SIMILARITY AND GENERALIZATION: FROM NOISE TO CORRUPTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Contrastive learning aims to extract distinctive features from data by finding an embedding representation where similar samples are close to each other, and different ones are far apart. We study how NNs generalize the concept of similarity in the presence of noise, investigating two phenomena: Double Descent (DD) behavior and online/offline correspondence. While DD examines how the network adjusts to the dataset during a long training time or by increasing the number of parameters, online/offline correspondence compares the network performances varying the quality (diversity) of the dataset. We focus on the simplest contrastive learning representative: Siamese Neural Networks (SNNs). We point out that SNNs can be affected by two distinct sources of noise: Pair Label Noise (PLN) and Single Label Noise (SLN). The effect of SLN is asymmetric, but it preserves similarity relations, while PLN is symmetric but breaks transitivity. We find that DD also appears in SNNs and is exacerbated by noise. We show that the dataset topology crucially affects generalization. While sparse datasets show the same performances under SLN and PLN for an equal amount of noise, SLN outperforms PLN in the overparametrized region in dense datasets. Indeed, in this regime, PLN similarity violation becomes macroscopical, corrupting the dataset to the point where complete overfitting cannot be achieved. We call this phenomenon *Density-Induced Break of Similarity (DIBS)*. Probing the equivalence between online optimization and offline generalization in SNNs, we find that their correspondence breaks down in the presence of label noise for all the scenarios considered.

1 INTRODUCTION

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. In this work, we investigate how under- and over-parameterized deep neural networks (DNNs) generalize similarity relations via two frameworks: *Double Descent* (DD) and *online/offline learning correspondence*, which we describe in the following. The empirical success of overparameterized DNNs challenges conventional wisdom in classical statistical learning as it is widely known among practitioners that larger models (with more parameters) usually obtain better generalization Szegedy et al. (2015); Huang et al. (2019); Radford et al. (2019). The DD Belkin et al. (2019) behavior connects "classical" and "modern" machine learning by observing that once the model complexity is large enough to interpolate the dataset (i.e., when training error reaches zero), the test error decreases again. This same pattern has been empirically demonstrated for several models and datasets, ranging from linear models Loog et al. (2020) to modern DNNs Spigler et al. (2019); Nakkiran et al. (2020a). Instead, the online/offline learning correspondence studies the equivalence between online optimization and offline generalization. This was proposed for supervised image classification in Nakkiran et al. (2021). The conjecture states that understanding generalization in an offline setting can be effectively reduced to an optimization problem in the infinite-data limit. Just as the DD pattern described above, this framework connects under- and overparameterized limits. Here, however, the training dataset size dictates the two regimes instead of the number of parameters. Interestingly, online and offline test soft-errors match each other for classification tasks under certain conditions. This phenomenon can be interpreted as a correspondence between overparameterized models (trained on a finite number of samples) and underparameterized models (trained on very large datasets).

DD and online/offline correspondence are two complementary approaches that look at different generalization properties: while DD studies how the network adjusts to the dataset during a long training time or by increasing the number of parameters, online/offline training compares the network performances varying the quality (diversity) of the dataset. These approaches were mainly applied to classification and regression, but, if true, they should also hold for other tasks such as similarity learning.

To take the first steps towards understanding how networks generalize the concept of similarity in the presence and the absence of noise, we investigate both DD and online/offline correspondence in Siamese Neural Networks (SNNs) Bromley et al. (1994); Chopra et al. (2005) and study their phenomenology. While DD was also studied in the presence of noise, very little (if none) attention was devoted to noise in the online/offline setting. Noise is crucial in understanding generalization as it appears in every real-world dataset. In this regard, SNNs turn out to be the perfect playground as they can be affected by two noise sources with very distinctive properties as discussed in the following. 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. Several subtleties should be taken into account in selecting the data and creating the pairs. In particular, we will show that if the pairs are created from populations with different levels of image diversity, the resulting learning model is different, even if the total number of training pairs is fixed. Interestingly, SNNs allow for two sources of label noise in the training dataset: Single Label Noise (SLN) and Pair Label Noise (PLN) that we introduce below (see the top of Fig. 1 for an illustration).

Single Label Noise (SLN). Let us consider a dataset with N samples $X^S = \{x_1, x_2, \ldots, x_N\}$ 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. 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 left panel of Fig. 2). Finally, in 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 $X^P = \{\{x_1^a, x_1^b\}, \ldots, \{x_j^a, x_j^b\}, \ldots, \{x_N^a, x_N^b\}\}$ with pair labels $Y^P = \{y_1^P, y_2^P, \ldots, y_N^P\}$, which can be similar ($y^P = 1$) or different ($y^P = 0$). We construct them so that they are balanced (half are similar, half 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 can impact how models learn similarity relations in distinct ways. Notably, it is empirically known that the DD curve is exacerbated in the presence of random label noise in supervised classification Nakkiran et al. (2020a). Our experiments in Sec. 2.3 show that DD appears in SNNs and is distinctly manifested for SLN and PLN scenarios when similarity relations in the training set are dense (we discuss sparse and dense connections in detail in Sec. 2.1). An example of this behavior is shown in the bottom right plot of Fig. 1. There, we trained a Siamese network made of two fully connected branches on the MNIST dataset LeCun et al. (1998). In this case, dense pairs are formed using a reduced version of the original MNIST. We clearly see that, for dense connections, SLN outperforms PLN in the overparametrized region. The difference between DD curves comes from the similarity-breaking nature of PLN that becomes manifest when input data are highly connected. We call this phenomenon *Density-Induced Break of Similarity (DIBS)*.

This intrinsic property of PLN, i.e., similarity breaking, should not be underestimated as it may appear in widely employed training settings. In fact, modern approaches to self-supervised contrastive learning Ohri & Kumar (2021); Liu et al. (2021); Jaiswal et al. (2020); Le-Khac et al. (2020) heavily rely on data augmentation to learn representations Tian et al. (2020). The massive use of data

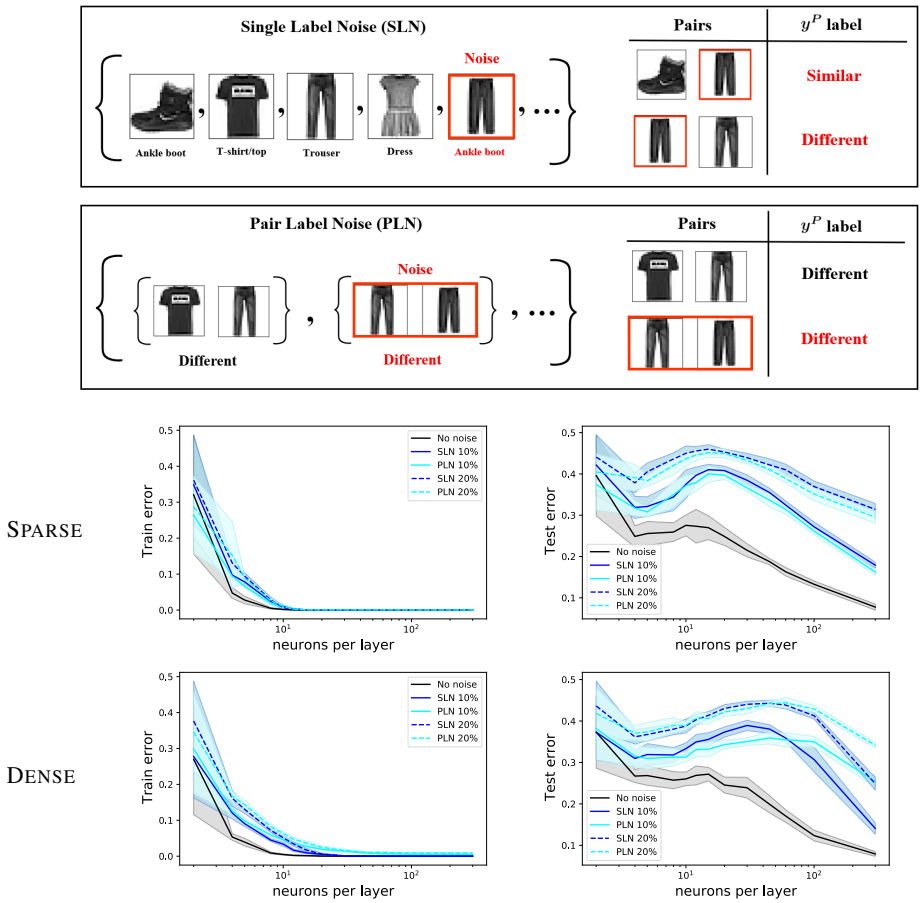

Figure 1: Top: Pictorial view of the two types of random label noise in a Siamese-like setup. Bottom: Train (left) and test error (right) as a function of model size. We train a 3-layer MLP architecture with ReLU activation function on MNIST with 10% and 20% label noise. We considered both sparse and dense connections in the training pair data (scenario 1 and 2 in Sec. 2.1). We train the model using Adam for 2k epochs.

augmentation, however, may result in partial representation learning (feature suppression) or lead to semantic errors Purushwalkam & Gupta (2020). Moreover, as exposed in Huynh et al. (2022), 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. This can break transitivity and, therefore, similarity, compromising the training efficiency. As we show in Fig. 6 in the supplementary material, the error unbalance towards false-negative pairs is the same effect we find in the asymptotic training error in the presence of PLN. Despite all their problems, data augmentation and random selection of negative samples are intrinsic to self-supervised methods.[1] Therefore, several works in contrastive learning have focused on controlling the quality of augmented data and mitigating the effects of false negatives[2] (see section 1.1). For this reason, feature extraction in self-supervised contrastive learning is usually affected by noise by construction.

We test our results on MNIST, FMNIST Xiao et al. (2017) and CIFAR10 Krizhevsky et al. (2009) datasets using Multi-Layer Perceptron (MLP) and Convolution Neural Network (CNN) branch architectures. Moreover, we varied the dataset quality and pairs topology also considering two

[1]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.

[2]Indeed, when two different images belonging to the same class of objects (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 be harmful to the formation of features useful for downstream tasks.

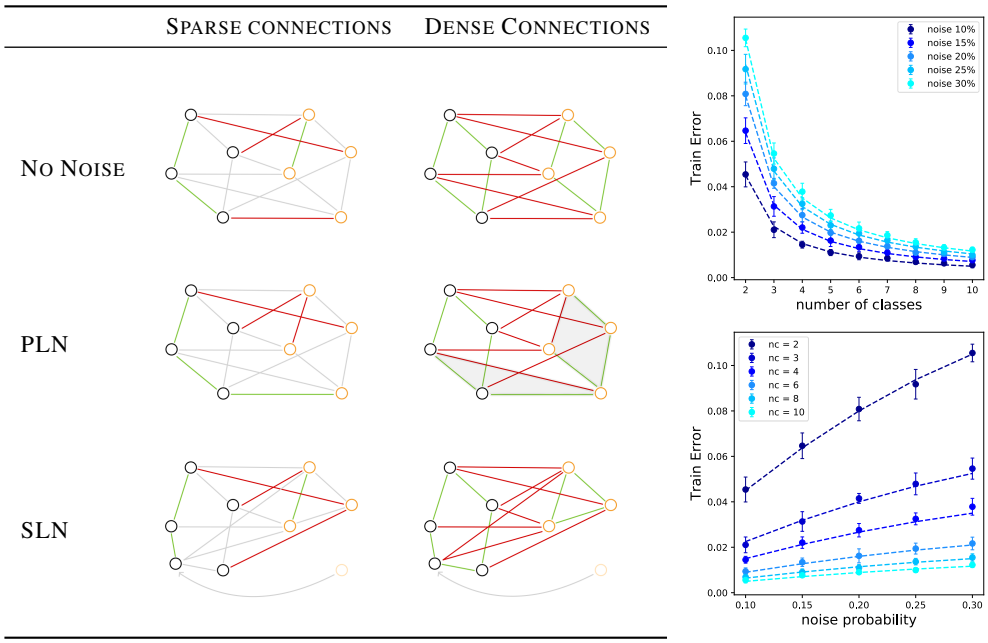

Figure 2: Left: Pictorial view of data relation appearing in Scenario 1 (left) and 2 (right) for two classes of data (represented by black and orange dots). Positive pairs are connected by green edges, negative pairs by red edges. Scenario 1 implies sparse connections among data (ignored connections and data are in light gray), while Scenario 2 is densely connected. Gray shaded areas are examples of DIBS in the presence of PLN. Right: Comparison between analytic and numerical estimates of the asymptotic training error behavior at varying number of classes n_c (top) and noise (bottom) in the presence of PLN in scenario 2. We consider the FMNIST dataset and compute the value of the training error coming from a 3-layers MLP with 500 neurons per layer after 4k epochs. The Siamese output layer computes the Euclidean distance, and we train the network using contrastive loss. Numerical results (mean and standard error bar) come from 10 runs where we choose different random classes each time. Analytic estimates (dashed lines) come from Eq. (6).

training setups. In one case, we compute the Euclidean distance in the output layer training the network using Contrastive Loss Hadsell et al. (2006), in the other one we compute the cosine similarity training the network using Cosine Embedding Loss. All the plots related to the different datasets, architectures, sparse and dense similarity connections, and training setups can be found in Sec. F in the supplementary material.

Our main contributions are:

- We point out that similarity learning may be affected by two distinct noise sources and study the DD behavior in Siamese architectures. Our findings indicate that, for the same amount of effective noise, SLN outperforms PLN on densely connected datasets in the overparametrized regime.
- We identify a phenomenon we call *Density-Induced Break of Similarity (DIBS)*. We find that PLN transitivity breaking introduces significant inconsistencies into the training labels of dense datasets. We show that the interpolation threshold (training error $= 0$) cannot be achieved in this scenario, and we derive the analytic formula for the asymptotic training error value in the deep overparametrized regime.
- We test the correspondence between offline generalization and online optimization for similarity learning. We discuss how the architecture and the presence of noisy labels can differently impact on these two regimes. In particular, the effect of label noise is notably more relevant in the offline case.

1.1 RELATED WORK

Among all possible similarity learning methods, contrastive learning Chopra et al. (2005); Hadsell et al. (2006); Oord et al. (2018) has become one of the most prominent supervised Khosla et al. (2020);

Gunel et al. (2020) and self-supervised Bachman et al. (2019); Tian et al. (2020); He et al. (2020); Chen et al. (2020) ML techniques to learn representations of high-dimensional data, producing impressive results in several fields Le-Khac et al. (2020); Jaiswal et al. (2021). Despite its success, contrastive learning usually requires huge datasets and a considerable use of data augmentation techniques Ohri & Kumar (2021); Liu et al. (2021); Jaiswal et al. (2020); Le-Khac et al. (2020). 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, contrastive loss does not always sufficiently guide which features are extracted Robinson et al. (2021); Wang & Liu (2021). For these reasons several works tackled the problem of discriminating against faulty negativesHuynh et al. (2022); Kalantidis et al. (2020); Chuang et al. (2020); Iscen et al. (2018), removing faulty positives and negatives dynamically Robinson et al. (2021); Zhu et al. (2021) and creating more robust contrastive setups introducing new losses Chuang et al. (2022); Morgado et al. (2021) or architectural components Grill et al. (2020).

In the past few years, much effort has been made to understand how neural networks were able to generalize in classification problems in the presence of noise (e.g., Li et al. (2019); Han et al. (2019); Arazo et al. (2019); Harutyunyan et al. (2020); Song et al. (2020)). Remarkably, the DD behavior allowed to investigate the networks behavior as the number of trainable parameters, the evolution time and the dimensionality of the sample vary Nakkiran et al. (2020a); Bodin & Macris (2021); Heckel & Yilmaz (2020); Pezeshki et al. (2021). Subsequently, other works have produced analytical studies of some of these phenomena d'Ascoli et al. (2020a;b); Mei & Montanari (2022). Another complementary tool used to study generalization in classification tasks is the online/offline correspondence proposed in Nakkiran et al. (2021), which focuses on datasets without noise. This study empirically showed that a correspondence between online optimization and offline generalization holds 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, Bottou & LeCun (2004; 2005)).

2 Similarity learning

Let us start this section by defining the criteria we used to construct the pairs dataset. Indeed, this is an arbitrary choice that may considerably impact the final result. 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 relations among pairs (or groups) of images define an unoriented graph of similarity relations inside the input space.

Calling N the total number of images in the full dataset, the density of this graph, $\rho = |N_{\text{pairs}}|/\binom{N}{2}$, tells us how much information we have about the input images. Therefore, to maximize the information about a certain dataset, we should construct all possible 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 a randomly chosen one belonging to a different class. 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 $N_{\text{pairs}} = 2 \times N = 2 \times n \times n_c$, where n_c is the total number of classes.[3] The following sections describe the procedure to form the pairs, our experimental setup, and how noise is introduced in the training data.

2.1 Sparse and dense connections

The topology of the dataset crucially affects the resulting learning model. Here, we provide details regarding the two methods we use to form pairs in our experiments. These lead to setups with sparse and dense connections in the input space. In the left panel of Fig. 2, we show a pictorial view of the data relations in these two scenarios in the absence of noise, and the SLN and PLN cases. Below we describe how the datasets are created in these two scenarios. In the supplementary material in Sec. B,

[3] Note that this formula holds for dataset with at least 2 images per class.

we give the pseudocodes describing the exact steps to prepare the datasets and how PLN and SLN are added to each case.

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 from the N_{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_{pairs} pairs, we select $N_{\text{reduced}} = N_{\text{pairs}}/2$ images from the original training set. The reduced dataset is balanced so that we have $N_{\text{pairs}}/(2n_c)$ 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 belonging to a different class so that we get exactly N_{pairs} pairs that will be automatically balanced. We repeat this procedure n_s times.

2.2 Effective noise

To make a consistent comparison, we need to introduce the same amount of input label noise in both the SLN and the PLN cases. 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 that is applied to the whole dataset as

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

and the PLN transformation as

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

As the introduction of SLN happens before pair creation and the pairs are constructed so that the dataset is balanced, i.e., half pairs are equal, and 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 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 full derivation of these results is given in the supplementary material in Sec.C.

2.3 Architecture and optimizer

In this work, we consider two simple 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 Glorot & Bengio (2010). The second architecture is a 4-layer CNN based on the model described in Page (2018). 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 ($2k$). To understand the impact of overparameterization, we study how training and test errors vary at increasing network width. Namely, we increase the number of neurons per layer in the fully connected architecture and the parameter k in the CNN. 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 explicitly stated. All the hyper-parameters and the margins were chosen empirically. All our experiments make use of the TensorFlow/Keras framework Abadi et al. (2015). We perform our experiments using two different output layers and loss functions, namely Euclidean distance with contrastive loss and cosine similarity with cosine embedding loss. Further details can be found in Section D in the supplementary material.

3 RESULTS

Double descent. We study DD on MNIST, FMNIST and CIFAR10 datasets in scenarios 1 (Sparse) and 2 (Dense). For all datasets, we consider 6000 training pairs and 9000 test pairs. We run 15 evolutions of the MLP (CNN) network using different training and test samples at each time. Our results, showing the average training and test errors together with error bars, can be found in Fig. 1 and in the supplementary material in Sec. F.1. In all examples, we can see the DD phenomenon, regardless of the architecture, the loss function, the scenario, and the noise. Nevertheless, as expected, DD becomes more prominent in the presence of noise. In most of the experiments, unless otherwise stated, we considered $\tilde{q} = 0.2$, i.e., an effective noise of 10%. In Fig. 1 we show how the network reacts to different amounts of noisy labels.

A distinct aspect of our results compared to the literature for classification problems is that the DD characteristic curves are clearly visible in similarity learning, even without noisy data. To support the consistency of our results with the previous literature on DD, in the supplementary material, we show that epoch-wise DD appears in SNN training. Surprisingly, epoch-wise DD is also observed in the absence of noise, a phenomenon that often occurs only with noisy data for classification tasks (see, e.g., Nakkiran et al. (2020a)).

In Scenario 1, the input dataset connections are sparse, and PLN and SLN have the same impact on training. This makes sense as there should not be any difference between PLN and SLN in the extreme case where every image appears only once in the training set. The results obtained in scenario 1 show that SLN test error tends to be slightly higher than PLN one. This happens because SLN is a slightly unbalanced noise source. Indeed, we experimentally saw that unbalanced noise generically leads to higher test errors.

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 in some but not all setups compared to Scenario 1. Specifically, this happens in MLP-Euclidean Distance and CNN-Cosine Loss setups where the PLN peaks appear to be shifted to the right-hand side, hinting that PLN data are harder to interpolate. No such thing appears using CNN-Euclidean Distance. We believe that the presence of this shift signifies that the right NN setup is being used (i.e., the natural architecture-loss function match). This intuition is supported by further online/offline correspondence experiments we present in the supplementary material. Moreover, increasing the amount of noise consequently enhances the test errors as expected, but we could not see any significant peak shift.

SLN test error tends to be higher from small to medium network sizes. A hint about how this happens is given in Fig. (2). Indeed, on top of being asymmetric, 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. On the other hand, PLN stays higher in the deep overparametrized regime. Indeed, randomly changing similarity relations in the input dataset, PLN ends up breaking transitivity, thus making the training labels inconsistent. Beyond keeping test error high, this inconsistency also implies that the network is not able to overfit the training data completely: the training error will no longer vanish just by increasing the number of network parameters. The amount of transitivity breaking can be quantified at leading order by estimating the probability associated to the following minimal local triangular configuration: two elements, belonging to the same class, are connected to each other, and with the same element of another class and one of the two negative pairs gets mislabeled (see last two rows in Fig. 7 in the supplementary material). In fact, this is given by the probability of having a misclassified equal pair, attached to a correctly classified one. Both pairs should be attached to the same other class of images. Finally, we need to take into account the number of possible configurations. These are two and are given by swapping colors between the two pairs attached to different classes. This leads to:

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

where $\frac{N_{\text{eq.pairs}}}{N_{\text{pairs}}} = \frac{N_{\text{diff.pairs}}}{N_{\text{pairs}}} = \frac{1}{2}$ as we consider balanced pairs. Therefore, the dominant contribution to the asymptotic training error is given by:

$$\lim_{n_\theta \to \infty} \text{TrainError}^{\text{PLN}}_{\text{Dense}}(P, n_c) = \frac{P\,(1-P)}{2(n_c - 1)}, \tag{6}$$

where n_θ is the number of network parameters and $P = P_{\text{PLN}}(\tilde{q})$ is the effective amount of noise. In Fig 2 we validate our formula by comparing it with experimental results showing how, in the overparametrized regime, the training error changes with the effective noise and with the different number of classes.

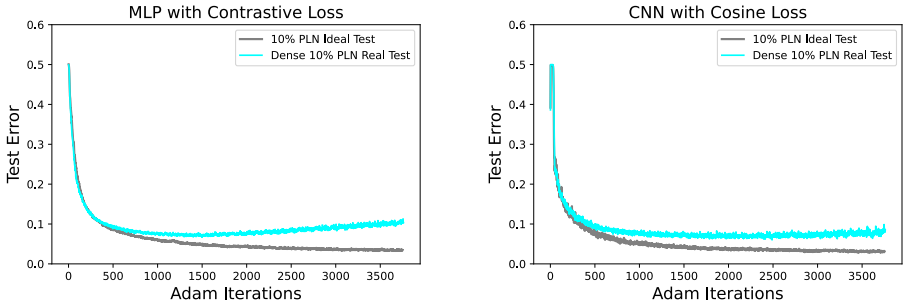

Figure 3: Ideal *vs.* Dense Real worlds with 10% PLN. Plots show the Test Errors as a function of minibatch Adam iterations for a Siamese architecture with MLP (left) branches with 200 nodes per layer and with CNN (right) blocks with width $k = 47$. The architectures details are given in Sec. 2.3.

Online *vs.* offline. We probe the Deep Bootstrap framework Nakkiran et al. (2021) in the context of similarity learning. This framework introduces a correspondence between online and offline settings for classification problems. The proposal establishes a relation between the *Real World* (offline), which has a fixed number of samples in the training set, with the *Ideal World* (online), where the optimizer's updates always use fresh samples. The relation between Real and Ideal scenarios is manifested by the following alternative decomposition of the test error Nakkiran et al. (2021):

$$\text{TestError}(f_t) = \text{TestError}(f_t^{\text{iid}}) + [\text{TestError}(f_t) - \text{TestError}(f_t^{\text{iid}})],$$

where, f_t refers to a neural network after t optimizer steps in the Real World with a fixed number of samples, and f_t^{iid} is a network in the Ideal World that is trained as f_t but without re-using samples. This means that the optimization over minibatches gets new samples at each gradient iteration in the online setting. Note that the decomposition in Eq. (7) makes the *bootstrap error* (the subtraction in brackets) explicit.[4] This measures how well models trained offline represent the online regime. Nakkiran et al. Nakkiran et al. (2021) give empirical evidence that an online/offline correspondence holds for classification problems by showing that the bootstrap error is small in several network models for supervised image classification. The conjecture suggests that understanding generalization in the Real World can be effectively reduced to an optimization problem in online learning. Here, we test if this correspondence is also valid for similarity tasks.

Since we cannot reuse samples for the online training, we consider an extended version of the standard MNIST dataset, namely the EMNIST Cohen et al. (2017). We use the digit section of EMNIST that contains 240,000 training (and 40,000 test) 28×28 greyscale pixel images. The networks are trained considering the setups given in Sec. 2.3. We train Real World over 40 epochs using 12k pairs that are created considering Sparse and Dense scenarios, as described in Sec. 2.1. The 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 in Sec. 2.3.

Real- and Ideal-world scenarios with 10% of of PLN are shown in Fig. 3, where offline and online settings are compared after the same number of training iterations. In order to examine networks with

[4] See Appendix C in Nakkiran et al. (2021) for connections to the nonparametric bootstrap in statistics Efron (1979); Efron & Tibshirani (1986). Also, as noted in Nakkiran et al. (2021), the bootstrap error can be related to algorithmic stability (e.g., Bousquet & Elisseeff (2001); Hardt et al. (2016)).

a comparable total number of parameters, we use 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 ($\lambda = 5 \times 10^{-5}$). We plot the median over 5 trials (random network initialization and random noise sampling selection) for the Real-world cases with MLP blocks, and the median over 4 trials for the Real-world CNN scenarios. We observe that both Ideal and Real test errors are affected by noise, but this effect is exacerbated in the Real World scenarios. This can be understood because "fresh" samples add more diversity into the model, which improves generalization even if these new samples present noisy labels.

Interestingly, the online/offline correspondence for similarity tasks is influenced by the network architecture and the loss function choice. In particular, Fig. 3 shows that the Real-world scenario for the MLP architecture diverges from the corresponding online case earlier (with less iteration steps) than the CNN case. 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. Similar behavior occurs for the sparse case as shown in Sec. F.5 in the supplementary material. We present several additional comparisons between the architectures, losses and noise levels in Sec. F.5 in the supplementary material.

4 Discussion and conclusions

This work studies generalization in similarity learning with noisy labeled data focusing, in particular, on SNNs. To do so, we use two different frameworks: DD, and online/offline correspondence. We find that DD clearly appears in SNNs and becomes more evident in the presence of noisy labels. We check DD using different architectures, datasets, pairs arrangements, and loss functions. We present two kinds of noise, SLN, and PLN, that may affect the input data. While SLN preserves similarity relations, PLN breaks transitivity. We show that similarity-breaking noise sources deeply affect generalization performances. The same noise sources presented in this work can be easily generalized to models where the network input is given by multiple images. We also investigate the equivalence between online optimization (infinite-data regime) and offline generalization (finite number of samples) for similarity problems. Our results indicate that both the network architecture and the loss function choice can disturb an online/offline correspondence for similarity tasks. Notably, we find that the online/offline equivalence breaks down in the presence of label noise for all the scenarios considered.

Limitations. This is an exploratory work that does not investigate all possible setups which may affect or lead to DD, such as regularization Nakkiran et al. (2020b); Mei & Montanari (2022), epoch and sample-wise DD Nakkiran et al. (2020a); Bodin & Macris (2021); Heckel & Yilmaz (2020); Pezeshki et al. (2021). Moreover, we focus on the under- and over-parametrised regime without providing quantitative results about the interpolation threshold itself d'Ascoli et al. (2020a;b); Mei & Montanari (2022). 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 Rahimi & Recht (2007) can be constructed. Indeed, as opposed to classification or regression tasks where the output layer size is known, SNNs do not have a predefined output layer size. For this reason, we believe that an analytic study of DD in SNNs may require some other 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 were we find that noise crucially impacts generalization. While the technological developments and the applications of contrastive learning kept on expanding during the last years, a fundamental study about how it generalises and reacts to noise is still missing.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew

Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL https://www.tensorflow.org/. Software available from tensorflow.org.

Ben Adlam and Jeffrey Pennington. The neural tangent kernel in high dimensions: Triple descent and a multi-scale theory of generalization. In *International Conference on Machine Learning*, pp. 74–84. PMLR, 2020.

Eric Arazo, Diego Ortego, Paul Albert, Noel E. O'Connor, and Kevin McGuinness. Unsupervised label noise modeling and loss correction. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 312–321. PMLR, 2019. URL http://proceedings.mlr.press/v97/arazo19a.html.

Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. *arXiv preprint arXiv:1906.00910*, 2019.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, , and Soumik Mandal. 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. URL https://doi.org/10.1073/pnas.1903070116.

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.

Léon Bottou and Yann LeCun. Large scale online learning. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf (eds.), *Advances in Neural Information Processing Systems 16 (NIPS 2003)*. MIT Press, Cambridge, MA, 2004. URL http://leon.bottou.org/papers/bottou-lecun-2004.

Léon Bottou and Yann LeCun. On-line learning for very large datasets. *Applied Stochastic Models in Business and Industry*, 21(2):137–151, 2005. URL http://leon.bottou.org/papers/bottou-lecun-2004a.

Olivier Bousquet and André Elisseeff. Algorithmic stability and generalization performance. *Advances in Neural Information Processing Systems*, pp. 196–202, 2001.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a "siamese" time delay neural network. In J. Cowan, G. Tesauro, and J. Alspector (eds.), *Advances in Neural Information Processing Systems*, volume 6. Morgan-Kaufmann, 1994. URL https://proceedings.neurips.cc/paper/1993/file/288cc0ff022877bd3df94bc9360b9c5d-Paper.pdf.

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

S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pp. 539–546 vol. 1, 2005. doi: 10.1109/CVPR.2005.202.

Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. *Advances in neural information processing systems*, 33:8765–8775, 2020.

Ching-Yao Chuang, R Devon Hjelm, Xin Wang, Vibhav Vineet, Neel Joshi, Antonio Torralba, Stefanie Jegelka, and Yale Song. Robust contrastive learning against noisy views. *arXiv preprint arXiv:2201.04309*, 2022.

G. Cohen, S. Afshar, J. Tapson, and A. van Schaikf. EMNIST: an extension of MNIST to handwritten letters. 2017. URL https://github.com/hosford42/EMNIST.

Stéphane d'Ascoli, Maria Refinetti, Giulio Biroli, and Florent Krzakala. Double trouble in double descent: Bias and variance (s) in the lazy regime. pp. 2280–2290, 2020a.

Stéphane d'Ascoli, Levent Sagun, and Giulio Biroli. Triple descent and the two kinds of overfitting: Where & why do they appear? *Advances in Neural Information Processing Systems*, 33:3058–3069, 2020b.

B. Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1):1–26, 1979. ISSN 00905364. URL http://www.jstor.org/stable/2958830.

Bradley Efron and Robert Tibshirani. Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. *Statistical science*, pp. 54–75, 1986.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256. JMLR Workshop and Conference Proceedings, 2010.

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.

Beliz Gunel, Jingfei Du, Alexis Conneau, and Ves Stoyanov. Supervised contrastive learning for pre-trained language model fine-tuning. *arXiv preprint arXiv:2011.01403*, 2020.

R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pp. 1735–1742, 2006. doi: 10.1109/CVPR.2006.100.

Jiangfan Han, Ping Luo, and Xiaogang Wang. Deep self-learning from noisy labels. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pp. 5137–5146. IEEE, 2019. doi: 10.1109/ICCV.2019.00524. URL https://doi.org/10.1109/ICCV.2019.00524.

Moritz Hardt, Ben Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International Conference on Machine Learning*, pp. 1225–1234. PMLR, 2016.

Hrayr Harutyunyan, Kyle Reing, Greg Ver Steeg, and Aram Galstyan. Improving generalization by controlling label-noise information in neural network weights. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 4071–4081. PMLR, 2020. URL http://proceedings.mlr.press/v119/harutyunyan20a.html.

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*, pp. 9729–9738, 2020.

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

Yanping Huang, Youlong Cheng, Ankur Bapna, Orhan Firat, Dehao Chen, Mia Chen, HyoukJoong Lee, Jiquan Ngiam, Quoc V Le, Yonghui Wu, et al. Gpipe: Efficient training of giant neural networks using pipeline parallelism. *Advances in neural information processing systems*, 32: 103–112, 2019.

Tri Huynh, Simon Kornblith, Matthew R Walter, Michael Maire, and Maryam Khademi. Boosting contrastive self-supervised learning with false negative cancellation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2785–2795, 2022.

Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondřej Chum. Mining on manifolds: Metric learning without labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7642–7651, 2018.

Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2020.

Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2021.

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

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.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Phuc H Le-Khac, Graham Healy, and Alan F Smeaton. Contrastive representation learning: A framework and review. *IEEE Access*, 8:193907–193934, 2020.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S. Kankanhalli. Learning to learn from noisy labeled data. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 5051–5059. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00519. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Li_Learning_to_Learn_From_Noisy_Labeled_Data_CVPR_2019_paper.html.

Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

Marco Loog, Tom Viering, Alexander Mey, Jesse H Krijthe, and David MJ Tax. A brief prehistory of double descent. *Proceedings of the National Academy of Sciences*, 117(20):10625–10626, 2020.

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.

Pedro Morgado, Ishan Misra, and Nuno Vasconcelos. Robust audio-visual instance discrimination. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12934–12945, 2021.

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. In *International Conference on Learning Representations*, 2020a. URL https://openreview.net/forum?id=B1g5sA4twr.

Preetum Nakkiran, Prayaag Venkat, Sham Kakade, and Tengyu Ma. Optimal regularization can mitigate double descent. *arXiv preprint arXiv:2003.01897*, 2020b.

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, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=guetrIHLFGI.

Kriti Ohri and Mukesh Kumar. Review on self-supervised image recognition using deep neural networks. *Knowledge-Based Systems*, 224:107090, 2021.

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

David Page. How to train your resnet. 2018. URL https://myrtle.ai/how-to-train-your-resnet-4-architecture/.

Mohammad Pezeshki, Amartya Mitra, Yoshua Bengio, and Guillaume Lajoie. Multi-scale feature learning dynamics: Insights for double descent. *arXiv preprint arXiv:2112.03215*, 2021.

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.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

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

Joshua Robinson, Li Sun, Ke Yu, Kayhan Batmanghelich, Stefanie Jegelka, and Suvrit Sra. Can contrastive learning avoid shortcut solutions? *Advances in Neural Information Processing Systems*, 34, 2021.

Hwanjun Song, Minseok Kim, Dongmin Park, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *CoRR*, abs/2007.08199, 2020. URL https://arxiv.org/abs/2007.08199.

Stefano Spigler, Mario Geiger, Stéphane d'Ascoli, Levent Sagun, Giulio Biroli, and Matthieu Wyart. A jamming transition from under-to over-parametrization affects generalization in deep learning. *Journal of Physics A: Mathematical and Theoretical*, 52(47):474001, 2019.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *European conference on computer vision*, pp. 776–794. Springer, 2020.

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

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. 2017. URL https://github.com/zalandoresearch/fashion-mnist.

Rui Zhu, Bingchen Zhao, Jingen Liu, Zhenglong Sun, and Chang Wen Chen. Improving contrastive learning by visualizing feature transformation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10306–10315, 2021.