REVISITING CONTRASTIVE LEARNING THROUGH THE LENS OF NEIGHBORHOOD COMPONENT ANALYSIS: AN INTEGRATED FRAMEWORK

Anonymous authors
Paper under double-blind review

ABSTRACT

As a seminal tool in self-supervised representation learning, contrastive learning has gained unprecedented attention in recent years. In essence, contrastive learning aims to leverage pairs of positive and negative samples for representation learning, which relates to exploiting neighborhood information in a feature space. By investigating the connection between contrastive learning and neighborhood component analysis (NCA), we provide a novel stochastic nearest neighbor viewpoint of contrastive learning and subsequently propose a series of contrastive losses that outperform the existing ones. Under our proposed framework, we show a principled way to design integrated contrastive losses that simultaneously achieve good accuracy and robustness on downstream tasks.

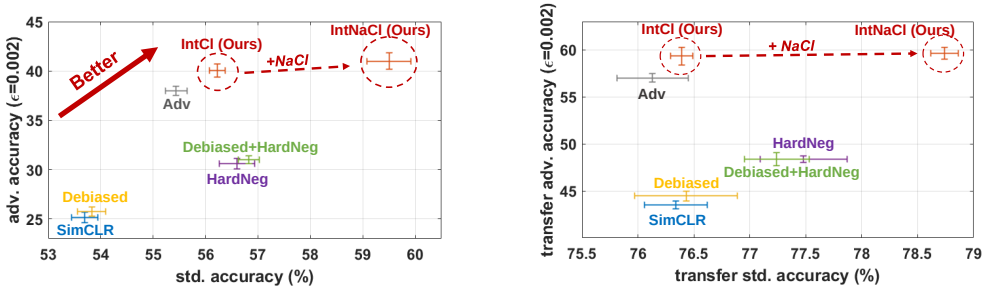
1 INTRODUCTION

With the growing need for deployable machine learning, contrastive learning has drawn much attention and has become one of the most effective representation learning techniques in the past years. The contrastive paradigm (Oord et al., 2018; Wu et al., 2018; He et al., 2020; Chen et al., 2020a; Chuang et al., 2020; Grill et al., 2020) constructs an objective for embeddings based on an assumed semantic similarity, and the ability to distinguish dissimilar instances. This, in turn, stems from instance-level classification (Dosovitskiy et al., 2015; Bojanowski & Joulin, 2017; Wu et al., 2018). Specifically, the contrastive loss $\mathcal{L}_{\text{CL}}$ (Oord et al., 2018; Chen et al., 2020a) is given by

$$\mathcal{L}_{\text{CL}} := \mathbb{E}_{x\sim\mathcal{D},x^+\sim\mathcal{D}_x^+,x_i^-\sim\mathcal{D}_x^-}\left[-\log\frac{e^{f(x)^T f(x^+)}}{e^{f(x)^T f(x^+)}+\sum_{i=1}^{N} e^{f(x)^T f(x_i^-)}}\right], \tag{1}$$

where, for a sample x, (x, x^+) denotes a positive pair and (x, x^-) denotes a negative pair. The function f is normally parameterized by a neural network and the number of negative pairs N is typically treated as a hyperparameter. When constructing loss $\mathcal{L}_{\text{CL}}$ in Equation 1, ideally, one draws x^+ from the data distribution $\mathcal{D}_x^+$ that characterizes the semantically-*similar* (i.e., *positive*) samples to x; similarly, one wants to draw x^- from $\mathcal{D}_x^-$ that characterizes the semantically-*dissimilar* (*negative*) samples.

However, this construction faces several challenges in practice. First, the definition of semantically-*similar* and semantically-*dissimilar* is contingent on downstream tasks. For example, an image of a cat can be considered semantically similar to that of a dog if the downstream task is to distinguish between animal and non-animal classes. To provide a surrogate of measuring similarity, mainstream contrastive learning algorithms (He et al., 2020; Chen et al., 2020a;b; Grill et al., 2020) typically build up $\mathcal{D}_x^+$ by considering data augmentation $\mathcal{D}_x^{\text{aug}}$ of a data sample x. We note that although $\mathcal{D}_x^{\text{aug}}$ might over-simplify the construction of $\mathcal{D}_x^+$, as only the standard data augmentation is used, the positive samples generated in $\mathcal{D}_x^{\text{aug}}$ are *valid* positive examples w.r.t. x. In contrast, the construction of $\mathcal{D}_x^-$ in existing contrastive learning literature (Oord et al., 2018; Chen et al., 2020a) contains a non-negligible portion of *invalid* samples due to a lack of label information in the unsupervised learning setting (Chuang et al., 2020), where $\mathcal{D}_x^-$ is approximated by the joint distribution $\mathcal{D}$ or $\mathcal{D}_{\setminus x}^{\text{aug}} := \cup_{x'\in\mathcal{D}\setminus\{x\}}\mathcal{D}_{x'}^{\text{aug}}$ in practice. More precisely, if the negative samples are sampled from $\mathcal{D}_x^-$,



(a) CIFAR100 (b) CIFAR10

Figure 1: The performance of existing methods and our proposal (IntNaCl & IntCl) in terms of their standard accuracy (x-axis) and adversarial accuracy under FGSM attacks $\epsilon = 0.002$ (y-axis). The transfer performance refers to fine-tuning a linear layer for CIFAR10 with representation networks trained on CIFAR100.

we will receive with $1/K$ probability a positive sample in a K-class classification task with balanced classes. This may be undesirable. This heuristic loss was proposed in (Chen et al., 2020a) and is known as $\mathcal{L}_{\text{SimCLR}}$:

$$\mathcal{L}_{\text{SimCLR}} := \mathbb{E}_{x\sim\mathcal{D},x^+\sim\mathcal{D}_x^{\text{aug}},x_i^-\sim\mathcal{D}_{\setminus x}^{\text{aug}}}\left[-\log\frac{e^{f(x)^T f(x^+)}}{e^{f(x)^T f(x^+)}+\sum_{i=1}^{N} e^{f(x)^T f(x_i^-)}}\right]. \tag{2}$$

Another challenge is the computation of positive and negative pairs grows quadratically with the size of the dataset. Therefore, computing all the pairwise comparisons on a large dataset is not practical, and most implementations approximate the loss by reducing the number of comparisons to random subsets of images during training (Wu et al., 2018; He et al., 2020; Chen et al., 2020a).

In this paper, we aim to examine the above challenges through the lens of the nearest neighbor classification in Neighborhood Component Analysis (NCA) (Goldberger et al., 2004). Specifically, we uncover the relationship between stochastic nearest neighbors and positive pairs in contrastive learning, which then motivates a sequence of augmented contrastive losses that work better under practical computational constraints. Furthermore, representation learning has been evaluated mostly by how they cluster or by a metric such as the standard downstream classification accuracy. However, by inspecting the adversarial accuracy of several existing methods (e.g., Figure 1's y-axis, the classification accuracy when inputs are corrupted by crafted perturbations), one can see the insufficiency of those methods in addressing robustness. We thus present a general integrated contrastive framework that accounts for *both* the standard accuracy and adversarial cases; this method's performance remains in the desired upper-right region (circled) as shown in Figure 1. A conceptual illustration of our proposals is given in Figure 2. We summarize our main contributions as follows:

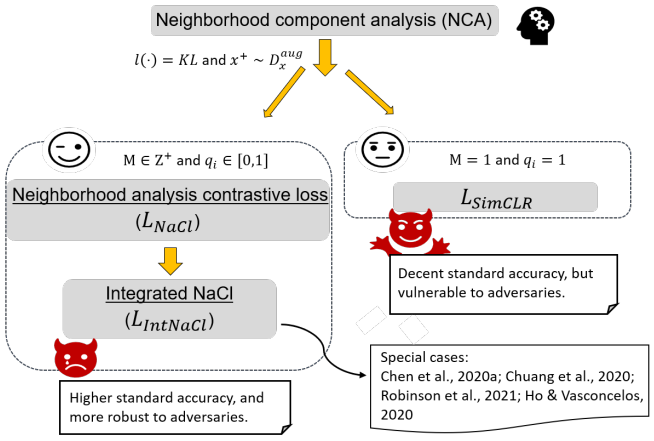


Figure 2: A conceptual illustration of the relationships among NCA, $\mathcal{L}_{\text{SimCLR}}$, and our proposals.

- We relate contrastive losses with Neighborhood Component Analysis (NCA) and generalize it in the contrastive learning setting, which we dub **NaCl**, Neighborhood analysis Contrastive loss.
- Building on top of NaCl, we proposed a generic framework called Integrated contrastive learning (**IntCl** and **IntNaCl**) where we show that the spectrum of recently-proposed contrastive learning losses (Chuang et al., 2020; Robinson et al., 2021; Ho & Vasconcelos, 2020) can be included as special cases of our framework;
- We provide extensive experiments that demonstrate the effectiveness of **IntNaCl** in improving standard accuracy and adversarial accuracy. Specifically, **IntNaCl** improves upon literature (Chen et al., 2020a; Chuang et al., 2020; Robinson et al., 2021; Ho & Vasconcelos, 2020) by 3-6% and 4-16% in CIFAR100 standard and adversarial accuracy, and 2-3% and 3-17% in CIFAR100 standard and adversarial accuracy, respectively.

2 Related Work

Contrastive learning. In the early work of Dosovitskiy et al. (2015), authors treat every individual image in a dataset as one own class and do multi-class classification tasks under the setting. However, this regime will soon become intractable as we have a large dataset. To cope with this, Wu et al. (2018) design a memory bank for storing presented representations (keys) and utilize noise contrastive estimation (Gutmann & Hyvärinen, 2010; Mnih & Teh, 2012; Jozefowicz et al., 2016; Oord et al., 2018) for representation comparisons. Then, He et al. (2020) and Chen et al. (2020b) further improve upon this by storing keys inferred from a momentum encoder other than the representation encoder for x. Finally, besides the practical tricks introduced in SimCLR (Chen et al., 2020a) (e.g. stronger data augmentation scheme and projector heads), authors of SimCLR also get rid of the memory bank and instead makes use of other samples from the same batch to form contrastive pairs. Caron et al. (2020) consider cluster assignment tasks instead of instance learning and design an online algorithm for learning. Specifically, if we fix the trainable prototypes C to be the data batch and choose code Q to be identity, then the swapped prediction problem reduces to similar forms of Equation 3. Grill et al. (2020) argue the possibility of forming contrastive losses without the need for negative examples. In the rest of this paper, we will focus on the setups of SimCLR and the related follow up work (Chuang et al., 2020; Robinson et al., 2021; Ho & Vasconcelos, 2020) due to computational efficiency. We let $g_0(x, \{x_i^-\}_i^N)$ denote the negative term $\frac{1}{N}\sum_{i=1}^N e^{f(x)^T f(x_i^-)}$, where the subscript i identifies the summation index and the superscript N identifies the summation limits. We omit the subscript i when the sample index is one dimensional (e.g. x_i^- has 1-D index, x_{ij}^- has 2-D index). Moreover, we define $K(A, B) = -\log(A/(A+B))$. Then $\mathcal{L}_{\text{SimCLR}}$ in Equation 2 can be re-written as

$$\mathcal{L}_{\text{SimCLR}} := \mathbb{E}_{x\sim\mathcal{D}, x^+\sim\mathcal{D}_x^{\text{aug}}, x_i^-\sim\mathcal{D}_{\setminus x}^{\text{aug}}}\Big[K(e^{f(x)^T f(x^+)}, Ng_0(x, \{x_i^-\}^N))\Big]. \tag{3}$$

Designing *negative* pairs in contrastive learning. Several works (Saunshi et al., 2019; Chuang et al., 2020) have come to the awareness of the sampling bias of negative pairs as discussed in Section 1. Chuang et al. (2020) propose a *de-biased* constrastive loss to mitigate the sampling bias by assuming the priors on the downstream tasks (e.g., with probability 0.1, a positive sample can be used as x_i^- in CIFAR10) without using any explicit label information. We denote the loss from Chuang et al. (2020) as $\mathcal{L}_{\text{Debiased}}$ and the full equation is shown below:

$$\mathcal{L}_{\text{Debiased}} := \mathbb{E}_{x\sim\mathcal{D}, x^+, v_j\sim\mathcal{D}_x^{\text{aug}}, u_i\sim\mathcal{D}_{\setminus x}^{\text{aug}}}\Big[K(e^{f(x)^T f(x^+)}, Ng_1(x, \{u_i\}^n, \{v_j\}^m))\Big], \tag{4}$$

where the estimator $g_1(x, \{u_i\}^n, \{v_j\}^m)$ is defined by

$$g_1(x, \{u_i\}^n, \{v_j\}^m) = \max\{\frac{1}{1-\tau^+}(\frac{1}{n}\sum_{i=1}^n e^{f(x)^T f(u_i)} - \tau^+ \frac{1}{m}\sum_{j=1}^m e^{f(x)^T f(v_j)}), e^{-1/t}\}$$

and n and m represents the numbers of sampled points in $\mathcal{D}_{\setminus x}^{\text{aug}}$ and $\mathcal{D}_x^{\text{aug}}$ for the re-weighted negative term with τ^+ being the probability that a positive pair is mistakenly to be a negative pair when we sample x_i^- from $\mathcal{D}_{\setminus x}^{\text{aug}}$. Recently, Robinson et al. (2021) propose to weigh sample pairs through

the cosine distance in the estimator $g_1(x, \{u_i\}^n, \{v_j\}^m)$ based on $\mathcal{L}_{\text{Debiased}}$, and we denote their approach as $\mathcal{L}_{\text{Debiased+HardNeg}}$,

$$\mathcal{L}_{\text{Debiased+HardNeg}} := \mathbb{E}_{x\sim\mathcal{D},x^+,v_j\sim\mathcal{D}_x^{\text{aug}},u_i\sim\mathcal{D}_{\setminus x}^{\text{aug}}}\Big[K(e^{f(x)^T f(x^+)}, Ng_2(x, \{u_i\}^n, \{v_j\}^m))\Big], \tag{5}$$

where the estimator $g_2(x, \{u_i\}^n, \{v_j\}^m)$ is defined by

$$g_2(x, \{u_i\}^n, \{v_j\}^m) = \max\{\frac{1}{1-\tau^+}(\frac{\sum_{i=1}^n e^{(\beta+1)f(x)^T f(u_i)}}{\sum_{i=1}^n e^{\beta f(x)^T f(u_i)}} - \tau^+ \frac{1}{m}\sum_{j=1}^m e^{f(x)^T f(v_j)}), e^{-1/t}\}$$

A typical choice of n and m are $n = N$ and $m = 1$, and the hyperparameter τ^+ in g_2 is exactly the same as that in g_1 whereas the hyperparameter β controls the weighting mechanism. Specifically, when $\tau^+ = 0$, we denote $\mathcal{L}_{\text{Debiased+HardNeg}}$ as $\mathcal{L}_{\text{HardNeg}}$; when $\beta = 0$, Equation 5 degenerates to Equation 4 which is $\mathcal{L}_{\text{Debiased}}$.

Designing *positive* pairs in contrastive learning. In parallel to the above line of work, another direction is to augment the construction of positive pairs (Ho & Vasconcelos, 2020; Kim et al., 2020). Specifically, authors of Ho & Vasconcelos (2020) define the concept of *adversarial examples* in the regime of representation learning as the positive sample that maximizes $\mathcal{L}_{\text{SimCLR}}$ (i.e. Equation 3) within a pre-specified perturbation magnitude. The resulting loss function is $\mathcal{L}_{\text{Adv}}$:

$$\begin{aligned}\mathcal{L}_{\text{Adv}} := \mathbb{E}_{x\sim\mathcal{D},x^+\sim\mathcal{D}_x^{\text{aug}},x_{i_1}^-\sim\mathcal{D}_{\setminus x}^{\text{aug}},x_{i_2}^-\sim\mathcal{D}_{\setminus x}^{\text{adv}}}\Big[&K(e^{f(x)^T f(x^+)}, Ng_0(x, \{x_{i_1}^-\}^N)) \\ &+\alpha K(e^{f(x)^T f(x^{\text{adv}})}, Ng_0(x, \{x_{i_2}^-\}^N))\Big],\end{aligned} \tag{6}$$

where the $\mathcal{D}_{\setminus x}^{\text{adv}}$ is defined by $\cup_{x'\in\mathcal{D}\setminus\{x\}} x' \cup x'^{,\text{adv}}$. Notably, one can adjust the importance of the adversarial term by tuning α in Equation 6. A relevant idea is exploited by Kannan et al. (2018) albeit in the supervised learning setting, where the adversarial logit paring works under a supervised learning regime and encourages an adversarial sample x' to have the same logit output as the clean x. Notice that the definition of adversary in Kannan et al. (2018) is different from that in Ho & Vasconcelos (2020), where it is defined by a perturbed sample that is predicted differently from the original. This can be viewed as supervised contrastive learning where the representation space lies in $\mathbb{R}^1$ with the similarity evaluated by ℓ_2 distances. In Khosla et al. (2020), authors construct supervised contrastive loss using multiple positive pairs defined by samples with the same label. Another related work is InfoMin (Tian et al., 2020), which needs a small amount of labeled data to perform two-step trainings. In the first step, a view generator is learned by minimizing the mutual information between latents from views while ensuring all of them can be correctly classified; in the second step, the trained generator is frozen and the representation network is learned by maximizing the mutual information between latents from views. We note that InfoMin can be regarded as inserting a learned "hard positive sample" into contrastive learning. Dwibedi et al. (2021) inherit the usage of a memory bank as in He et al. (2020) and select from it the representation candidate z' nearest to $z = f(x)$ (determined by nearest neighbor). Eventually the loss is formed by substituting z' for z in Equation 3.

Adversarial Robustness. Despite neural networks' supremacy in achieving impressive performance, they have been proved vulnerable to human-imperceptible perturbations (Goodfellow et al., 2015; Szegedy et al., 2014; Nguyen et al., 2015; Moosavi-Dezfooli et al., 2016). In the supervised learning setting, an adversarial perturbation δ is defined to render inconsistent classification result of the input x: $r(x + \delta) \neq r(x)$, where r is a neural network classifier. A stronger adversarial attack means it can find δ with higher success attack rate under the same ϵ-budget ($\|\delta\|_p \leq \epsilon$). One of the most popular and classical attack algorithms is FGSM (Goodfellow et al., 2015), where with a fixed perturbation magnitude ϵ, FGSM uses the sign of cross entropy gradient to decide between $\delta = \epsilon$ and $\delta = -\epsilon$. Another popular attack method we consider in this paper is PGD (Madry et al., 2018), which assembles the iterative-FGSM (Kurakin et al., 2016) but with additional projection steps. Considering the robustness in representations, except the inclusion of adversarial examples as positive samples (Ho & Vasconcelos, 2020; Kim et al., 2020), recent literature (Gowal et al., 2020) has strengthened BYOL (Grill et al., 2020) by including an adversarial training loss.

3 NEIGHBORHOOD ANALYSIS CONTRASTIVE LOSS

In this section, we first establish a connection between the literature on Neighborhood Component Analysis (NCA) (Goldberger et al., 2004) and the contrastive learning loss in Section 3.1. Inspired by our result in Section 3.1, we further design three novel contrastive losses in Section 3.2, which we refer to as **N**eighborhood **a**nalysis **C**ontrastive **l**oss (**NaCl**), that are beneficial for learning more powerful representations that can achieve higher downstream task accuracy.

3.1 STOCHASTIC NEAREST NEIGHBOR FRAMEWORK

NCA is a supervised learning algorithm concerned with learning a distance metric that maximizes the performance of nearest neighbour classification. As nearest neighbor is a non-smooth function of points, the optimization problem is generally given using the concept of stochastic nearest neighbors. In the stochastic nearest neighbor setting, nearest neighbor selection is regarded as a random event, where the probability point x_j is selected as the nearest neighbor for x_i is given as $p(x_i, x_j)$ with

$$p_{ij} = p(x_i, x_j) = \frac{e^{-\|Ax_i - Ax_j\|^2}}{\sum_{k \neq i} e^{-\|Ax_i - Ax_k\|^2}} \propto e^{-\|Ax_i - Ax_j\|^2}.$$

Let c_i denote the label of x_i, in the leave-one-out classification loss, the probability a point is classified correctly is given as $p_i = \sum_{j|c_j=c_i} p_{ij}$, where $\{j \mid c_j = c_i\}$ defines an index set with which all points x_j belong to the same class as point x_i. The probability x_i's label is c_i is given as q_i, which is exactly 1. Thus the optimization problem can be written as $\min_A \sum_{i=1}^n \ell(q_i, \sum_{j|c_j=c_i} p_{ij})$. This learning objective then naturally maximizes the expected accuracy of a 1-nearest neighbor classifier. Two popular choices for $\ell(\cdot)$ are the total variation distance and the KL divergence. In the seminal paper of Goldberger et al. (2004), the authors showed both losses give similar results. We will focus on the KL divergence loss in this work. For $\ell(\cdot) = \text{KL}$, the relative entropy from p to q is $D_{\text{KL}}(q\|p) = \sum_i -q_i \log \frac{p_i}{q_i} = \sum_i -\log p_i$, when $q_i = 1$, and the optimization problem becomes

$$\min_A \sum_{i=1}^n -\log \left(\sum_{j|c_j=c_i} \frac{e^{-\|Ax_i - Ax_j\|^2}}{\sum_{k \neq i} e^{-\|Ax_i - Ax_k\|^2}} \right). \tag{7}$$

With the above formulation, we argue that the contrastive learning loss can be given assuming only positive pairs belong to the same class and the metric Ax is instead parametrized by a function $\frac{f(x)}{\sqrt{2}} = \frac{h(x)}{\sqrt{2}\|h(x)\|}$, where h is a neural network. Specifically, Equation 7 becomes Equation 9:

$$\min_f \sum_{i=1}^n -\log \left(\sum_{j=1}^M \frac{e^{-\frac{1}{2}\left\|f(x_i) - f(x_j^+)\right\|^2}}{\sum_{k \neq i} e^{-\frac{1}{2}\|f(x_i) - f(x_k)\|^2}} \right) \tag{8}$$

$$\xrightarrow{\|f(x)\|=1} \min_f \mathbb{E}_{x \sim \mathcal{D}} \left[K(\sum_{j=1}^M e^{f(x)^T f(x_j^+)}, N g_0(x, \{x_i^-\}^N)) \right]. \tag{9}$$

We give the full derivation from Equation 8 to Equation 9 in the appendix. We note that the contrastive loss in Chen et al. (2020a) is a special case of Equation 9 with $M = 1, x^+ \sim \mathcal{D}_x^{\text{aug}}$, which yields the minimization over the exact form of $\mathcal{L}_{\text{SimCLR}}$ as given in Equation 3,

$$\min_f \mathbb{E}_{x \sim \mathcal{D}, x^+ \sim \mathcal{D}_x^{\text{aug}}, x_i^- \sim \mathcal{D}_{\setminus x}^{\text{aug}}} \left[K(e^{f(x)^T f(x^+)}, N g_0(x, \{x_i^-\}^N)) \right].$$

3.2 NCA INSPIRED CONTRASTIVE LOSS

Although we have shown the connection between NCA and contrastive learning, applying the NCA framework exactly is challenging in two senses. On one hand, it requires us to use all possible negative pairs which is approximately the size of the entire dataset. Furthermore, to decide the "demographic " of a point's neighborhood, M depends on the relative density of positive to negative pairs one expects to have in the underlying data distribution. To tackle these, as using the entire dataset to approximate the population loss is computationally infeasible, we propose to use a stochastic approximation to

the population loss, where N is determined as a hyperparameter in a fashion similar to the batch size hyperparameter (Chen et al., 2020a). In order to determine M, as the expected relative density is task-dependent, we treat the M/N ratio as a hyperparameter similar to the class probabilities τ^+ introduced by Chuang et al. (2020).

We start this section by examining the specifications $\mathcal{L}_{\text{SimCLR}}$ has made when simplifying from a general stochastic nearest neighbor algorithm: $\ell(\cdot) = \text{KL}$, $q_i = 1$, $M = 1, x^+ \sim \mathcal{D}_x^{\text{aug}}$. If we keep the first parts of the simplification fixed (i.e. $\ell(\cdot) = \text{KL}$, $q_i = 1$, $M = 1$), then in practice, the expectation over the probability distribution $\mathcal{D}_x^{\text{aug}}$ is estimated by only one sample. To reduce the variance of such an estimator, with a slight abuse of notation, we denote the number of trials by M and propose to simulate M trials of the procedure for every x. This yields the following loss[1]

$$\mathcal{L}_{\text{VAR}}(g = g_0, M) := \mathbb{E}_{x\sim\mathcal{D}, x_j^+\sim\mathcal{D}_x^{\text{aug}}, x_{ij}^-\sim\mathcal{D}_{\setminus x}^{\text{aug}}} \left[\frac{1}{M}\sum_{j=1}^{M} K(e^{f(x)^T f(x_j^+)}, Ng_0(x, \{x_{ij}^-\}_i^N))\right].$$

By shifting our focus to the number of neighbors that are considered belonging to the same class (M), we admit the potential bias induced by assuming $M = 1$ (i.e. the relative density of positive to negative pairs to be $1/N$). Therefore, we experiment with enlarging the index set $\{j \mid c_j = c_i\}$ to include more than one element or equivalently $M \neq 1$. This leads us to the following objective

$$\mathcal{L}_{\text{BIAS}}(g = g_0, M) := \mathbb{E}_{x\sim\mathcal{D}, x_j^+\sim\mathcal{D}_x^{\text{aug}}, x_i^-\sim\mathcal{D}_{\setminus x}^{\text{aug}}} \left[K(\sum_{j=1}^{M} e^{f(x)^T f(x_j^+)}, Ng_0(x, \{x_i^-\}^N))\right].$$

Finally, we challenge the specification of $q_i = 1$ and consider a synthetic data point $x' = \lambda x_i + (1-\lambda)y, y \sim \mathcal{D}$ that belongs to a synthetic class $c_{\lambda,i}$. Assume the probability x_i's label is $c_{\lambda,i}$ is $q_{\lambda,i} = \lambda + (1-\lambda)[c_y = c_i]$, then $q_{\lambda,i}$ should matches the probability $p_{\lambda,i} = \sum_{j|c_j=c_{\lambda,i}} p_{ij}$, where $\{j \mid c_j = c_{\lambda,i}\}$ is a singleton containing only the index of x', which yields[2]

$$\begin{aligned}\mathcal{L}_{\text{MIXUP}}(g = g_0, M, \lambda) :=& \mathbb{E}_{x\sim\mathcal{D}, x^+\sim\mathcal{D}_x^{\text{aug}}, x_{i_1}^-, x_{i_2}^-, x_j^-\sim\mathcal{D}_{\setminus x}^{\text{aug}}} \Big[K(e^{f(x)^T f(x^+)}, Ng_0(x, \{x_{i_1}^-\}^N)) \\ &+ \frac{\lambda}{M-1}\sum_{j=1}^{M-1} K(e^{f(x)^T f(\lambda x^+ + (1-\lambda)x_j^-)}, Ng_0(x, \{x_{i_2 j}^-\}_{i_2}^N)) \\ &+ \frac{1-\lambda}{M-1}\sum_{j=1}^{M-1} K(Ng_0(x, \{x_{i_2 j}^-\}_{i_2}^N), e^{f(x)^T f(\lambda x^+ + (1-\lambda)x_j^-)})\Big].\end{aligned}$$

Interestingly, the construction of x' herein assembles the mixup (Zhang et al., 2018) philosophy in supervised learning. We therefore name this loss by $\mathcal{L}_{\text{MIXUP}}$. Notably, as the above NCA inspired contrastive losses are designed from orthogonal perspectives, they are complementary to each other.

4 AN INTEGRATED FRAMEWORK FOR CONTRASTIVE LEARNING

Building on top of our proposed NCA inspired loss (NaCl) in Section 3, we propose a novel framework that generalizes and integrates many of the existing work in contrastive learning. Our integrated contrastive learning framework has two versions, one is **IntCl** and the other is **IntNaCl**, where both of them consist of two components – a standard loss and a robustness-promoting loss. For **IntCl**, the standard loss can be existing contrastive learning loss (Chen et al., 2020a; Chuang et al., 2020; Robinson et al., 2021), whereas for **IntNaCl** we use our proposed NaCl loss as the standard loss. We will show that this new framework not only generalizes existing methods but also achieves good accuracy and robustness simultaneously. Note that as shown in Figure 1(a) and 1(b), most of the existing works have primarily focused on improving the clean downstream accuracy only, which often has an undesired robustness accuracy.

[1] Sohn (2016) briefly explores the relationship between NCA and metric learning in a supervised learning setting and optimizes a similar loss with multiple positive examples.

[2] Lee et al. (2021) augment the dataset by including similar synthetic data point. Since the perspective is different, the formed contrastive loss takes different form (cf. Eqn. 5 and 6 in the literature).

4.1 INTEGRATED CONTRASTIVE LOSS

To design a generic loss that accounts for both clean and adversarial accuracy, together with a robustness-promoting term, we utilize the NaCl developed in Section 3 to construct a contrastive learning framework, called **Int**egrated **N**eighborhood **a**nalysis **C**ontrastive **l**oss (**IntNaCl**). A general form of $\mathcal{L}_{\text{IntNalL}}$ is given by

$$\mathcal{L}_{\text{IntNaCl}}(\alpha, \mathcal{L}_{\text{Na}}(g^1, M, \lambda), \mathcal{L}_{\text{Robust}}(g^2, w)) := \mathcal{L}_{\text{Na}}(g^1, M, \lambda) + \alpha \mathcal{L}_{\text{Robust}}(g^2, w), \tag{10}$$

where $\mathcal{L}_{\text{Na}}(g^1, M, \lambda)$ can be chose from $\{\mathcal{L}_{\text{VAR}}(g^1, M), \mathcal{L}_{\text{BIAS}}(g^1, M), \mathcal{L}_{\text{MIXUP}}(g^1, M, \lambda)\}$ and

$$\mathcal{L}_{\text{Robust}}(g^2, w) := \mathbb{E}\Big[K(e^{f(x)^T f(x^{\text{adv}})}, Ng^2(x, \cdot))w(x)\Big].$$

The variable g^1 and g^2 allow us to pick the estimators from $\{g_0, g_1, g_2\}$ and $w(x)$ facilitates goal-specific weighting scheme. Furthermore, we remark that as $\mathcal{L}_{\text{VAR}}$, $\mathcal{L}_{\text{BIAS}}$, and $\mathcal{L}_{\text{MIXUP}}$ all reduce to one same form when $M = 1$, we denote the $\mathcal{L}_{\text{IntNaCl}}$ under these cases by **Int**egrated **C**ontrastive **l**oss (**IntCl**):

$$\mathcal{L}_{\text{IntCL}}(\alpha, g^1, g^2, w) := \mathbb{E}\Big[K(e^{f(x)^T f(x^+)}, Ng^1(x, \cdot)) + \alpha K(e^{f(x)^T f(x^{\text{adv}})}, Ng^2(x, \cdot))w(x)\Big], \tag{11}$$

where we show many of the existing works are a special case:

$$\begin{cases} \alpha = 0, g^1 = g_0, & \mathcal{L}_{\text{SimCLR}} \text{ (Chen et al., 2020a) (i.e. Equation 3);} \\ \alpha = 0, g^1 = g_1, & \mathcal{L}_{\text{Debiased}} \text{ (Chuang et al., 2020) (i.e. Equation 4);} \\ \alpha = 0, g^1 = g_2, & \mathcal{L}_{\text{Debiased+HardNeg}} \text{ (Robinson et al., 2021) (i.e. Equation 5);} \\ \alpha = 1, g^1 = g_0, g^2 = g_0, w(x) \equiv 1, & \mathcal{L}_{\text{Adv}} \text{ (Ho \& Vasconcelos, 2020) (i.e. Equation 6).} \end{cases}$$

4.2 ADVERSARIAL WEIGHTING

Weighting sample loss based on their margins has been proven to be effective in the adversarial training under supervised settings (Zeng et al., 2020). Specifically, it is argued that training points that are closer to the decision boundaries should be given more weight in the supervised loss. While the margin of a sample in supervised settings is well-defined, it is underdefined in unsupervised settings. To tackle this, we borrow the intelligence from Ho & Vasconcelos (2020) and mimic how the authors transfer the definition of adversarial examples in supervised learning to unsupervised learning. Specially, we see that as an adversarial example in supervised learning is defined by a perturbed sample that has a zero margin to the decision boundary, authors of Ho & Vasconcelos (2020) define adversarial example in unsupervised learning to be an augmented sample that maximizes the contrastive loss. With this, we also give our weighting function as the value of the contrastive loss $\hat{w}(x) := K(e^{f(x)^T f(x^+)}, Ng(x, \cdot))$, where the estimator g can be g_0, g_1, g_2. Using this, we see that samples that are originally hard to be distinguished from other samples (i.e. small probability) are now assigned with bigger weights.

5 EXPERIMENTAL RESULTS

5.1 EXPERIMENTAL SET-UP

Implementation details. All the proposed methods are implemented based on open source repositories provided in the literature (Chen et al., 2020a; Ho & Vasconcelos, 2020; Robinson et al., 2021). Five benchmarking contrastive losses are considered as baselines that include: $\mathcal{L}_{\text{SimCLR}}$ (Chen et al., 2020a), $\mathcal{L}_{\text{Debiased}}$ (Chuang et al., 2020), $\mathcal{L}_{\text{Debiased+HardNeg}}$ (Robinson et al., 2021), $\mathcal{L}_{\text{Adv}}$ (Ho & Vasconcelos, 2020) (i.e. Equation 3, Equation 4, Equation 5, Equation 6). We train representations on resnet18 and include MLP projection heads (Chen et al., 2020a). A batch size of 256 is used across all the experiments. Unless otherwise specified, the representation network is trained for 100 epochs. We run five independent trials for each of the experiments and report the mean and standard deviation in the entries. Throughout our experiments, no adversarial fine-tuning is performed. We implement the proposed framework using PyTorch to enable the use of an NVIDIA GeForce RTX 2080 Super GPU, two NVIDIA Tesla P100 GPUs, and four NVIDIA Tesla V100 GPUs.

Table 1: The CIFAR100 linear evaluation results (%) of NaCl on $\mathcal{L}_{\text{SimCLR}}$, $\mathcal{L}_{\text{Debiased+HardNeg}}$, and $\mathcal{L}_{\text{IntCl}}$ (ours, cf. Equation 11). The best improvement over the individual baseline is in boldface.

M	$\mathcal{L}_{\text{SimCLR}}: 53.69 \pm 0.25$			$\mathcal{L}_{\text{Debiased+HardNeg}}: 56.83 \pm 0.20$			$\mathcal{L}_{\text{IntCl}}: 56.22 \pm 0.15$		
	$\mathcal{L}_{\text{VAR}}$	$\mathcal{L}_{\text{BIAS}}$	$\mathcal{L}_{\text{MIXUP}}$	$\mathcal{L}_{\text{VAR}}$	$\mathcal{L}_{\text{BIAS}}$	$\mathcal{L}_{\text{MIXUP}}$	$\mathcal{L}_{\text{VAR}}$	$\mathcal{L}_{\text{BIAS}}$	$\mathcal{L}_{\text{MIXUP}}$
2	56.04±0.17	55.72±0.15	56.20±0.33	58.17±0.39	57.87±0.15	**60.69±2.43**	57.51±0.12	56.71±0.11	58.97±0.19
3	57.11±0.21	56.67±0.12	56.41±0.13	59.08±0.29	58.42±0.23	59.81±0.25	58.08±0.18	57.13±0.26	59.26±0.18
4	57.27±0.14	57.09±0.26	56.00±0.42	59.29±0.16	58.86±0.18	59.75±0.33	58.31±0.23	57.06±0.19	59.32±0.21
5	**57.91±0.12**	57.32±0.17	56.63±0.31	59.67±0.38	58.81±0.21	59.85±0.30	**58.64±0.24**	57.46±0.04	59.43±0.23

Evaluation protocol. In this section, we will evaluate three major properties of representation learning methods: standard discriminative power, naive transferability, and adversarial robustness. To evaluate the standard discriminative power, we train representation networks on CIFAR100 (Krizhevsky et al., 2009), freeze the network, and only fine-tune a fully-connected layer that maps representations to outputs on CIFAR100. To evaluate the transferability, we use the same representation networks as above, and only fine-tune a fully-connected layer that maps representations to outputs on CIFAR10. All the adversarial robustness evaluations are completed using the implementation provided by Wong et al. (2020). We supplement more FGSM and PGD attack results in the appendix.

5.2 Linear Evaluation of Self-supervised Representations

Improvement over baselines. In this section, we test the effectiveness of NaCl in improving the downstream CIFAR100 classification accuracy. Specifically, we consider three baseline methods: 1) $\mathcal{L}_{\text{SimCLR}}$, or equivalently $\alpha = 0, g^1 = g_0, M = 1$ in Equation 10; 2) $\mathcal{L}_{\text{Debiased+HardNeg}}$, or equivalently $\alpha = 0, g^1 = g_2, M = 1$ in Equation 10; and 3) $\mathcal{L}_{\text{IntCl}}(\alpha = 1, g^1 = g_2, g^2 = g_2, w = \hat{w})$. We list the results in Table 1. Due to page limit, we only give the results of $\mathcal{L}_{\text{MIXUP}}$ with $\lambda = 0.9$ when applied on $\mathcal{L}_{\text{SimCLR}}$, and with $\lambda = 0.5$ when applied on $\mathcal{L}_{\text{Debiased+HardNeg}}$ and $\mathcal{L}_{\text{IntCl}}$. Complete tables of results obtained with $\lambda = 0.6, 0.7, 0.8$ can be found in the appendix. By referring to the Table 1, one can see that all three NaCl losses are able to improve the standard performance upon their baselines. Among the three losses, $\mathcal{L}_{\text{VAR}}$ is generally the most successful in boosting the standard accuracy when applied to SimCLR (i.e. 57.91% vs. 57.32% / 56.63%), whereas $\mathcal{L}_{\text{MIXUP}}$ demonstrates a better ability when applied to Debiased+HardNeg (i.e. 60.69% vs. 59.67% / 58.86%). Although $\mathcal{L}_{\text{BIAS}}$ does not bring more gain in the metric of standard accuracy compared with $\mathcal{L}_{\text{MIXUP}}$ and $\mathcal{L}_{\text{VAR}}$, its improved accuracy still suggests that the bias is causing harm to the original baseline and a bias reduction scheme can do help to the training.

We have also validated the usefulness of NaCl with experiments on CIFAR10. That is, we train the representation networks on CIFAR10, freeze the network, and fine-tune a fully-connected layer that maps representations to outputs on CIFAR10. In HaoChen et al. (2021), $\mathcal{L}_{\text{SimCLR}}$ is reported to give 83.73% standard accuracy with 200 training epochs. As a comparison, $\mathcal{L}_{\text{IntNaCl}}(1, \mathcal{L}_{\text{MIXUP}}(g_2, 2, 0.5), \mathcal{L}_{\text{Robust}}(g_2, \hat{w}))$ gives 84.62% after 100 training epochs and 86.69% after 200 training epochs, demonstrating a clear improvement over the baseline.

Accuracy on larger epochs. As training the representation with more epochs can also expose the data to more augmentations, we carry out an additional experiments to compare the efficiency and effectiveness of baseline methods with significant more training epochs. Specially, HaoChen et al. (2021) has reported a $\mathcal{L}_{\text{SimCLR}}$ CIFAR100 accuracy of 54.74% after 200 epochs, compared to $\mathcal{L}_{\text{VAR}}(g_0, 2)$'s 56.04% after 100 epochs. In our reproduction of the $\mathcal{L}_{\text{SimCLR}}$ 200-epoch result, we have witnessed an accuracy of 57.45% however at the cost of 1.34X training time (cf. 200 epochs with $\mathcal{L}_{\text{SimCLR}}$ takes 211 mins vs. 100 epochs with $\mathcal{L}_{\text{VAR}}(g_0, 2)$ takes 158 mins). Additionally, we see $\mathcal{L}_{\text{SimCLR}}$ reaches 61.90% and Debiased+HardNeg stops at 62.74%, while $\mathcal{L}_{\text{VAR}}(g_0, 2)$ and $\mathcal{L}_{\text{VAR}}(g_2, 2)$ improve upon them individually by reaching 62.37% and 63.51%. We refer the readers to the appendix for detailed linear evaluation results on CIFAR100 with extended training epochs.

5.3 Evaluation of Model Robustness and Transferability

Robustness. In addition to the discriminative power, we also want to empower the learned representation with strong adversarial performance. In Table 2, we list the classification accuracy on CIFAR100 under FGSM attacks with magnitude $\epsilon = 0.002$. From the table, one can see that, with the absence of robustness promoting loss ($\alpha = 0$), all NaCl methods manage to improve upon the

Table 2: The CIFAR100 adversarial evaluation results (%) of NaCl on $\mathcal{L}_{\text{SimCLR}}$, $\mathcal{L}_{\text{Debiased+HardNeg}}$, and $\mathcal{L}_{\text{IntCl}}$ (ours, cf. Equation 11). The best improvement over the individual baseline is in boldface.

M	$\mathcal{L}_{\text{SimCLR}}: 25.17 \pm 0.55$			$\mathcal{L}_{\text{Debiased+HardNeg}}: 31.03 \pm 0.41$			$\mathcal{L}_{\text{IntCl}}: 40.05 \pm 0.67$		
	$\mathcal{L}_{\text{VAR}}$	$\mathcal{L}_{\text{BIAS}}$	$\mathcal{L}_{\text{MIXUP}}$	$\mathcal{L}_{\text{VAR}}$	$\mathcal{L}_{\text{BIAS}}$	$\mathcal{L}_{\text{MIXUP}}$	$\mathcal{L}_{\text{VAR}}$	$\mathcal{L}_{\text{BIAS}}$	$\mathcal{L}_{\text{MIXUP}}$
2	27.19±0.79	27.04±0.45	30.95±0.36	31.92±0.45	32.50±0.48	32.22±0.35	41.01±0.36	39.80±0.57	40.25±0.52
3	27.39±0.36	28.41±0.24	**30.98±0.90**	32.63±0.74	**33.19±0.60**	32.04±0.67	41.02±0.83	40.53±0.29	40.96±0.58
4	27.63±0.78	28.20±0.81	29.90±0.63	32.48±0.62	32.65±1.07	32.03±0.34	**41.49±0.51**	40.85±0.31	40.82±0.54
5	28.37±0.56	28.33±0.59	30.58±0.52	33.10±0.71	32.86±0.47	32.06±0.72	40.50±0.23	41.00±0.86	41.01±0.34

Table 3: The CIFAR10 transfer evaluation results (%) of NaCl on $\mathcal{L}_{\text{SimCLR}}$, $\mathcal{L}_{\text{Debiased+HardNeg}}$, and $\mathcal{L}_{\text{IntCl}}$ (ours, cf. Equation 11). The best improvement over the individual baseline is in boldface.

M	$\mathcal{L}_{\text{SimCLR}}: 76.34 \pm 0.28$			$\mathcal{L}_{\text{Debiased+HardNeg}}: 77.24 \pm 0.29$			$\mathcal{L}_{\text{IntCl}}: 76.39 \pm 0.10$		
	$\mathcal{L}_{\text{VAR}}$	$\mathcal{L}_{\text{BIAS}}$	$\mathcal{L}_{\text{MIXUP}}$	$\mathcal{L}_{\text{VAR}}$	$\mathcal{L}_{\text{BIAS}}$	$\mathcal{L}_{\text{MIXUP}}$	$\mathcal{L}_{\text{VAR}}$	$\mathcal{L}_{\text{BIAS}}$	$\mathcal{L}_{\text{MIXUP}}$
2	77.32±0.14	77.40±0.14	76.96±0.15	77.43±0.18	77.43±0.11	79.36±0.65	76.88±0.49	76.55±0.27	78.61±0.20
3	78.02±0.27	77.53±0.24	77.10±0.21	77.87±0.29	77.41±0.17	79.41±0.17	76.95±0.19	76.67±0.22	**78.83±0.22**
4	77.91±0.29	77.75±0.22	77.11±0.40	77.92±0.17	77.46±0.29	79.42±0.18	77.30±0.30	76.34±0.22	78.83±0.27
5	**78.09±0.29**	77.93±0.40	77.04±0.19	78.04±0.09	77.58±0.23	**79.45±0.20**	77.42±0.17	76.60±0.37	78.80±0.21

baselines. Notably, by referring to the full table in the appendix, one will see that $\mathcal{L}_{\text{MIXUP}}$ with $\lambda = 0.9$ boosts the CIFAR100 adversarial accuracy to 34.65% when applied to Debiased+HardNeg. That said, although $\mathcal{L}_{\text{VAR}}$ and $\mathcal{L}_{\text{BIAS}}$ are both useful in enhancing the adversarial performance, $\mathcal{L}_{\text{MIXUP}}$ improves the baseline by the largest margin.

When we explicitly regularize the adversarial robustness performance ($\alpha \neq 0$), the representation network learned via $\mathcal{L}_{\text{IntCl}}$ yields an adversarial accuracy of 40.05%. When we strengthen the loss with NaCl. $\mathcal{L}_{\text{IntNaCl}}$, the adversarial performance is further improved.

Transferability. We validate the transferability of all the methods by fine-tuning a fully-connected layer that maps representations to outputs on CIFAR10. This is in analogy to the evaluation procedure in Chen et al. (2020a) - train a fixed feature extractor on a large-scale dataset and train a linear classifier on top of the frozen base network with smaller-scale datasets. Table 3 shows that besides decent standard and adversarial performance, NCA inspired losses can also improve the transferability. We refer the readers to the appendix for the transfer robustness results.

Comparison to other methods. In Figure 1, we compare the standard and transfer accuracy of various benchmark methods. Specially, we plot the adversarial accuracy defined under FGSM attacks (Goodfellow et al., 2015) along the y-axis. In essence, one will want a representation network that pushes the performance to the upper-right corner in the 2D accuracy grid (standard-adversarial accuracy plot). From the figure, we see that SimCLR, Debiased, HardNeg, and Debiased+HardNeg all score relatively poorly, obtaining an adversarial accuracy of around or below 30% on CIFAR100 and 50% on CIFAR10. One exception, Adv, performs adequately and reaches an accuracy of more than 35% on CIFAR100 and 55% on CIFAR10, while sacrificing the standard accuracy. We highlight the results of $\mathcal{L}_{\text{IntNaCl}}$ and $\mathcal{L}_{\text{IntCl}}$ in circles, through which we see that while $\mathcal{L}_{\text{IntCl}}$ can already train representations that are decently robust without sacrificing the standard accuracy on CIFAR100, the transfer accuracy on CIFAR10 is inferior to some baselines (HardNeg and Debiased+HardNeg). Comparatively, $\mathcal{L}_{\text{IntNaCl}}$ wins over the baselines by a large margin on both datasets, proving the ability of learning representation networks that also transfer robustness property. We refer the readers to Figure S2 in the appendix for a detailed analysis of the effect of λ on these different metrics.

6 Conclusion

In this paper, we discover the relationship between contrastive loss and Neighborhood Component Analysis (NCA), which motivates us to generalize the existing contrastive loss to a set of Neighborhood analysis Contrastive losses (NaCl). We further propose a generic contrastive learning framework based on NaCl, which learns representations that score high in both standard accuracy and adversarial accuracy in downstream tasks. Future work includes addressing the current limitation of assuming $k = 1$ for k-nearest neighbor in NCA to $k > 1$ (Tarlow et al., 2013), by doing which we expect to extend the current framework to an even more general form.

REFERENCES

Piotr Bojanowski and Armand Joulin. Unsupervised learning by predicting noise. In *ICML*, pp. 517–526. PMLR, 2017.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, volume 33, pp. 9912–9924, 2020.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1597–1607, Virtual, 13–18 Jul 2020a. PMLR.

Xinlei Chen, Haoqi Fan, Ross B. Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *CoRR*, abs/2003.04297, 2020b. URL https://arxiv.org/abs/2003.04297.

Ching-Yao Chuang, Joshua Robinson, Lin Yen-Chen, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. *arXiv preprint arXiv:2007.00224*, 2020.

Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(9):1734–1747, 2015.

Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations, 2021.

Jacob Goldberger, Geoffrey E Hinton, Sam Roweis, and Russ R Salakhutdinov. Neighbourhood components analysis. *NeurIPS*, 17:513–520, 2004.

I. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

Sven Gowal, Po-Sen Huang, Aaron van den Oord, Timothy Mann, and Pushmeet Kohli. Self-supervised adversarial robustness for the low-label, high-data regime. In *ICLR*, 2020.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent - a new approach to self-supervised learning. In *NeurIPS*, volume 33, pp. 21271–21284, 2020.

Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 297–304. JMLR Workshop and Conference Proceedings, 2010.

Jeff Z HaoChen, Colin Wei, Adrien Gaidon, and Tengyu Ma. Provable guarantees for self-supervised deep learning with spectral contrastive loss. *arXiv preprint arXiv:2106.04156*, 2021.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, June 2020.

Chih-Hui Ho and Nuno Vasconcelos. Contrastive learning with adversarial examples. *arXiv preprint arXiv:2010.12050*, 2020.

Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*, 2016.

Harini Kannan, Alexey Kurakin, and Ian J. Goodfellow. Adversarial logit pairing. *CoRR*, abs/1803.06373, 2018. URL http://arxiv.org/abs/1803.06373.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*, 2020.

Minseon Kim, Jihoon Tack, and Sung Ju Hwang. Adversarial self-supervised contrastive learning. *arXiv preprint arXiv:2006.07589*, 2020.

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

Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. Adversarial machine learning at scale. In *ICLR*, 2016.

Kibok Lee, Yian Zhu, Kihyuk Sohn, Chun-Liang Li, Jinwoo Shin, and Honglak Lee. i-mix: A domain-agnostic strategy for contrastive representation learning. In *ICLR*, 2021.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

Andriy Mnih and Yee Whye Teh. A fast and simple algorithm for training neural probabilistic language models. In *ICML*, 2012.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *CVPR*, pp. 2574–2582, 2016.

A. Nguyen, J. Yosinski, and J. Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *CVPR*, 2015.

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

Joshua David Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. In *ICLR*, 2021.

Nikunj Saunshi, Orestis Plevrakis, Sanjeev Arora, Mikhail Khodak, and Hrishikesh Khandeparkar. A theoretical analysis of contrastive unsupervised representation learning. In *ICML*, pp. 5628–5637, 2019.

Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *NeurIPS*, volume 29, 2016.

C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *ICLR*, 2014.

Daniel Tarlow, Kevin Swersky, Laurent Charlin, Ilya Sutskever, and Rich Zemel. Stochastic k-neighborhood selection for supervised and unsupervised learning. In *ICML*, 2013.

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? In *NeurIPS*, volume 33, pp. 6827–6839, 2020.

Eric Wong, Leslie Rice, and J. Zico Kolter. Fast is better than free: Revisiting adversarial training. In *ICLR*, 2020.

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, pp. 3733–3742, 2018.

Huimin Zeng, Chen Zhu, Tom Goldstein, and Furong Huang. Are adversarial examples created equal? a learnable weighted minimax risk for robustness under non-uniform attacks. *arXiv preprint arXiv:2010.12989*, 2020.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018.