstrate that RED-SSL captures more semantic information that is invariant to augmentations.

	MoCo-V2	RED-MoCo-V2
Train augmentation	77.27%	79.58%
Angle rotate	36.20%	38.61%
Gaussian Blur	78.41%	80.60%
Center Crop	15.41%	18.83%
Random Posterize	46.68%	47.57%

Table 8: Classification accuracy on augmented data. The "Train Augmentation" row stands for the augmentation types used during the pre-training. The SSL model is MoCo-V2 trained for 200 epochs in CIFAR-10 with ResNet-18 as the encoder.

6.3 Other Experiments

SSL methods could suffer from out-of-distribution shift [Hu *et al.*, 2022]. The results in Table 9 indicates that RED can improve the out-of-distribution generalization and training details in Appendix B.5. The ablation study in Appendix B.6 verifies that RED can steadily boost downstream performance.

	CIFAR-10 kNN|linear	CIFAR-100 kNN|linear
MoCo-V2	-	48.05%|55.46%
RED-MoCo-V2	-	48.87%|57.27%
MoCo-V2	73.39%|79.39%	-
RED-MoCo-V2	74.29%|79.86%	-

Table 9: Transfer learning. In the first and second rows, the models are trained with CIFAR-10, and in the third and fourth rows, the models are trained with CIFAR-100.

7 Discussion

This paper presents a comprehensive analysis of the projection head design in SSL, revealing that it promotes uniformity and allows the encoder to focus on alignment. This explains why combining the encoder and projection head outperforms using the encoder alone. The projection head also ensures non-collapsed training in non-contrastive models like SimSiam. The encoder & projection head combination enables the encoder to spotlight boosting alignment without worrying about uniformity; thus, the representation vectors enjoy more robustness to augmentation, lower entropy, and better downstream task performance than the projection vectors. Drawing on these insights, we introduce the Representation Evaluation Design (RED), an adaptable approach to diverse SSL models. Our experiments demonstrate that RED-SSL outperforms baseline models in downstream tasks and exhibits greater robustness to augmentation and out-of-distribution data. Our research sheds light on SSL model structure and can inspire further research in this area.

We think our work can be extended in several ways. First, it will be meaningful to explore ways to further simplify the RED while preserving its effectiveness. Second, exploring the performance of RED on different architectures, such as NLP: SimCSE [2021] or multi-modalities architecture: CLIP [Radford *et al.*, 2021], is an interesting topic for future exploration. We hope our work can inspire more effective and generalizable SSL methods.

References

[Arora *et al.*, 2019] Sanjeev Arora, Hrishikesh Khandeparkar, Mikhail Khodak, Orestis Plevrakis, and Nikunj Saunshi. A theoretical analysis of contrastive unsupervised representation learning. *arXiv preprint arXiv:1902.09229*, 2019.

[Beirlant *et al.*, 1997] 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.

[Caron *et al.*, 2020] 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.

[Caron *et al.*, 2021] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[Cazenavette *et al.*, 2021] George Cazenavette, Calvin Murdock, and Simon Lucey. Architectural adversarial robustness: The case for deep pursuit. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7150–7158, 2021.

[Chen and He, 2021] 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.

[Chen *et al.*,] X Chen, S Xie, and K He. An empirical study of training self-supervised vision transformers. in 2021 ieee. In *CVF International Conference on Computer Vision (ICCV)*, pages 9620–9629.

[Chen *et al.*, 2020a] 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.

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

[Deng *et al.*, 2009] 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. Ieee, 2009.

[Dosovitskiy *et al.*, 2020] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[Gao *et al.*, 2021] Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*, 2021.

[Grill *et al.*, 2020] 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.

[Gupta *et al.*, 2022] Kartik Gupta, Thalaiyasingam Ajanthan, Anton van den Hengel, and Stephen Gould. Understanding and improving the role of projection head in self-supervised learning. *arXiv preprint arXiv:2212.11491*, 2022.

[HaoChen *et al.*, 2021] 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*, 34:5000–5011, 2021.

[He *et al.*, 2016] 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.

[He *et al.*, 2020] 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.

[He *et al.*, 2022] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022.

[Hu *et al.*, 2022] Tianyang Hu, Zhili Liu, Fengwei Zhou, Wenjia Wang, and Weiran Huang. Your contrastive learning is secretly doing stochastic neighbor embedding. *arXiv preprint arXiv:2205.14814*, 2022.

[Ji *et al.*, 2021] Wenlong Ji, Zhun Deng, Ryumei Nakada, James Zou, and Linjun Zhang. The power of contrast for feature learning: A theoretical analysis. *arXiv preprint arXiv:2110.02473*, 2021.

[Jing *et al.*, 2021] Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. *arXiv preprint arXiv:2110.09348*, 2021.

[Khosla *et al.*, 2020] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.

[Krizhevsky, 2009] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[Li *et al.*, 2020] Junnan Li, Pan Zhou, Caiming Xiong, and Steven CH Hoi. Prototypical contrastive learning of unsu-

pervised representations. *arXiv preprint arXiv:2005.04966*, 2020.

[Li *et al.*, 2022] Alexander C Li, Alexei A Efros, and Deepak Pathak. Understanding collapse in non-contrastive siamese representation learning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXI*, pages 490–505. Springer, 2022.

[Lombardi and Pant, 2016] Damiano Lombardi and Sanjay Pant. Nonparametric k-nearest-neighbor entropy estimator. *Physical Review E*, 93(1):013310, 2016.

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

[Purushwalkam and Gupta, 2020] Senthil Purushwalkam and Abhinav Gupta. Demystifying contrastive self-supervised learning: Invariances, augmentations and dataset biases. *Advances in Neural Information Processing Systems*, 33:3407–3418, 2020.

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

[Reshniak and Webster, 2020] Viktor Reshniak and Clayton G Webster. Robust learning with implicit residual networks. *Machine Learning and Knowledge Extraction*, 3(1):34–55, 2020.

[Saunshi *et al.*, 2022] Nikunj Saunshi, Jordan Ash, Surbhi Goel, Dipendra Misra, Cyril Zhang, Sanjeev Arora, Sham Kakade, and Akshay Krishnamurthy. Understanding contrastive learning requires incorporating inductive biases. *arXiv preprint arXiv:2202.14037*, 2022.

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

[Tian *et al.*, 2021] 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.

[Tian, 2022] Yuandong Tian. Understanding the role of nonlinearity in training dynamics of contrastive learning. *arXiv preprint arXiv:2206.01342*, 2022.

[Tosh *et al.*, 2021] Christopher Tosh, Akshay Krishnamurthy, and Daniel Hsu. Contrastive learning, multi-view redundancy, and linear models. In *Algorithmic Learning Theory*, pages 1179–1206. PMLR, 2021.

[Von Kügelgen *et al.*, 2021] Julius Von Kügelgen, Yash Sharma, Luigi Gresele, Wieland Brendel, Bernhard Schölkopf, Michel Besserve, and Francesco Locatello. Self-supervised learning with data augmentations provably isolates content from style. *Advances in neural information processing systems*, 34:16451–16467, 2021.

[Wang and Isola, 2020] 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.

[Wang and Liu, 2021] Feng Wang and Huaping Liu. Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2495–2504, 2021.

[Wang *et al.*, 2021] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3024–3033, 2021.

[Wang *et al.*, 2022a] Yifei Wang, Qi Zhang, Yisen Wang, Jiansheng Yang, and Zhouchen Lin. Chaos is a ladder: A new theoretical understanding of contrastive learning via augmentation overlap. *arXiv preprint arXiv:2203.13457*, 2022.

[Wang *et al.*, 2022b] Yizhou Wang, Shixiang Tang, Feng Zhu, Lei Bai, Rui Zhao, Donglian Qi, and Wanli Ouyang. Revisiting the transferability of supervised pretraining: an mlp perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9183–9193, 2022.

[Wang *et al.*, 2022c] Zhaoqing Wang, Qiang Li, Guoxin Zhang, Pengfei Wan, Wen Zheng, Nannan Wang, Mingming Gong, and Tongliang Liu. Exploring set similarity for dense self-supervised representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16590–16599, 2022.

[Wen and Li, 2021] Zixin Wen and Yuanzhi Li. Toward understanding the feature learning process of self-supervised contrastive learning. In *International Conference on Machine Learning*, pages 11112–11122. PMLR, 2021.

[Wen and Li, 2022] Zixin Wen and Yuanzhi Li. The mechanism of prediction head in non-contrastive self-supervised learning. *arXiv preprint arXiv:2205.06226*, 2022.

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

[Yeh *et al.*, 2022] Chun-Hsiao Yeh, Cheng-Yao Hong, Yen-Chi Hsu, Tyng-Luh Liu, Yubei Chen, and Yann LeCun. Decoupled contrastive learning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVI*, pages 668–684. Springer, 2022.

[Zbontar *et al.*, 2021] 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.