ECAI 2023
K. Gal et al. (Eds.)

doi:10.3233/FAIA230339

Generalizing Similarity in Noisy Setups: The DIBS Phenomenon

Nayara Fonseca[*,a] **and Veronica Guidetti**[*,b]

[a]Rudolf Peierls Centre for Theoretical Physics, University of Oxford, Oxford, UK
[b]UNIMORE, FIM Department, Via G. Campi 213/a, 41125, Modena, Italy

Abstract. This work uncovers an interplay among data density, noise, and the generalization ability in similarity learning. We consider Siamese Neural Networks (SNNs), which are the basic form of contrastive learning, and explore two types of noise that can impact SNNs, Pair Label Noise (PLN) and Single Label Noise (SLN). Our investigation reveals that SNNs exhibit double descent behaviour regardless of the training setup and that it is further exacerbated by noise. We demonstrate that the density of data pairs is crucial for generalization. When SNNs are trained on sparse datasets with the same amount of PLN or SLN, they exhibit comparable generalization properties. However, when using dense datasets, PLN cases generalize worse than SLN ones in the overparametrized region, leading to a phenomenon we call Density-Induced Break of Similarity (DIBS). In this regime, PLN similarity violation becomes macroscopical, corrupting the dataset to the point where complete interpolation cannot be achieved, regardless of the number of model parameters. Our analysis also delves into the correspondence between online optimization and offline generalization in similarity learning. The results show that this equivalence fails in the presence of label noise in all the scenarios considered.

1 Introduction

In recent years, several works have studied generalization in neural networks (NNs) and the connection between the classical underparametrized regime, where the number of training samples is larger than the number of parameters in the model, and that of deep learning, where the opposite is usually the norm. Indeed, the empirical success of overparametrized NNs challenges conventional wisdom in classical statistical learning. It is widely known among practitioners that larger models (with more parameters) often obtain better generalization performance [49, 23, 44].

Two frameworks adopted to study generalization in regression or classification tasks are *Double Descent* (DD) and *online/offline learning correspondence*, which we describe in the following. The DD from [3] connects "classical" and "modern" machine learning by observing that once the model complexity is large enough to interpolate the dataset (i.e., when the training error reaches zero), the test error decreases again, reducing the generalization gap. This pattern has been empirically observed for several models and datasets, ranging from linear models, as in [33], to modern deep neural networks, as in [48, 36]. Instead, the online/offline learning correspondence of [38] studies the relationship between online optimization and offline generalization. The conjecture, empirically verified on supervised image classification, states that generalization in an offline setting can be effectively reduced to an optimization problem in the infinite-data limit. This means that online and offline test errors coincide if the NN is trained for a fixed number of weight updates. This setup aims to *link* the under- and overparametrized models: the infinite-data limit (online) sits in the under-parameterized region (number of samples $>$ number of parameters), while the finite-data case (offline) corresponds to the overparametrized regime (number of samples $<$ number of parameters). Here, we test if this correspondence is also valid for similarity tasks.

DD phenomenon and online/offline correspondence are two complementary approaches that look at different generalization properties: while the DD analysis studies how the network adjusts to an increasing number of parameters, the online/offline training compares the network performance by varying the dataset size while fixing the number of weight updates. Although these approaches have mainly been applied to classification and regression, if they are associated with some fundamental properties of deep neural networks, they should also hold for other tasks such as similarity learning.

There are key differences between similar-different discrimination and classification. For similarity learning, the relation among features is crucial, but not necessarily the features themselves. For this reason, a priori, it is not possible to predict whether the DD behavior and the online/offline learning correspondence will also occur in similarity problems. To take the first steps towards understanding how deep neural networks generalize in similarity learning, we export both frameworks to the simplest contrastive learning representative, Siamese Neural Networks (SNNs) from [7, 9]. A Siamese architecture is made of two identical networks sharing weights and biases that are simultaneously updated during supervised training. The two networks are connected by a final layer, which computes the distance between branch outputs. SNNs are trained using pairs of data that are labeled as similar or different. The task of a successfully trained network is to decide if the pair samples belong to the same class.

Studying the DD and online/offline correspondence in SNNs and comparing the results with those found in classification problems requires identifying which properties/characteristics of the training set most influence similarity learning. We identified two crucial sources of variability: *(i)* the effect of noisy data in SNNs, and *(ii)* the density of pairs in the training set. Noise is crucial in understanding generalization as it appears in every real-world dataset and may compro-

* Equal contribution. Emails: nayara.fonsecadesa@physics.ox.ac.uk; veronica.guidetti@unimore.it.

mise model performance. While DD was also studied in the presence of noise,[1] very little (if none) attention was devoted to noise in the online/offline setting. By construction, SNNs can be affected by more complex types of noise than classification problems. This derives from the use of pairwise relations defining a similarity graph. To show the reaction of SNNs to different noise sources, we introduce two representative examples with distinctive properties: Single Label Noise (SLN) and Pair Label Noise (PLN), which we describe in detail in Sec. 2. As we will show, SLN breaks similar/different pairs balancing but preserves similarity relations. Instead, PLN acts symmetrically on pair labels, but it breaks transitivity and, thus, similarity. See top of Fig. 1 for a pictorial view of SLN and PLN. Furthermore, we show that similarity learning is strongly influenced by the *density of pairs in the training set*. Concretely, we demonstrate how pairs created from populations with different levels of similarity graph density (or image diversity) give rise to very different learning models. This means that the average number of different images appearing in a set of pairs can significantly impact the learning process and ultimately influence the performance of the model. We discuss sparse and dense connections in detail in Sec. 2

The results in this work are summarized as follows.

- DD clearly appears in SNNs, regardless of the noise level, a phenomenon rarely found in classification problems in the absence of noise.
- DD is exacerbated by noise (in line with [36]), and its shape is affected by the density of pairs in the training set. While SNNs trained on sparse datasets show similar DD curves in the presence of SLN and PLN, these become quite distinct when the similarity relations in the training set are dense. Specifically, the interpolation threshold in the presence of PLN requires more parameters, and its test error remains higher in the overparametrized region. An example of this behavior is shown at the bottom of Fig. 1.
- We show that complete interpolation (training error $= 0$) cannot be achieved in the PLN scenario with dense connections and derive an upper and lower bound for the asymptotic training error value in the deep overparametrized regime. We call this phenomenon *Density-Induced Break of Similarity (DIBS)*.
- We test the correspondence between offline generalization and online optimization for similarity learning. We study how the architecture and the presence of noisy labels can differently impact these two regimes. We find that the conjecture only holds for clean data.
- In the presence of label noise, we find that the online/offline correspondence breaks down for all choices of training settings considered. In particular, the effect of label noise is notably more relevant in the offline case.

1.1 Related work

Over the past few years, significant strides have been made to understand how neural networks generalize in the presence of noise in classification problems (e.g., [31, 19, 1, 20, 47]). Remarkably, the DD phenomenon enabled a closer examination of the NN behavior as the number of trainable parameters, the evolution time, and the size of the dataset vary [36, 4, 22, 42]. Subsequently, other works have produced analytical studies of some of these phenomena [13, 14, 34]. Another complementary tool used to study generalization in classification tasks is the online/offline correspondence proposed in [38], which focuses on datasets without noise. This study empirically showed a correspondence between online optimization and offline generalization for modern deep NNs trained to classify images. Earlier studies have proposed a similar comparison for linear models focusing on the asymptotic regime of training (see, e.g, [5, 6]).

Contrastive learning, introduced by [9, 18, 40], has become one of the most prominent supervised [27, 17] and self-supervised [2, 50, 21, 8] ML techniques to learn similarity relations of high-dimensional data, producing impressive results in several fields, see, e.g. [29, 25]. Despite its success, [39, 32, 29] show that contrastive learning usually requires huge datasets and considerable use of data augmentation techniques. Dealing with augmentation techniques and unlabeled data where negative samples are randomly selected introduces instance discrimination challenges, i.e., the need to find ways to limit the appearance of faulty positive and negative samples. Indeed, [46, 51] show that the contrastive loss does not always sufficiently guide which features are extracted. For these reasons, several works tackled the problem of discriminating against faulty negatives, as [24, 26, 10], removing faulty positives and negatives dynamically (see [46, 53]) and creating more robust contrastive setups introducing new losses (see [11, 35]) or architectural components, [16].

2 Dataset construction

In this section, we describe the choices we made to study the dataset features that influence training and generalization in similarity learning, i.e., the density of the image pairs and the presence of noise. We start by defining the criteria used to construct the pairs.

Similarity graph. As opposed to classification problems, where the main concerns during dataset creation are class balancing and image diversity, in contrastive learning, we should consider that pair (or group) relations between images define an unoriented similarity graph inside the input space. Calling N the total number of images in the full dataset and N_{pairs} the number of pairs, the density of this graph, $\rho = N_{\text{pairs}}/\binom{N}{2}$, tells us the extent of knowledge we have about the input images. To enhance our understanding of a given dataset, we ought to create all possible labeled pairs, $\binom{N}{2} \sim N^2$, but this quickly becomes unfeasible when considering large datasets. For this reason, we construct pairs in a way that maximizes the information about similar images (all similar images are transitivity-related) and scales linearly with N. In practice, we construct closed chains of positive pairs within the same class, c, $\{\{x_1^c, x_2^c\}, \ldots, \{x_k^c, x_{k+1}^c\}, \ldots, \{x_n^c, x_1^c\}\}$, where n is the total number of images in c. Then, to build negative pairs, each image is connected to the element having the same index in a different class, chosen at random $\{x_k^c, x_k^{c'}\}$ where $c' \neq c$. If the original dataset classes are balanced, each image appears on average in 4 different pairs (2 times in the positive and 2 times in the negative pairs). Therefore, the total number of pairs is given by $N_{\text{pairs}} = 2 \times N = 2 \times N_c \times n_c$, where n_c is the total number of classes, and N_c is the number of elements per class. Finally, we describe the dataset construction method we used to study how density in the similarity graph affects training.

- **Scenario 1: sparse connections.** To train the network in the absence of noise, we first create the pairs using the full dataset. We follow the procedure described at the beginning of this section so that $N_{\text{pairs}} = 2 \times N$. We then take N_{sample} balanced pairs (data

[1] Notably, it is known that the DD curve is exacerbated in the presence of random label noise in supervised classification (see, e.g., [36]).

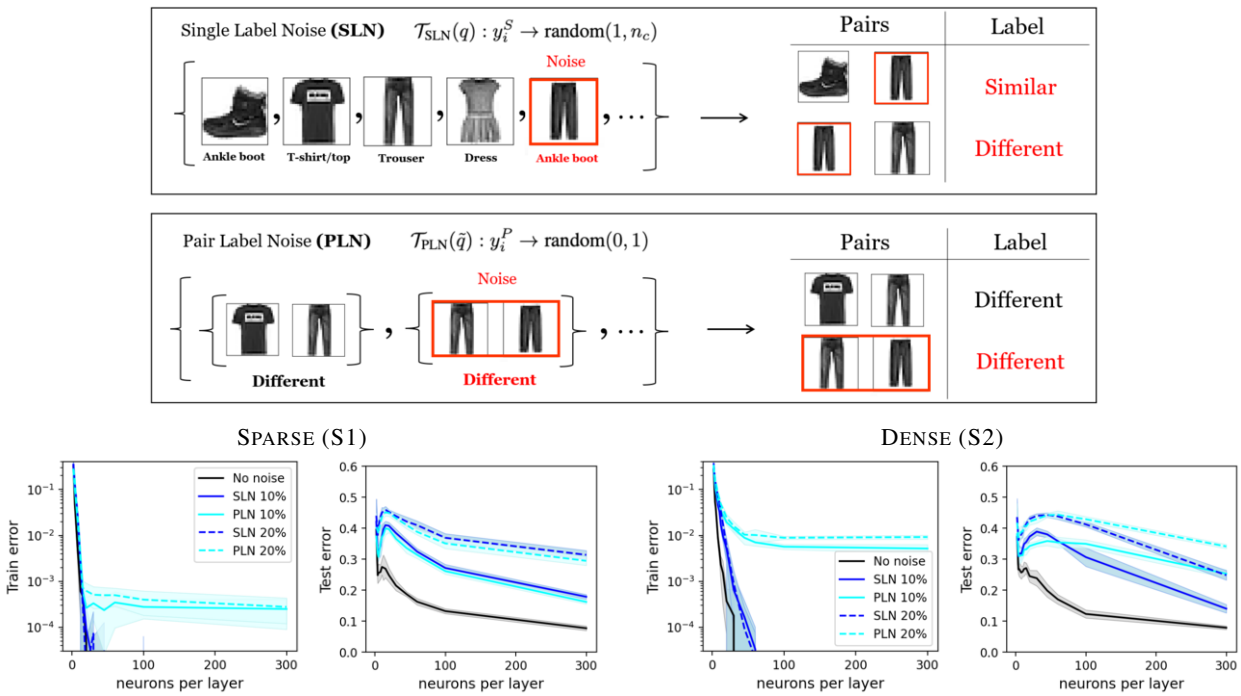

Figure 1: **Top:** Illustration of SLN and PLN applied to image samples and the resulting dataset of pairs. As discussed in the text, PLN leads to inconsistent relations in the dataset. This effect becomes more apparent when using dense datasets. On the other hand, similarity breaking does not appear in SLN, where the similarity relations may go against image features but are self-consistent. **Bottom:** Train and test errors as a function of model size for sparse (S1) and dense (S2) configurations. We consider a 3-layer MLP with ReLU activation functions trained on sparse and dense pairs of MNIST with 10% and 20% effective noise (see Sec. 2.1 for details). Note that both no-noise and SLN cases reach complete interpolation in the training set, while PLN train error no longer vanishes by increasing the number of network parameters.

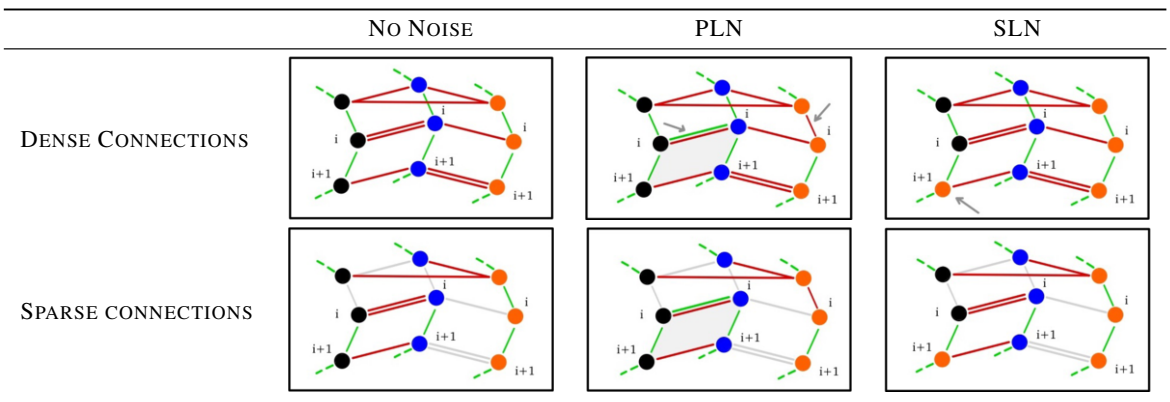

Figure 2: Pictorial view of data relation appearing in Scenario 2 (top) and 1 (bottom) for different classes of data (black, blue, and orange). Each image corresponds to a node in the similarity graph whose color is representative of its class. Positive (similar) pairs are connected by green edges, negative (different) pairs by red edges. Ignored connections are in light gray. Multiple edges between two nodes refer to repeated pairs. Gray arrows indicate where noise appears, shaded areas (PLN column) are examples of transitivity breaking (DIBS). As discussed in Sec. 2, we create closed chains of positive pairs within the same class c, while the negative pairs are formed by connecting each image in c to the element with the same index i in a different class c'.

used to train the model) from the $N_{\rm pairs}$ list to train the NN and repeat this procedure n_s times.

- **Scenario 2: dense connections.** In this setup, we create a reduced version of the original dataset. Being interested in training the network on $N_{\rm pairs}$ pairs, we select $N_{\rm reduced} = N_{\rm pairs}/2$ images from the original training set. The reduced dataset is balanced so that we have $N_{\rm pairs}/(2n_c)$ images per class. Then, we create our training and test samples using the same prescription described at the beginning of this section. We connect adjacent images within the same class, and each of them with a random image with the same index in a different class. This way, we get exactly $N_{\rm pairs}$ pairs that will be automatically balanced. We repeat this procedure n_s times.

Pictorial views of the similarity graph are shown in Fig. 2, where we represent elements belonging to different classes with nodes of different colors (black, blue, and orange classes), and similarity and

dissimilarity relations with green and red edges, respectively. We will show that the qualitative relation between generalization and dataset density is independent of the specific method used in pair construction. In short, the relevant quantity is the probability of finding closed paths in the similarity graph. However, the approach used in this work allows for dealing with the problem analytically, as shown in Sec. 3.

Noise introduction. SNNs can be subjected to different types of noise having different properties. To show their impact on the training process, we introduce two simple representatives, namely Single Label Noise (SLN) and Pair Label Noise (PLN), which are described below (see Fig. 2 for an illustration).

- **Single Label Noise (SLN).** Let us consider a dataset with N samples belonging to n_c classes and their corresponding labels $Y^S = \{y_1^S, y_2^S, \ldots, y_N^S\}$. Suppose the classes are uniformly populated $N_c = N/n_c$. If some label noise is present in the original dataset, this will propagate to the training pairs as these are created. If SLN is uniformly introduced across all classes, it will keep the original class balancing on average (over multiple samples). On the other hand, in every single run, statistical fluctuations of uniform distribution introduce some asymmetry in the original class representative number (see Fig. 2). Finally, in the presence of SLN, similarity relations (reflexive, symmetric, and transitive properties) are preserved as mislabeling appears in all pairs containing a misclassified image.
- **Pair Label Noise (PLN).** Let us now consider a dataset of N_{pairs} pairs with pair labels $Y^P = \{y_1^P, y_2^P, \ldots, y_{N_{\text{pairs}}}^P\}$, which can be similar ($y^P = 1$) or different ($y^P = 0$). We construct them so that they are balanced (half are similar, and half are different). Suppose we randomly shuffle some fraction of the total labels. In that case, the noise we introduce is symmetric under similar $\leftrightarrow$ different changes, and it acts democratically on every class of the original dataset. On the other hand, PLN can lead to inconsistent relations in the pairs dataset. Indeed, as we will show in the following sections, it breaks transitivity and, therefore, similarity.

As discussed later, these two sources of noise impact how models learn similarity relations in distinct ways. To fairly compare the outcome of the model in the presence of PLN and SLN, we need to ensure that we introduce the same amount of input label noise in the two setups. We present below how we ensured that the same amount of *effective noise* was introduced. Being n_c the number of image classes, y_i^S the label of the i-th image, and y_i^P the label of the i-th pair of images, we can define the SLN transformation as

$$\mathcal{T}_{\text{SLN}}(q) : y_i^S \to \text{random}(1, n_c) \quad \text{with probability } q \tag{1}$$

and the PLN transformation as

$$\mathcal{T}_{\text{PLN}}(\tilde{q}) : y_i^P \to \text{random}(0, 1) \quad \text{with probability } \tilde{q}\,. \tag{2}$$

As SLN appears in the dataset before pair creation and the pairs are constructed so that the dataset is balanced (half pairs are similar, half are different), the probability of effective pair mislabeling induced by SLN, $P_{\text{SLN}}(q)$, is given by

$$P_{\text{SLN}}(q) = q - \frac{q^2}{2}\,. \tag{3}$$

while the probability of effective pair mislabeling coming from PLN, $P_{\text{PLN}}(\tilde{q})$, is

$$P_{\text{PLN}}(\tilde{q}) = \frac{\tilde{q}}{2}\,. \tag{4}$$

The requirement of having the same amount of effective noise in the dataset ($P_{\text{SLN}}(q) = P_{\text{PLN}}(\tilde{q})$) boils down to the following relation between q and $\tilde{q}$:

$$q = 1 - \sqrt{1 - \tilde{q}}. \tag{5}$$

The details of this derivation and the pseudocodes describing dataset creation can be found at https://arxiv.org/abs/2201.12803.

2.1 Experimental setup

In this work, we consider two Siamese branch architectures. The first one is an MLP with 3 hidden layers having the same width and ReLU activation functions with Xavier uniform initialization, see [15]. The second architecture is a 4-layer CNN. We also considered two training setups: in one case, we compute the Euclidean distance in the output layer training the network using Contrastive Loss from [18], and in the other one, we compute the cosine similarity training the network using Cosine Embedding Loss (see https://arxiv.org/abs/2201.12803 for details). The CNN architecture is based on the model described in [41], it contains three Convolution-BatchNormalization-ReLU-MaxPooling layers and a fully-connected output layer. The number of filters in each convolution layer scales as [k, $2k$, $2k$] while the MaxPooling is [1, 2, 8]. We fix the kernel size = 3, stride = 1 and padding = 1. When we train the network using contrastive loss (cosine embedding loss), we set the fully-connected output layer width to k ($2k$).

Double Descent (DD) setup. We test the presence of DD using MNIST [30], FMNIST [52] and CIFAR10 [28] datasets. To understand the impact of overparameterization, we study how training and test errors vary at increasing network width and training time. To do so, we increase the number of neurons per layer in the fully connected architecture and the parameter k in the CNN. For all datasets, we consider 6000 training and 9000 test pairs. In every DD experiment, we let the network evolve for 2000 epochs using Adam optimizer with minibatches of size 128 and learning rate $\lambda = 10^{-4}$, except when explicitly stated. All the hyper-parameters and the margins were chosen empirically. To see the average effect regardless of the particular choice of images in the dataset and weights initialization, we run 15 evolutions of the network using different training and test samples at each time. In most of the experiments, unless otherwise stated, we considered $\tilde{q} = 0.2$, i.e., an effective noise of 10%.

Online/offline setup. Since we cannot reuse samples for the online training, we consider an extended version of the standard MNIST dataset, namely the EMNIST (from [12]). We use the digit section of EMNIST that contains 240,000 training (and 40,000 test) 28 $\times$ 28 greyscale pixel images. We train the offline case (Real world) over 40 epochs using 12k pairs that are created considering sparse and dense scenarios. The online scenario (Ideal world) is trained once on 480k pairs created using the full training set of 240k samples. We test the models with 9k pairs constructed from the test set and consider Siamese networks with MLP and CNN blocks described at the beginning of this section. In order to compare the results on different network architectures, we used a comparable total number of parameters, namely, 200 nodes per layer for the MLP cases (total of 237,400 parameters) trained with the contrastive loss ($\lambda = 10^{-4}$); and width $k = 47$ (total of 235,611 parameters) for the CNN cases trained with the cosine loss ($\lambda = 5 \times 10^{-5}$). To provide an estimate of the results regardless of the particular choice of images and network initialization, we run MLP (CNN) experiments 5 (4) times. Each of the experiments mentioned above was performed in the presence and absence of noise and considering sparse (scenario 1) and dense pairs (scenario 2) in the training set.

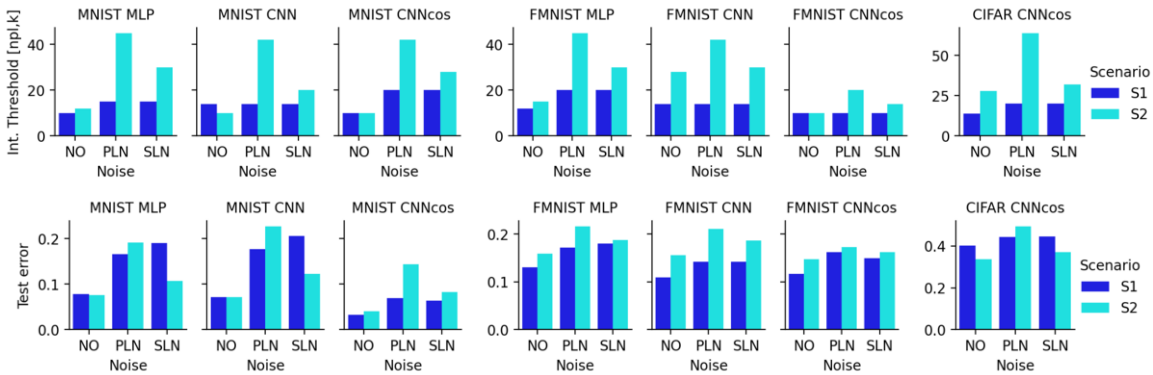

Figure 3: **Top:** Relation between training setup (see Sec. 2.1), noise source ('NO' refers to the scenario without noise), and interpolation threshold (DD peak location) expressed using number of neurons per layer (npl) for MLP setups or number of filters, k, for CNN cases. Note that PLN in S2 typically requires more parameters to interpolate. **Bottom:** Average test errors in the deep overparametrized regime after 2000 epochs. Due to the breaking of transitivity (DIBS phenomenon), PLN average test errors in S2 stay higher in the deep overparametrized regime.

3 Results

Double Descent (DD) results. In all experiments we see the DD behavior, regardless of architecture, loss function, scenario and noise. This does not happen in classification problems which typically require the presence of noise to make the DD curves clearly visible (see, e.g., [36]). As expected, DD becomes more prominent in the presence of noise. At the bottom of Fig. 1, we show how the network reacts to different amounts of noisy labels. In **Scenario 1**, the input dataset connections are sparse, and PLN and SLN have the same impact on training. This is understandable as there should not be any difference between PLN and SLN effects in the extreme case where every image appears only once in the training set. Instead, **Scenario 2** is characterized by dense input connections, and the system behaves differently under SLN and PLN. We experimentally observe that the DD peak location changes between PLN and SLN in almost all setups considered, see the rightmost plot at the bottom of Fig. 1 and the top of Fig. 3. Specifically, PLN peaks are shifted to the right-hand side, hinting that PLN is harder to interpolate than SLN as it requires more parameters. Increasing the amount of noise enhances the test errors as expected, but does not induce any significant peak shift.

SLN test error tends to be higher in small to medium network sizes, see Fig. 1. A hint about how this happens is given in Fig. 2. Indeed, SLN introduces a systematic error: a mislabeled image appears to be mislabeled in every pair. Therefore, given that the image features are not going to agree with pair labels, the only way the network has to classify correctly is by extracting the image from its natural distribution. NNs being continuous functions, this implies that a neighborhood of said image must be extracted as well, increasing the test error. At higher network widths, the volume of the mislabeled image neighborhood can become arbitrarily small, and the test error is free to go down again. In fact, *SLN introduces systematic errors that do not compromise the consistency of the similarity graph*. On the other hand, PLN stays higher in the deep overparametrized regime (see bottom plots in Fig. 3). Indeed, randomly changing similarity relations in the input dataset, *PLN ends up breaking transitivity, making the training set similarity graph inconsistent*. Beyond keeping test error high, this inconsistency also implies that the network is never able to overfit completely: the training error will no longer vanish just by increasing the number of network parameters, see e.g., train error plots at the bottom of Fig. 1. This effect is exacerbated when using dense datasets.

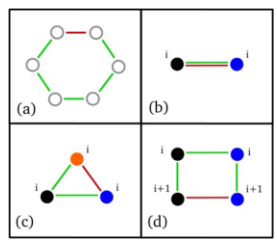

Figure 4: Similarity breaking configurations (a) and their leading contributions (b,c,d).

Origin and magnitude of DIBS. We now explain the origin of the phenomenon we call *Density-Induced Break of Similarity* originating from PLN. We can see if similarity is satisfied or violated in the training set by evaluating the consistency of the closed paths in the training pairs graph. Examples of inconsistent paths are depicted in Fig. 4. Similarity breaking on 2-paths, configuration (b), corresponds to symmetry breaking, while on $n-$paths, as in configurations (a,c,d), where $n > 2$, corresponds to transitivity breaking.

Theorem 1 *Let $\mathcal{D}$ be a dataset containing elements belonging to n_c classes, each having N_c elements. Let x_i^c denote the i-th element of the c-th class. Let $\mathcal{G}_S$ be the similarity graph induced by the creation of similar and dissimilar data pairs. Let the positive pairs be constructed as $\{x_i^c, x_{i+1}^c\}$, so as to form closed chains of similar pairs. Let the negative pairs be constructed as $\{x_i^c, x_i^{c'}\}$, $c' \neq c$, so as to generate random graphs of dissimilar pairs between elements of the same index. If the transformation $\mathcal{T}_{\rm PLN}(2P)$ is applied to the pairs thus created, it induces the break of similarity resulting in an asymptotic training error,*

$$\text{Error}_{\rm DIBS} = \lim_{n_\theta \to \infty} \text{TrainError}^{\rm PLN}_{\rm Dense}(P, n_c)\,, \tag{6}$$

that is limited by:

$$\frac{P\,(1-P)}{2(n_c-1)} \leq \text{Error}_{\rm DIBS} - \mathcal{E}_{\rm sim} < \mathcal{E}_{\rm diff} \tag{7}$$

where n_θ is the number of network parameters, P is the probability of effective pair mislabeling induced by PLN, $\mathcal{T}_{\rm PLN}$ is the transfor-

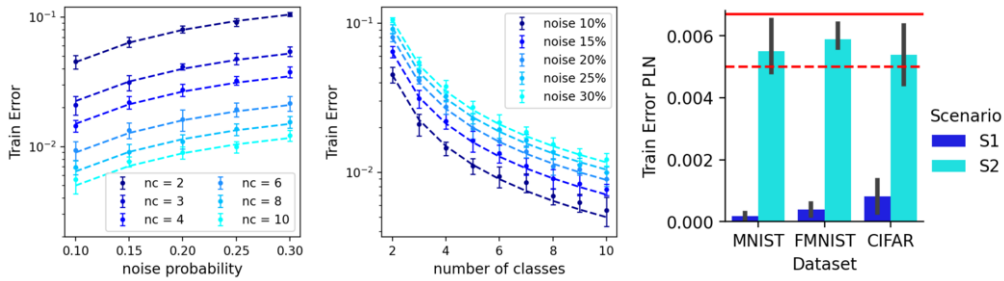

Figure 5: Analytic 1st order (dashed lines) and numerical (scatter points) estimates of the asymptotic training error behavior at varying number of classes n_c (**left**) and effective noise (**center**) in the presence of PLN in Scenario 2 (Dense) for the FMNIST dataset trained on the MLP architecture with 500 neurons per layer, using Euclidean distance and contrastive loss. **Right**: Comparison between experimental training error distributions and lower (dashed line) and upper (solid line) bounds of Theorem 1. The DIBS phenomenon is observed in all datasets considered.

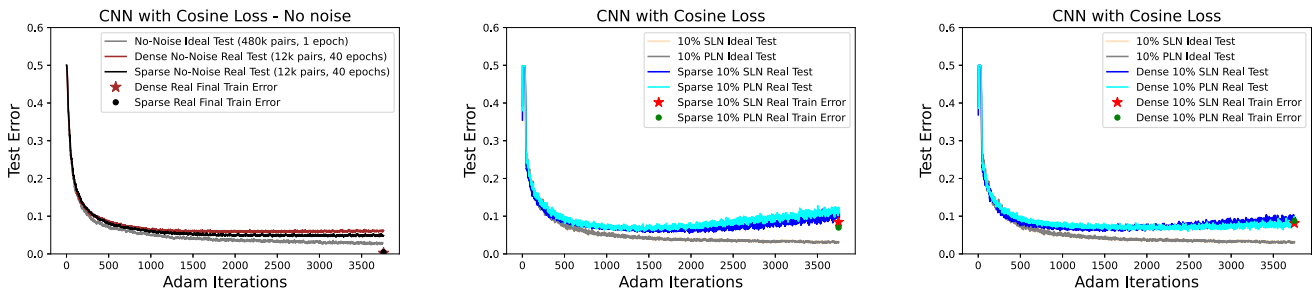

Figure 6: Ideal (online) *vs.* sparse/dense Real (offline) worlds for the CNN architecture with $k = 47$ trained with the cosine loss in the absence of noise (**left**) and with 10% of label noise for Scenario 1 (Sparse) (**center**) and Scenario 2 (Dense) (**right**). Plots show the median test errors as a function of minibatch Adam iterations. The stars (dots) correspond to the median real-world training errors at the end of training.

mation given in Eq. 2, and

$$\begin{aligned}\mathcal{E}_{\rm sim} &= \frac{P(1-P)^{N_c-1}}{2}\,,\\ \mathcal{E}_{\rm diff} &= \sum_{m=2}^{nc}\frac{m\,P^{m-1}(1-P)}{2^m(n_c-1)^{m-1}}\frac{(n_c-2)!}{(n_c-m)!}\\ &\times\sum_{i=0}^{N_c/2}\left[\frac{(1-P)}{2}\right]^{2i}.\end{aligned}\tag{8}$$

Corollary 1.1 *If the assumptions of Theorem 1 hold and* $n_c = 2$ *then:* $\mathrm{Error}_{\rm DIBS} - \mathcal{E}_{\rm sim} = \frac{P(1-P)}{2}$.

Below we briefly list the highlights of the proof of the theorem (and the corollary) to be analyzed to derive the above results. The upper bound in Theorem 1 can be proved by noticing that reflexivity and transitivity breaking can only appear in:

1. Closed chains of similar pairs containing only 1 mislabeled pair (as in Fig. 4, in configuration (a), when all nodes belong to the same class);
2. Random closed n-paths containing elements of different classes with the same index (as in configurations (b,c));
3. n-paths containing multiple classes and more elements of the same class (as in configurations (d)).

The lower bound in Theorem 1 is the first-order approximation of the upper bound coming from 2-path only. The inequality can be proved by showing that there is no one-to-one correspondence between inconsistent n-paths and the number of classification errors.

More graph inconsistencies can lead to a single error. To facilitate the counting of the unavoidable errors associated with a given configuration, one can resort to collapsed configurations, i.e., collapsing nodes connected by green vertices, and then counting the number of inconsistent 2-paths. In particular, the Corollary 1.1 follows from noticing that, if $n_c = 2$, the similarity-breaking contribution coming from point 3) is completely redundant with that of point 2). The formula associated with the lower bound, and thus to inconsistent 2-paths, is given by the probability of having two elements, with the same index and belonging to different classes, connected by a correctly classified different pair and a noisy one. This leads to:

$$\underbrace{\frac{N_{\rm diff.pairs}}{N_{\rm pairs}}}_{\text{noisy different pair}}\times P\overbrace{\frac{N_{\rm diff.pairs}}{N_{\rm pairs}}}^{\text{correct different pair}}\times(1-P)\qquad\underbrace{\frac{\overbrace{2}^{\text{\# configurations}}}{n_c-1}}_{\text{connected to same 2 classes}}$$

where $N_{\rm diff.pairs}$ is the number of different pairs, and $N_{\rm pairs}$ is the total number of pairs, with $\frac{N_{\rm diff.pairs}}{N_{\rm pairs}} = \frac{1}{2}$ as we consider balanced equal/different pairs. Finally, it is easy to see that the contribution coming from $\mathcal{E}_{\rm sim}$ is negligible in standard situations where $N_c \gg 1$. In Fig. 5, we validate our formula by comparing it with experimental results. In particular, in the left and central panels, we consider the FMNIST dataset trained on our MLP architecture with 500 neurons per layer, using Euclidean distance and contrastive loss (see Sec. 2.1). Numerical results (mean and standard error bar) come from 10 runs where we choose different random classes each time. These results show that, in the overparametrized regime, the training error follows the behaviors of solid lines given by the lower bound of Theorem 1.

The right panel of Fig. 5 shows the mean and standard deviation of the training errors in the overparametrized regime obtained in all our DD experiments in scenarios 1 and 2. Moreover, we compare them with the lower (red dashed line) and upper bound (solid red line) of Theorem 1 for 10 classes and $P = 0.1$. We find perfect agreement between experiments and theoretical results.

This analysis shows that the macroscopic presence of transitivity breaking is linked to the presence and number of closed paths in the similarity graph and therefore to the dataset density.

Online (Ideal world) *vs.* Offline (Real world) learning. We probe the correspondence between offline generalization and online optimization [38] for similarity tasks by studying how the training setting and the presence of noisy labels can impact these two regimes. Considering usual training settings (i.e., natural choices of architecture-loss function match), the conjecture holds for data without noise, regardless of the dataset density. See the left panel of Fig. 6 for the CNN architecture equipped with cosine loss in the absence of label noise (experimental details were given in Sec. 2.1). In the presence of noise, however, we find that the online/offline correspondence breaks down for all choices of training settings considered.

Two representative examples where the conjecture breaks are depicted in Fig. 6. There, we show the median test error values on dense and sparse datasets of real- and ideal-world scenarios with 10% of PLN and SLN trained using the CNN architecture. We compare offline and online settings after the same number of training iterations. We observe that while both ideal and real test errors are affected by noise, this effect is exacerbated in the real world scenarios. In fact, we observe that the introduction of "fresh" samples to the model, even if they possess noisy labels, enhances the model's diversity and ultimately improves its generalization. Note that the ideal world curves (gray and bisque) overlap with each other. Interestingly, we also find that the online/offline correspondence for similarity tasks is influenced by the network architecture and the loss function choice. Nevertheless, independently of the architecture-loss matching, the equivalence between online and offline settings breaks down in the presence of label noise for all the scenarios considered.

DIBS and modern contrastive learning. The similarity-breaking nature of PLN in dense datasets should not be underestimated as it may appear in widely employed settings. Modern approaches to self-supervised contrastive learning (see the recent reviews of [39, 32, 25, 29]) heavily rely on data augmentation to learn representations [50]. The massive use of data augmentation, however, may result in partial representation learning (feature suppression) or lead to semantic errors as in [43]. Moreover, as exposed in [24], if negative pairs are formed by sampling views from different images, regardless of their semantic information, this may lead to the appearance of false-negative pairs, potentially breaking transitivity and compromising the training efficiency. Interestingly, this skewness towards false-negative pairs is the same effect we find studying the asymptotic training error balance with DIBS. Notwithstanding these issues, data augmentation and random selection of negative samples are intrinsic to self-supervised methods.[2] Therefore, several works in contrastive learning have focused on controlling the quality of augmented data and mitigating the effects of false negatives (see Sec. 1.1). When two different images belonging to the same class (sharing semantic features) are classified as different, convergence slows down and semantic information gets lost. This goes under the name of instance discrimination task (i.e., the problem of discriminating pairs of similar points from dissimilar ones), and failing it can harm the formation of features useful for downstream tasks. For this reason, feature extraction in self-supervised contrastive learning is usually affected by pair-label noise by construction.

[2] For example, in a pretext task, the original image acts as an anchor, its augmentations act as positive samples, and the rest of the images in the batch (or in the training data) act as negative samples.

4 Discussion and conclusions

We move the first steps towards understanding generalization in similarity learning focusing on SNNs. To do so, we borrow the frameworks of DD and online/offline correspondence from classification tasks. We show that DD appearance is magnified in SNNs as it appears also in the absence of noise. Notably, we find that noise and the density of pairs in the training set crucially affect generalization. We present two kinds of noise: SLN, preserving similarity relations, and PLN, breaking transitivity. The same noise sources presented in this work can be easily generalized to models where the network input is given by multiple images. Studying DD, we show that similarity-breaking noise compromises the asymptotic generalization performance (large training time) of the network in the overparametrized regime. Moreover, these effects get magnified at increasing training set density, preventing perfect interpolation. Studying the online/offline correspondence, we find that the generalization properties before overfitting time are not sensitive to the density of the training set and only depend on noise. In particular, in the presence of noise, the online/offline correspondence breaks down and the differences between the real and ideal generalization gap are not universal and depend on the training setup.

Limitations. This is an exploratory work that does not investigate all possible setups which may affect or lead to DD, such as regularization (see [37, 34]), epoch and sample-wise DD (see, [36, 4, 22, 42]). Moreover, we focus on the under- and over-parametrized regime without providing quantitative results about the interpolation threshold itself, [13, 14, 34]. This is because, to the best of our knowledge, there is no predefined way of treating SNNs analytically as no proxy model as Random Fourier Features (see [45]) can be constructed. Indeed, while in classification or regression tasks the output layer size is known, this is not true for SNNs. For this reason, we believe that an analytic study of DD in SNNs may require another approach, and we leave this study for future work.

Outlook. In the majority of modern contrastive learning works, the final graph of similarity relations in the dataset becomes really dense as each training step involves multiple images at a time. Moreover, from instance discrimination task examples, we know that contrastive learning tends to be affected by faulty positive and negative pair relations. This is the setting where we find that noise crucially impacts generalization. While the technological developments and the applications of contrastive learning kept expanding during the last years, a fundamental study about how it generalises and reacts to noise is still missing.

Acknowledgements

NF acknowledges the UKRI support through the Horizon Europe guarantee Marie Skłodowska-Curie grant (EP/X036820/1). This research was supported in part through the Maxwell computational resources operated at Deutsches Elektronen-Synchrotron DESY, Hamburg, Germany. This work was partially supported by the *CoSubmitting Summer (CSS) program at ICLR 2022*. We thank Preetum Nakkiran for useful discussions and for proposing the idea of probing the online/offline correspondence in Siamese networks.

References

[1] Eric Arazo et al., 'Unsupervised label noise modeling and loss correction', in *36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*.

[2] Philip Bachman et al., 'Learning representations by maximizing mutual information across views', *arXiv preprint arXiv:1906.00910*, (2019).

[3] Mikhail Belkin and othersw, 'Reconciling modern machine-learning practice and the classical bias-variance trade-off', in *Proceedings of the National Academy of Sciences of the United States of America vol. 116,32 (2019): 15849-15854*, (2019).

[4] Antoine Bodin and Nicolas Macris, 'Model, sample, and epoch-wise descents: exact solution of gradient flow in the random feature model', *Advances in Neural Information Processing Systems*, **34**, (2021).

[5] Léon Bottou and Yann LeCun, 'Large scale online learning', in *Advances in Neural Information Processing Systems 16 (NIPS 2003)*, eds., Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, MIT Press, Cambridge, MA, (2004).

[6] Léon Bottou and Yann LeCun, 'On-line learning for very large datasets', *Applied Stochastic Models in Business and Industry*, **21**(2), 137–151, (2005).

[7] Jane Bromley et al., 'Signature verification using a "siamese" time delay neural network', in *Advances in Neural Information Processing Systems*, volume 6. Morgan-Kaufmann, (1994).

[8] Ting Chen et al., 'A simple framework for contrastive learning of visual representations', in *International conference on machine learning*, pp. 1597–1607. PMLR, (2020).

[9] S. Chopra et al., 'Learning a similarity metric discriminatively, with application to face verification', in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pp. 539–546 vol. 1, (2005).

[10] Ching-Yao Chuang et al., 'Debiased contrastive learning', *Advances in neural information processing systems*, **33**, 8765–8775, (2020).

[11] Ching-Yao Chuang et al., 'Robust contrastive learning against noisy views', *arXiv preprint arXiv:2201.04309*, (2022).

[12] G. Cohen, S. Afshar, J. Tapson, and A. van Schaikf, 'EMNIST: an extension of MNIST to handwritten letters.', (2017).

[13] Stéphane d'Ascoli et al., 'Double trouble in double descent: Bias and variance (s) in the lazy regime', 2280–2290, (2020).

[14] Stéphane d'Ascoli et al., 'Triple descent and the two kinds of overfitting: Where & why do they appear?', *Advances in Neural Information Processing Systems*, **33**, 3058–3069, (2020).

[15] Xavier Glorot and Yoshua Bengio, 'Understanding the difficulty of training deep feedforward neural networks', in *Proceedings of the 13th international conference on artificial intelligence and statistics*, pp. 249–256. JMLR Workshop and Conference Proceedings, (2010).

[16] Jean-Bastien Grill et al., 'Bootstrap your own latent-a new approach to self-supervised learning', *Advances in Neural Information Processing Systems*, **33**, 21271–21284, (2020).

[17] Beliz Gunel et al., 'Supervised contrastive learning for pre-trained language model fine-tuning', *arXiv preprint arXiv:2011.01403*, (2020).

[18] R. Hadsell et al., 'Dimensionality reduction by learning an invariant mapping', in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pp. 1735–1742, (2006).

[19] Jiangfan Han et al., 'Deep self-learning from noisy labels', in *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, October 27 - November 2, 2019*.

[20] Hrayr Harutyunyan et al., 'Improving generalization by controlling label-noise information in neural network weights', in *37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020*, Proceedings of Machine Learning Research.

[21] Kaiming He et al., 'Momentum contrast for unsupervised visual representation learning', in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, (2020).

[22] Reinhard Heckel and Fatih Furkan Yilmaz, 'Early stopping in deep networks: Double descent and how to eliminate it', *arXiv preprint arXiv:2007.10099*, (2020).

[23] Yanping Huang et al., 'Gpipe: Efficient training of giant neural networks using pipeline parallelism', *Advances in neural information processing systems*, **32**, 103–112, (2019).

[24] Tri Huynh et al., 'Boosting contrastive self-supervised learning with false negative cancellation', in *IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2785–2795, (2022).

[25] Ashish Jaiswal et al., 'A survey on contrastive self-supervised learning', *Technologies*, **9**(1), 2, (2021).

[26] Yannis Kalantidis et al., 'Hard negative mixing for contrastive learning', *Advances in Neural Information Processing Systems*, **33**, 21798–21809, (2020).

[27] Prannay Khosla et al., 'Supervised contrastive learning', *Advances in Neural Information Processing Systems*, **33**, 18661–18673, (2020).

[28] Alex Krizhevsky et al., 'Learning multiple layers of features from tiny images', (2009).

[29] Phuc H Le-Khac et al., 'Contrastive representation learning: A framework and review', *IEEE Access*, **8**, 193907–193934, (2020).

[30] Yann LeCun et al., 'Mnist handwritten digit database', *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, **2**, (2010).

[31] Junnan Li et al., 'Learning to learn from noisy labeled data', in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5051–5059. Computer Vision Foundation / IEEE, (2019).

[32] Xiao Liu et al., 'Self-supervised learning: Generative or contrastive', *IEEE Transactions on Knowledge and Data Engineering*, (2021).

[33] Marco Loog et al., 'A brief prehistory of double descent', *Proceedings of the National Academy of Sciences*, **117**(20), 10625–10626, (2020).

[34] Song Mei and Andrea Montanari, 'The generalization error of random features regression: Precise asymptotics and the double descent curve', *Communications on Pure and Applied Mathematics*, **75**(4), 667–766, (2022).

[35] Pedro Morgado et al., 'Robust audio-visual instance discrimination', in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12934–12945, (2021).

[36] Preetum Nakkiran et al., 'Deep double descent: Where bigger models and more data hurt', in *International Conference on Learning Representations*, (2020).

[37] Preetum Nakkiran et al., 'Optimal regularization can mitigate double descent', *arXiv preprint arXiv:2003.01897*, (2020).

[38] Preetum Nakkiran, Behnam Neyshabur, and Hanie Sedghi, 'The deep bootstrap framework: Good online learners are good offline generalizers', in *9th International Conference on Learning Representations, ICLR 2021, Austria, May 3-7, 2021*.

[39] Kriti Ohri and Mukesh Kumar, 'Review on self-supervised image recognition using deep neural networks', *Knowledge-Based Systems*, **224**, 107090, (2021).

[40] Aaron Oord et al., 'Representation learning with contrastive predictive coding', *arXiv preprint arXiv:1807.03748*, (2018).

[41] David Page, 'How to train your resnet', (2018).

[42] Mohammad Pezeshki et al., 'Multi-scale feature learning dynamics: Insights for double descent', *arXiv preprint arXiv:2112.03215*, (2021).

[43] 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).

[44] Alec Radford et al., 'Language models are unsupervised multitask learners', *OpenAI blog*, **1**(8), 9, (2019).

[45] Ali Rahimi and Benjamin Recht, 'Random features for large-scale kernel machines', *Advances in neural information processing systems*, **20**, (2007).

[46] Joshua Robinson et al., 'Can contrastive learning avoid shortcut solutions?', *Advances in Neural Information Processing Systems*, **34**, (2021).

[47] Hwanjun Song et al., 'Learning from noisy labels with deep neural networks: A survey', *CoRR*, **abs/2007.08199**, (2020).

[48] Stefano Spigler et al., 'A jamming transition from under-to over-parametrization affects generalization in deep learning', *Journal of Physics A: Mathematical and Theoretical*, **52**(47), 474001, (2019).

[49] Christian Szegedy et al., 'Going deeper with convolutions', in *IEEE conference on computer vision and pattern recognition*, pp. 1–9, (2015).

[50] Yonglong Tian et al., 'Contrastive multiview coding', in *European conference on computer vision*, pp. 776–794. Springer, (2020).

[51] Feng Wang and Huaping Liu, 'Understanding the behaviour of contrastive loss', in *IEEE/CVF conference on computer vision and pattern recognition*, pp. 2495–2504, (2021).

[52] Han Xiao et al., 'Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms', (2017).

[53] Rui Zhu et al., 'Improving contrastive learning by visualizing feature transformation', in *IEEE/CVF International Conference on Computer Vision*, pp. 10306–10315, (2021).