DEEP CONTRASTIVE LEARNING APPROXIMATES ENSEMBLES OF ONE-CLASS SVMS WITH NEURAL TANGENT KERNELS

Anonymous authors
Paper under double-blind review

ABSTRACT

To demystify the (self-supervised) contrastive learning in representation learning, in the paper we show that a model learned by deep contrastive learning with a family of loss functions such as InfoNCE essentially approximates an ensemble of one-class support vector machines (SVMs) with neural tangent kernels (NTKs). This result comes from the fact that each gradient for network weight update in contrastive learning can be interpreted approximately as the primal solution for a one-class SVM with contrastive gradients as input. From the dual perspective, the Lagrange multipliers provide unique insights into the importance of the anchor-positive-negative triplet samples. In this way, we further propose a novel sequential convex programming (SCP) algorithm for contrastive learning, where each sub-problem is a one-class SVM. Empirically we demonstrate that our approach can learn better gradients than conventional contrastive learning approaches that significantly improve performance.

1 INTRODUCTION

Recently, self-supervised representation learning has drawn a great attention due to its potential of alleviating human annotations for a large amount of data. Specifically, contrastive learning (Chopra et al., 2005; Hadsell et al., 2006) has become the dominant method in self-supervised learning and has shown competitive performance over its supervised counterpart on several downstream tasks such as classification, object detection, and segmentation (Oord et al., 2018; Jaiswal et al., 2020; Deng, 2009; Misra & Maaten, 2020; He et al., 2020; Everingham et al., 2010; Güler et al., 2018; He et al., 2017; Lin et al., 2014; Faster, 2015).

Typically, given an anchor x, contrastive learning takes augmented views of the same data as positive pairs (x, x^+), and other data in the same batch as negative pairs (x, x^-). The contrastive representation learning attempts to pull the embeddings of positive pairs closer and push the embeddings of negative pairs away in the latent space by optimizing the objective such as the InfoNCE loss (Oord et al., 2018; Chen et al., 2020a). Data augmentation (by augmentation we mean any data transformation, multi-view, or sampling strategies) plays an important role in contrastive learning and is attracting more and more attention recently. The positive augmentation has been studied intensively (Blum & Mitchell, 1998; Xu et al., 2013; Bachman et al., 2019; Chen et al., 2020a; Tian et al., 2020b; Chen et al., 2020c; Tian et al., 2020a; Srinivas et al., 2020; Logeswaran & Lee, 2018; Oord et al., 2018; Purushwalkam & Gupta, 2020; Sermanet et al., 2018), as well as the negative data augmentation *e.g.,* (Kalantidis et al., 2020; Ge et al., 2021; Robinson et al., 2021; Sinha et al., 2021) where most of the works focus on "hard" negative data augmentation. For instance, Robinson et al. (2021) provided a popular principle that "*The most useful negative samples are ones that the embedding currently believes to be similar to the anchor*", where the embedding refers to the network output. That is, letting x_1^-, x_2^- be two negative samples *w.r.t.* the anchor x and ϕ be the current network, then x_1^- is more useful (*i.e.,* harder) than x_2^- if $\phi(x)^T\phi(x_1^-) > \phi(x)^T\phi(x_2^-)$ holds, where $(\cdot)^T$ denotes the matrix transpose operator. However,

Do such "harder" negatives really help more in contrastive learning? To answer this question, we did a simple experiment to validate it. We trained ResNet-18 (He et al., 2016) as the backbone network using SimCLR (Chen et al., 2020a) on CIFAR-10 (Krizhevsky et al., 2009), with no negative

data augmentation as a baseline. Then we used the learned network to evaluate the cosine similarity between the output features from an anchor and different negative samples and illustrated their probability distribution in Fig. 1. Based on the popular principle, in general, the negative samples from the dataset are "harder" than the negatives augmented by non-semantic negatives (NSN) (Ge et al., 2021) which are "harder" than purely random Gaussian noise as negative augmentation. Surprisingly, however, with the same sufficient amount of augmented samples from either NSN or Gaussian noise, we observed that both well-trained models can achieve very similar accuracy results that outperform the baseline. This clearly contradicts the principle, because the augmented negatives as weak as Gaussian noise can also help contrastive learning. Then,

How to define the "hardness" for negative data? Recall that contrastive learning aims to learn such an embedding space where similar sample pairs stay close to each other while dissimilar ones are far apart. Therefore, the hardness of a negative sample becomes meaningless without taking its context into account which includes the anchor as well as positives and other (seen) negatives. In fact, we say that a negative x^- will be useful in contrastive learning, regardless of its hardness, as long as there exists a triplet sample (x, x^+, x^-) whose gradient helps reduce the contrastive loss. From this view, the *triplet importance* may be more appropriate to measure in contrastive learning which can be used to represent the hardness of the negative, to a certain degree.

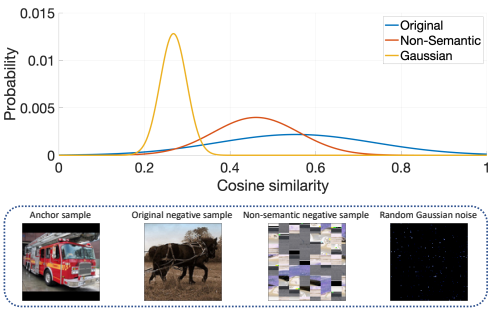

Figure 1: Illustration of probability distributions over cosine similarity between the anchor and negatives. Both negative data augmentation approaches can improve the SimCLR baseline classification accuracy by $\sim 7\%$ on CIFAR-10.

So how shall we measure "triplet importance"? To answer this question, in this paper we try to understand the effect of positive and negative samples on the network weight update through gradients in contrastive learning. We show that the gradient of a family of loss functions such as InfoNCE can be taken as an approximate primal solution for one-class SVMs (Schölkopf et al., 1999) with specific neural tangent kernels (NTKs) (Jacot et al., 2018). In this way, the Lagrange multipliers from the SVMs can be interpreted as the importance of such triplet samples. This analysis leads us to a conclusion that a model learned by deep contrastive learning essentially can be viewed approximately as an ensemble of one-class SVMs with NTKs.

We are aware that very recently Tian (2022) proposed interpreting contrastive learning from the perspective of feature composition as a minmax problem, and showed in particular that deep linear networks contrastive learning is equivalent to Principal Component Analysis (PCA). In contrast, we interpret contrastive learning as a *Sequential Convex Programming* (SCP) problem where each sub-problem is a one-class SVM. Besides, our approach has no restrictions on network architectures. Such differences distinguish our work dramatically from the current literature on contrastive learning.

Contributions. In summary, our main contributions are listed as follows:

- Theoretically, we are the *first*, to the best of our knowledge, to interpret contrastive learning from the perspective of one-class SVMs with NTKs, whose Lagrange multipliers indicate the importance of triplet samples. This results in a novel SCP formulation for contrastive learning.
- Empirically, we demonstrate that our SCP approach can learn better gradients than conventional approaches that significantly improve the performance for contrastive learning.

2 Related Work

Contrastive Learning. Recently, learning representations from unlabeled data in contrastive way Chopra et al. (2005); Hadsell et al. (2006) has been one of the most competitive research field (Oord et al., 2018; Hjelm et al., 2018; Wu et al., 2018; Tian et al., 2020a; Sohn, 2016; Chen et al., 2020a; Jaiswal et al., 2020; Li et al., 2020b; He et al., 2020; Chen et al., 2020c;b; Bachman et al., 2019; Misra & Maaten, 2020; Caron et al., 2020). Popular model structures like SimCLR (Chen et al., 2020a) and Moco (He et al., 2020) apply the commonly used loss function InfoNCE (Oord et al.,

2018) to learn latent representation that is beneficial to downstream tasks. Several theoretical studies show that contrastive loss optimizes data representations by aligning the same image's two views (positive pairs) while pushing different images (negative pairs) away on the hypersphere (Wang & Isola, 2020; Chen et al., 2021; Wang & Liu, 2021; Arora et al., 2019). (Arora et al., 2019) Though the existing works try to understand the properties and explain the behavior of contrastive learning, ours is the first work that bridge the connection between contrastive learning and one-class SVMs. The connections between the two help solve some of the key challenges regarding negative data augmentation in contrastive learning.

Data Augmentation. Empirically, the positive pairs could be different modalities of a signal (Arandjelovic & Zisserman, 2018; Tian et al., 2020a; Tschannen et al., 2020) or different data augmentations of the same image, *e.g.,* color distortion, random crop (Chen et al., 2020a;c; Grill et al., 2020). (Tian et al., 2020b) suggested generating the positive pairs with "InfoMin principle" so that the generated positive pairs maintain the minimal information necessary for the downstream tasks. (Selvaraju et al., 2021; Peng et al., 2022; Mishra et al., 2021; Li et al., 2022) proposed selecting meaningful but not fully overlapped contrastive crops with guidance like attention maps or object-scene relations. (Shen et al., 2020) empirically demonstrated that introducing extra convex combinations of data as positive augmentation improves the representation learning. Similar mixing data strategies could be found in (Lee et al., 2020; Kim et al., 2020; Verma et al., 2021; Li et al., 2020a; Ren et al., 2022). Other than exploring positive augmentation, a few recent works also focus on negative data selection in contrastive learning. Typically, negative samples are drawn uniformly from the training data. Basing on the argument that not all negative are true negatives, (Chuang et al., 2020; Robinson et al., 2020) developed debiased contrastive loss to assign higher weights to the hard negative samples. (Wang & Liu, 2021) proposed an explicit way to select the hard negative samples that are similar to the positive. To provide more meaningful negative samples, (Kalantidis et al., 2020) studied the Mixup (Zhang et al., 2017) strategy in latent space to generate hard negatives. (Hu et al., 2021) proposed to learn a set of negative adversaries directly. (Ge et al., 2021) generated negative samples by texture synthesis or selecting non-semantic patches from existing images. Different from previous studies, we do not propose a new method for negative data augmentation, but provide some insights on the real "hard" negatives from the perspective of the gradients of contrastive loss.

One-Class SVMs. The traditional classification algorithm is designed to classify the test data into two or more classes after training the classifier on the training set. When considering one-class learning *e.g.,* outlier detection, anomaly detection, novelty detection (Moya & Hush, 1996), the ultimate goal of a classifier becomes detecting whether the test data belong to the same distribution of training set. One-class support-vector machines (SVMs) (Schölkopf et al., 1999; Tax & Duin, 1999; Sain, 1996; Schölkopf et al., 2001; Tax & Duin, 2004; Tax, 2002), a classical one-class learning algorithm, are frequently used in outlier or novelty detection (Pimentel et al., 2014; Chandola, 2007; Ratsch et al., 2002). In general, the method of (Schölkopf et al., 1999; 2001) learns to separate the transformed training data from the origin with maximum margin using hyperplane in the feature space corresponding to a kernel. When estimating a region that contains a large fraction of the training data, the algorithm uses a parameter to decide the tolerance of outliers in the "normal" training data. Instead of using hyperplane, another related approach proposed by (Tax & Duin, 1999) minimizes the volume of a hypersphere that contains as many as possible of the "normal" training data. For certain kernels like Gaussian radial basis function (RBF), the hypersphere one-class SVM has been shown to be equivalent to (Schölkopf et al., 2001).

Sequential Convex Programming. Sequential convex programming (SCP) is a classic technique to iteratively optimize local convex approximations of a non-convex function Boyd et al. (2004); Duchi (2018). SCP has been widely studied in practice for different applications such as trajectory optimization and optimal control systems Augugliaro et al. (2012); Morgan et al. (2014); Bonalli et al. (2019); Wang & Grant (2017). More details can be found in a recent survey (Messerer et al., 2021).

3 Approach

3.1 Preliminaries

Notations. In the sequel, we denote $x, x^+, x^- \in \mathcal{X}$ as the anchor, its positive and negative samples, respectively, x' as an augmented (could be either positive or negative) sample of x, $h(x;\omega) : \mathcal{X} \to \mathbb{R}^d$

Table 1: List of contrastive gradient coefficients, α_{x^-}, for different contrastive losses.

Contrastive Loss	Analytic Solution for α_{x^-}
InfoNCE (Oord et al., 2018)	$\frac{\exp\left\{\frac{1}{\tau}f(x,x^+,x^-;\omega)\right\}}{\epsilon+\sum_{x^-}\exp\left\{\frac{1}{\tau}f(x,x^+,x^-;\omega)\right\}}$
MINE (Belghazi et al., 2018)	$\frac{\exp\{f(x,x^+,x^-;\omega)\}}{\sum_{x^-}\exp\{f(x,x^+,x^-;\omega)\}}$
Soft Triplet (Tian et al., 2020c)	$\frac{\exp\{\frac{1}{\tau}f(x,x^+,x^-;\omega)\}}{\exp\{-\epsilon\}+\sum_{x^-}\exp\{\frac{1}{\tau}f(x,x^+,x^-;\omega)\}}$
$N+1$ Tuplet (Sohn, 2016)	$\frac{\exp\{f(x,x^+,x^-;\omega)\}}{1+\sum_{x^-}\exp\{f(x,x^+,x^-;\omega)\}}$
Triplet (Schroff et al., 2015)	$\mathbf{1}_{\{f(x,x^+,x^-;\omega)+\epsilon>0\}}$
Lifted Structured (Oh Song et al., 2016)	$\mathbf{1}_{\{\beta=\log\{\sum_{x^-}\exp\{f(x,x^+,x^-;\omega)+\epsilon\}\}>0\}}\frac{2\beta\exp\{f(x,x^+,x^-;\omega)\}}{\sum_{x^-}\exp\{f(x,x^+,x^-;\omega)\}}$
Modified Triplet (Coria et al., 2020)	$c\sigma(cf(x,x^+,x^-;\omega))(1-\sigma(cf(x,x^+,x^-;\omega)))$, σ: sigmoid
Triplet Contrastive (Ji et al., 2021)	Constant

as a twice-differentiable function represented by a deep neural network and parametrized by $\omega \in \Omega$, $s(\cdot,x;\omega) = h(\cdot;\omega)^T h(x;\omega)$ as a similarity score, $d(\cdot,x;\omega) = \|h(\cdot;\omega) - h(x;\omega)\|_2$ as the Euclidean distance, $f(x,x^+,x^-;\omega) = \frac{1}{2}\Big[d(x^+,x;\omega)^2 - d(x^-,x;\omega)^2\Big]$, and $\|\cdot\|_2, \nabla, (\cdot)^T$ as the ℓ_2 norm, the gradient operator (*w.r.t.* ω by default), and the matrix transpose operator, respectively.

InfoNCE Loss. It is defined as $\ell_{NCE} = -\sum_x \tau \log \frac{\exp\left\{\frac{1}{\tau}s(x^+,x;\omega)\right\}}{\epsilon\exp\left\{\frac{1}{\tau}s(x^+,x;\omega)\right\}+\sum_{x^-}\exp\left\{\frac{1}{\tau}s(x^-,x;\omega)\right\}} = \sum_x \tau \log\left[\epsilon + \sum_{x^-}\exp\left\{\frac{1}{\tau}[s(x^-,x;\omega) - s(x^+,x;\omega)]\right\}\right]$, where τ is the temperature that controls the sharpness and $\epsilon \geq 0$ is a predefined constant. Note that $\epsilon = 1$ has been used in many works such as SimCLR (Chen et al., 2020a) and MoCo (He et al., 2020; Tian et al., 2020a), and $\epsilon = 0$ was used in decoupled contrastive learning (DCL) (Yeh et al., 2021). Now we can easily write down its gradient, $\nabla\ell_{NCE}(\omega) = \sum_x \nabla\ell_{NCE}(x;\omega)$ with $\nabla\ell_{NCE}(x;\omega) = \sum_{x^-} p(x^-)\nabla\Big[s(x^-,x;\omega) - s(x^+,x;\omega)\Big]$ with $p(x^-) = \frac{\exp\left\{\frac{1}{\tau}[s(x^-,x;\omega)-s(x^+,x;\omega)]\right\}}{\epsilon+\sum_{x^-}\exp\left\{\frac{1}{\tau}[s(x^-,x;\omega)-s(x^+,x;\omega)]\right\}} \in [0,1]$. Then stochastic gradient descent (SGD) can be used to update network weights.

(ϕ,ψ)-Family of Contrastive Loss. Tian (2022) defined a family of contrastive loss functions as $\ell_{\phi,\psi} = \sum_x \phi\left(\sum_{x^-}\psi\left(f(x,x^+,x^-;\omega)\right)\right)$, where ϕ,ψ are monotonously increasing and differentiable scalar functions, that generalizes several different contrastive losses including InfoNCE (*i.e.,* ϕ is the log function with ϵ, and ψ is the exp function with τ and $\|h(x;\omega)\|_2 = \|h(x^+;\omega)\|_2 = \|h(x^-;\omega)\|_2$). Similarly, its gradient can be written down as $\nabla\ell_{\phi,\psi}(\omega) = \sum_x \nabla\ell_{\phi,\psi}(x;\omega)$ with

$$\nabla\ell_{\phi,\psi}(x;\omega) = \phi'_x \sum_{x^-}\psi'_{x^-}\nabla\Big[d(x^+,x;\omega) - d(x^-,x;\omega)\Big] = \sum_{x^-}\alpha_{x^-}\nabla\Big[d(x^+,x;\omega) - d(x^-,x;\omega)\Big], \tag{1}$$

where $\phi'_x, \psi'_{x^-} \geq 0$ denote the first order derivatives of ϕ,ψ given the corresponding data and ω and $\alpha_{x^-} = \phi'_x\psi'_{x^-} \geq 0$. Table 1 lists some examples for α_{x^-} where $\mathbf{1}_{\{\cdot\}}$ denotes an indication function returning 1 if the condition holds, otherwise 0. For the first four losses in Table 1, we can see that each $\alpha_{x^-} \geq 0$ and $\sum_{x^-}\alpha_{x^-} \leq 1$, and for the last four losses, α_{x^-} is upper bounded (and so is $\sum_{x^-}\alpha_{x^-}$ that can be rescaled to $[0,1]$). All such losses can be minimized based on Eq. 1 using SGD.

One-Class SVM. Schölkopf et al. (1999) proposed a one-class SVM with the primal formulation as

$$\min_{\mathbf{w},\boldsymbol{\xi},\rho}\frac{1}{2}\|\mathbf{w}\|_2^2 + \frac{1}{\nu N}\sum_{i=1}^N \xi_i - \rho,\ \text{s.t.}\ \mathbf{w}^T\mathbf{x}_i \geq \rho - \xi_i, \xi_i \geq 0, \forall i \in [N], \tag{2}$$

where $\mathbf{x}_i$ is the i-th input vector, $\mathbf{w},\rho$ are the model parameters, $\nu \in (0,1)$ is a predefined trade-off scalar, and ξ_i is a slack variable. The corresponding dual form is

$$\min_{\boldsymbol{\alpha}}\frac{1}{2}\sum_{i,j}\alpha_i\alpha_j\mathbf{x}_i^T\mathbf{x}_j,\ \text{s.t.}\ 0 \leq \alpha_i \leq \frac{1}{\nu N}, \sum_i \alpha_i = 1, \forall i \quad\Longrightarrow\quad \mathbf{w} = \sum_i \alpha_i\mathbf{x}_i, \tag{3}$$

where each $\{\alpha_i\}$ denotes the Lagrange multipliers in the dual. By comparing Eq. 3 with Eq. 1, we can see that if we set $\alpha_i \leftarrow \alpha_{x^-}, \mathbf{x}_i \leftarrow \nabla\Big[d(x^+, x; \omega) - d(x^-, x; \omega)\Big]$, the gradient can be represented as a solution of the one-class SVM, provided that α_{x^-} satisfies the conditions in the SVM.

3.2 Sequential Convex Programming for Contrastive Learning

In this section, we will try to build the connections between contrastive learning and one-class SVMs.

Our Learning Objective. To simply our notations, we will refer to $f_i(\omega_t) \equiv f(x, x^+, x^-; \omega_t)$ where i denotes the index of triplet (x, x^+, x^-) in the batch. In order to match the contrastive gradients in Eq. 1 and α_{x^-} in Table 1, we define our objective as

$$\min_{\omega} \sum_i f_i(\omega) \Leftrightarrow \min_{\omega, \xi_i \geq 0} \sum_i \xi_i, \text{s.t.} f_i(\omega) \leq \xi_i - \rho, \tag{4}$$

where it holds that $f_i(\omega) + \rho \geq 0, \forall i, \forall \omega, \exists \rho \in \mathbb{R}$.

Linear Approximation & Network Update. The key idea in SCP is a locally linear approximation as

$$f_i(\omega_{t+\frac{1}{2}}) \approx f_i(\omega_t) - \nabla f_i(\omega_t)^T \Delta\omega_t \tag{5}$$

with $\omega_{t+\frac{1}{2}} = \omega_t - \Delta\omega_t$. To further reduce the approximation error, we utilize the linear interpolation (Powell, 1998) between ω_t and $\omega_{t+\frac{1}{2}}$, leading to $\omega_{t+1} = \omega_t - \eta_t \Delta\omega_t, \eta_t \in [0, 1]$ for network update.

Algorithm 1 Stochastic SCP with One-Class SVMs for Contrastive Learning

Input : triplet dataset $\mathcal{D} = \{(x, x^+, x^-)\}$, functions $\{f_i\}$, hyperparameters C and $\{\eta_t\}$
Output : network weights ω^*

Initialize network weights ω_0;
for $t = 0, \cdots, T-1$ **do**
  Select a batch of triplets from $\mathcal{D}$ uniformly at random;
  $\{\alpha_i^*\} \leftarrow$ OneClassSVM$(\{f_i\}, \{\nabla f_i\}, C)$ in Eq. 7;
  $\Delta\omega_t^* \leftarrow \sum_i \alpha_i^* \nabla f_i(\omega_t)$ in Eq. 8;
  $\omega_{t+1} \leftarrow \omega_t - \eta_t \Delta\omega_t^*$;
end
return $\omega^* \leftarrow \omega_T$;

Sequential Convex Programming with One-Class SVMs. Our basic idea in SCP is to control the loss reduction through the term $\nabla f_i(\omega_t)^T \Delta\omega_t$ for each triplet by sequentially approximating the loss landscapes locally with one-class SVMs. By comparing Eq. 4 with Eq. 2, one of the key differences is the regularization in the objective function. In order to connect one-class SVMs with the gradients in Eq. 1, we also introduce the same regularization into Eq. 4, and based on the linear approximation in Eq. 5 further propose a new family of one-class SVMs for contrastive learning as follows:

$$\min_{\Delta\omega_t, \xi_i \geq 0, \rho} \frac{1}{2}\|\Delta\omega_t\|_2^2 + C\sum_i \xi_i - \rho, \text{ s.t. } f_i(\omega_t) - \nabla f_i(\omega_t)^T \Delta\omega_t \leq \xi_i - \rho, \forall i \tag{6}$$

$$\xrightleftharpoons[primal]{dual} \max_{0 \leq \alpha_{i,t} \leq C} \sum_i \alpha_{i,t} f_i(\omega_t) - \frac{1}{2}\sum_{i,j} \alpha_{i,t}\alpha_{j,t} \nabla f_i(\omega_t)^T \nabla f_j(\omega_t), \text{ s.t. } \sum_i \alpha_{i,t} = 1, \tag{7}$$

where $C \geq 0$ is a predefined scalar and $\{\alpha_{i,t}\}$ are the Lagrange multipliers for the dual of Eq. 7. As a result, letting $\{\alpha_{i,t}^*\}$ be the optimal multipliers, we then have the primal solution $\Delta\omega_t^*$ as

$$\Delta\omega_t^* = \sum_i \alpha_{i,t}^* \nabla f_i(\omega_t) = \sum_i \alpha_{i,t}^* \nabla\Big[d(x^+, x; \omega) - d(x^-, x; \omega)\Big]. \tag{8}$$

By comparing Eq. 1 with Eq. 8, we can see that if we set $\alpha_{x^-} = \alpha_{i,t}^*$, our one-class SVM solution will share the same formula and similar properties, and thus $\alpha_{i,t}^*$ can indicate the importance of the triplet for learning, to a certain degree. The key difference is that those losses have analytic solutions of the weights for contrastive gradients with higher computational efficiency and better usage of memory in large-scale scenarios, while our approach involves optimization as intermediate steps that potentially find better weights for gradient combinations with much fewer samples. In summary, our stochastic learning algorithm is shown in Alg. 1.

3.3 Analysis

Lemma 1 (Trust Region for $\Delta\omega_t^*$)**.** *For $\Delta\omega_t^*$ in Eq. 8, it holds that*

$$\|\Delta\omega_t^*\|_2 \leq \min\left\{\max_i \|\nabla f_i(\omega_t)\|_2, \left[\max_i f_i(\omega_t) + \min_j \left\{f_j(\omega_t) + \max_i \left\{\left|\nabla f_j(\omega_t)^T \nabla f_i(\omega_t)\right|\right\}\right\}\right]^{\frac{1}{2}}\right\}. \tag{9}$$

Proof. Based on Eq. 8, we can easily get

$$\|\Delta\omega_t^*\|_2 \leq \sum_i \alpha_{i,t}^* \|\nabla f_i(\omega_t)\|_2 \leq \max_i \|\nabla f_i(\omega_t)\|_2. \tag{10}$$

Letting ξ_i^*, ρ^* be the optimal solutions for the primal in Eq. 6 as well, then we have

$$\begin{aligned}
&\frac{1}{2}\|\Delta\omega_t^*\|_2^2 - \rho^* \leq \frac{1}{2}\|\Delta\omega_t^*\|_2^2 + C\sum_i \xi_i^* - \rho^* = \sum_i \alpha_{i,t}^* f_i(\omega_t) - \frac{1}{2}\|\Delta\omega_t^*\|_2^2 \\
\Longrightarrow &\|\Delta\omega_t^*\|_2^2 \leq \sum_i \alpha_{i,t}^* f_i(\omega_t) + \rho^* \leq \max_i f_i(\omega_t) + \rho^*.
\end{aligned} \tag{11}$$

Based on the property of a support vector indexed by j, it also holds that $\forall j$,

$$\begin{aligned}
\rho^* &= f_j(\omega_t) - \nabla f_j(\omega_t)^T \Delta\omega_t^* \\
&\leq f_j(\omega_t) + \Big|\sum_i \alpha_{i,t}^* \nabla f_j(\omega_t)^T \nabla f_i(\omega_t)\Big| \leq f_j(\omega_t) + \max_i \left\{\left|\nabla f_j(\omega_t)^T \nabla f_i(\omega_t)\right|\right\} \\
\Longrightarrow \rho^* &\leq \min_j \left\{f_j(\omega_t) + \max_i \left\{\left|\nabla f_j(\omega_t)^T \nabla f_i(\omega_t)\right|\right\}\right\}.
\end{aligned} \tag{12}$$

By substituting Eq. 12 into Eq. 11 and together with Eq. 10, we can complete our proof. □

Definition 1 (Neural Tangent Kernel in Contrastive Learning)**.** *We define a* neural tangent kernel *as*

$$\kappa_\omega\left((x_i, x_i^+, x_i^-), (x_j, x_j^+, x_j^-)\right) = \nabla f(x_i, x_i^+, x_i^-; \omega)^T \nabla f(x_j, x_j^+, x_j^-; \omega), \tag{13}$$

for all the triplets $(x_i, x_i^+, x_i^-), (x_j, x_j^+, x_j^-) \in \mathcal{X} \times \mathcal{X} \times \mathcal{X}$.

Lemma 2 (Contrastive Difference Approximates Kernel Aggregation)**.** *Suppose that (1) function f is Lipschitz continuous and smooth over ω for any triplet,* i.e., *both ∇f and $\nabla^2 f$ exist everywhere and are upper bounded, and (2) $\{\eta_t\}$ in Alg. 1 satisfies* $\lim_{t\to\infty}\eta_t = 0, \sum_{t=0}^{\infty}\eta_t = \infty, \sum_{t=0}^{\infty}\eta_t^2 < \infty$. *Then given a new triplet (y, y', y'') we have*

$$\lim_{T\to\infty} f(y, y', y''; \omega_T) \approx -\lim_{T\to\infty}\sum_{t=0}^{T-1}\eta_t \sum_i \alpha_{i,t}^* \kappa_{\omega_t}\left((x_i, x_i^+, x_i^-), (y, y', y'')\right). \tag{14}$$

Proof. We can recursively apply the second-order Taylor series to decompose f. Based on Lemma 1 and the assumptions on f we know that $\Delta\omega_t^*$ can be upper bounded. Then since

$$\begin{aligned}
f(y, y', y''; \omega_{t+1}) - f(y, y', y''; \omega_t) &= -\eta_t \nabla f(y, y', y''; \omega_t)^T \Delta\omega_t^* + \frac{\eta_t^2}{2}[\Delta\omega_t^*]^T \nabla^2 f(y, y', y''; \tilde{\omega}_t)\Delta\omega_t^* \\
&= -\eta_t \sum_i \alpha_{i,t}^* \kappa_{\omega_t}\left((x_i, x_i^+, x_i^-), (y, y', y'')\right) + O\left(\eta_t^2\right),
\end{aligned}$$

where $\tilde{\omega}_t$ is a linear interpolation between ω_t and ω_{t+1}, we can sum up over t on both sides so that

$$f(y, y', y''; \omega_T) - f(y, y', y''; \omega_0) = \sum_{t=0}^{T-1}\left[-\eta_t \sum_i \alpha_{i,t}^* \kappa_{\omega_t}\left((x_i, x_i^+, x_i^-), (y, y', y'')\right) + O\left(\eta_t^2\right)\right],$$

where both $f(y, y', y''; \omega_0)$ and $\sum_{t=0}^{\infty} O(\eta_t^2)$ are upper bounded. Now by taking a limit over T, we can complete our proof. □

4 EXPERIMENTS

To verify our analysis that contrastive learning approximates the ensemble of one-class SVMs, we follow the representation learning and linear probe protocol (Oord et al., 2018; He et al., 2016; Yeh et al., 2021). We conduct comprehensive experiments on CIFAR-10 (Krizhevsky et al., 2009) and STL-10 (Coates et al., 2011). Details are shown as follows.

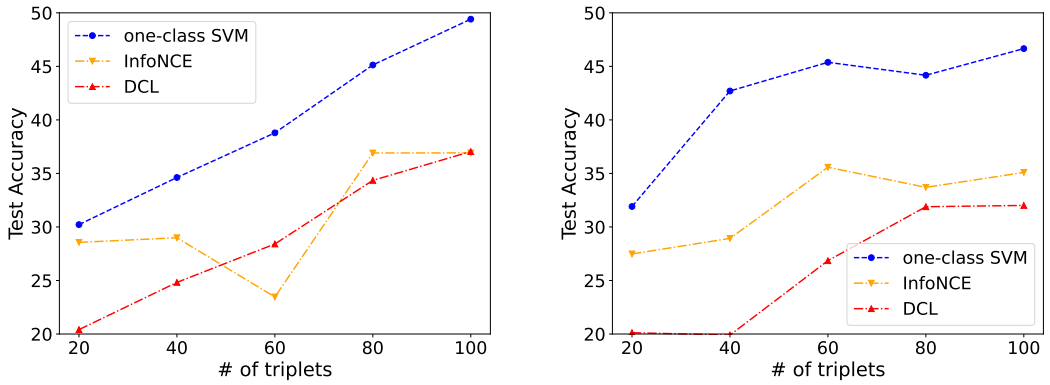

Figure 2: Comparison of our approach based on one-class SVMs and two contrastive losses on **(left)** CIFAR-10 and **(right)** STL-10 with the same linear classification protocol.

4.1 Datasets & Baseline Approaches

The CIFAR-10 dataset consists of 60,000 32×32 color images in 10 classes. There are 50,000 training images and 10,000 test images. STL-10 has 5,000 labeled training images in 10 classes and 100,000 unlabeled images. There are 800 test data for each class. Each instance has 96×96 pixels. We take the labeled part in our experiment without label leaking for the self-supervised pretraining. Since the purpose of this work is not to pursue the state-of-the-art performance on the widely used benchmark dataset but to demonstrate the theoretical analysis we made, we created a toy dataset CIFAR-10-toy by sampling 25% data from the original dataset for pretraining to mitigate the training overload in Alg. 1. The downstream linear evaluation is made on the original CIFAR-10 and STL-10. We compare our algorithm with InfoNCE loss (Oord et al., 2018) and decoupled contrastive learning (DCL) (Yeh et al., 2021) loss following SimCLR (Chen et al., 2020a) with ResNet-18 (He et al., 2016) as the backbone encoder. We refer to both methods as baselines in the following sections.

4.2 Implementation Details

During pretraining, we follow the commonly used instance discrimination pretext task (Wu et al., 2018; Ye et al., 2019; Bachman et al., 2019). We consider two views of the same image using the same data augmentation to form a positive pair. CIFAR-10-toy and STL-10 are randomly cropped to size 32×32 and 64×64, respectively. Then the random horizontal flip, color jittering, and random grayscale are taken following (Peng et al., 2022). The negative samples are other image views in the data pool. Usually, the negatives are all the other data from the same mini-batch. Due to the hardware limitation for extracting and storing per-sample gradients in Eq. 7, we design the experiment to reduce the number of data points in Eq. 7 by sampling triplets of (x, x^+, x^-) in every mini-batch so that we can finish our experiments within a reasonable time without any extra code optimization.

We sample 20, 40, 60, 80, 100 triplets randomly in every mini-batch from $128 \times 127 = 16,256$ in total to show the performance of our approach. For a fair comparison, the same number of triplets are used in InfoNCE and DCL. In the implementation of InfoNCE and DCL, we use the negative mask to sample (x, x^-) pairs first. Once the (x, x^-) is determined, the corresponding x^+ is also determined since it is just another view of x.

We train our algorithm and baseline methods representation backbones for 50 epochs with a batch size 64, SGD optimizer with a momentum of 0.9, and weight decay of 10^{-4}. we conduct our experiments on an Intel(R) Xeon(R) Silver 4214 CPU @ 2.20GHz and a single Nvidia Quadro RTX 6000 with 24GB memory. We apply CVXOPT (Vandenberghe, 2010) to solve Eq. 7, which runs on the CPU. We implement our algorithm and baseline methods based on the work of (Peng et al., 2022). Following the small-scale benchmark (Chen et al., 2020a; Yeh et al., 2021; Peng et al., 2022), we set the temperature τ to 0.07 for InfoNCE and DCL. We use a cosine-annealed learning rate of 0.5 for InfoNCE. For DCL learning rate, we use 0.0075 as suggested by (Yeh et al., 2021). For the learning rate in our algorithm, we utilize either a cosine-annealed learning rate starting from 0.5 or a fixed

Table 2: Accuracy comparison under the linear classification protocol on CIFAR-10.

pretrained	architecture	# triplets					
		20	40	60	80	100	16,256
InfoNCE	ResNet-18	28.56	28.99	23.46	36.91	36.92	57.75
DCL	ResNet-18	20.41	24.82	28.41	34.35	37.04	64.58
one-class SVM	ResNet-18	**30.22**	**34.62**	**38.79**	**45.13**	**49.41**	-

Table 3: Accuracy comparison under the linear classification protocol on STL-10.

pretrained	architecture	# triplets					
		20	40	60	80	100	16,256
InfoNCE	ResNet-18	27.48	28.93	35.58	33.70	35.09	50.65
DCL	ResNet-18	20.12	19.90	26.85	31.89	32.01	59.45
one-class SVM	ResNet-18	**31.91**	**42.70**	**45.38**	**44.17**	**46.66**	-

learning rate of 0.0075. The hyper-parameter C in Alg. 1 is also slightly tuned using cross-validation and finally set C values to 0.15 and 0.17 whose best performance is reported. We believe further tuning the C value would boost the one-class SVM algorithm performance.

To evaluate our approach and compare our results with baselines, we adopt the same setting as in (Peng et al., 2022) for training the linear classifier for all methods. The linear classifier is trained for 50 epochs with a learning rate of 10.0 for all the experiments with a batch size of 512 and SGD optimizer with a momentum of 0.9.

4.3 Results

We evaluate the performance of Alg. 1 as well as baseline methods with linear classification on frozen features following a common protocol in (He et al., 2020). After self-supervised pretraining, we freeze the network except for the last fully connected layer. We train the last layer classifier in a supervised way using the full dataset. We then report the top-1 classification accuracy on the test dataset. For the classifier, we do not search hyper-parameters, but keep exactly the same setup of the linear evaluation for all the experiments.

Our results on CIFAR-10 and STL-10 are shown in Table 2 and Table 3 and illustrated in Fig. 2. In the last column of the two tables, we also list the accuracy of baseline methods trained using all the triples without sampling. It is shown that on both datasets, reducing the number of triplets would reduce the top-1 accuracy in the linear probe greatly, compared with the result using the whole mini-batch. However, the one-class SVM can significantly outperform both InfoNCE and DCL under an extremely small number of triplets. The performance of all three methods is boosted with the increase of the number of triplets. The one-class SVM gains constantly with more samples, compared with the unstable performance boosting in InfoNCE and DCL when evaluating on CIFAR-10. The three methods all benefit from the increased number of samples in STL-10. Though the computational burden of the per-sample gradient in Eq. 7 prevents us from doing more experiments on a larger number of samples, the point is made clear that many one-class SVM updates could approximate the deep contrastive learning based on the behavior of our experiments. With 100 samples, the one-class SVM gets 49.41% accuracy compared to 36.92% and 37.04% counterparts in InfoNCE and DCL. Similarly, with only 60 samples, one-class SVM reaches 45.38% compared to 35.58% and 26.85% in the baseline methods. Although both InfoNCE and DCL outperform our method when using all the triplets, whose sizes are $163\times$ larger than ours on both datasets, we strongly believe that our performance could be further improved given such large number of triplets. This observation indicates that the number of triplets may be more important to achieve higher accuracy when the dual solutions are sufficiently good as an approximation. Such experimental results also support our analysis in Sec. 3 that solving the dual form in Eq. 7 and updating the network parameter sequentially is equivalent to the backpropagation of deep contrastive learning in terms of functionality.

By adjusting the parameter C in one-class SVMs, the classifier will learn the Lagrange multipliers, α's, that indicate the importance of each triplet in the mini-batches for constructing the decision boundaries. However, we find that it is very challenging to visualize the importance by showing the triples because in our experimental setting the positives and negatives come from many transformations such as cropping and jittering. Such operations are so random that we cannot even visually judge their importance or compare them with each other. Therefore, in this paper, we do not show any example of a triplet with higher or lower importance in a mini-batch. Instead, we show the $p(x^-)$ in InfoNCE and the Lagrange multipliers α for the same 127 triplets with the same x, x^+ in Fig. 3. The feature extraction network is pretrained with 60 samples in each mini-batch on STL-10. As shown in Fig. 3, the extremely high values of $p(x^-)$ and α co-occur quite frequently (see the peaks). The peak values around the 63rd triplet are 0.14 and 0.15. The co-occurences of high values in $p(x^-)$ and α also demonstrate that the triplets that decide the boundaries of SVMs are those that contribute most to the gradient update in deep contrastive learning.

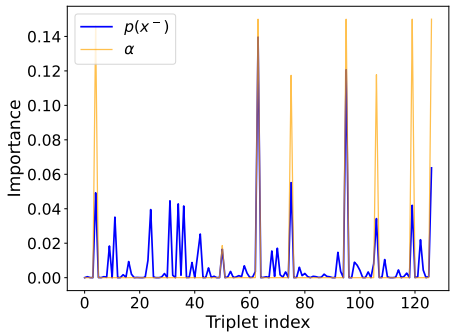

Figure 3: Comparison on triplet importance learned by InfoNCE, $p(x^-)$, and our one-class SVM, α.

To better understand the learning behavior of our method, we illustrate the training loss curves of the three methods in Fig. 4, where our loss curve is linearly scaled by 5 for a better view. In general, all the methods converge, which also supports our analysis of primal solution approximation in the one-class SVMs.

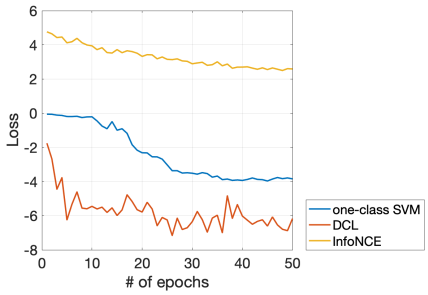

Figure 4: Loss comparison on STL-10.

5 Conclusion and Discussion

Compared with its prevailing application and impressive practical performance in recent years, contrastive learning is not fully understood from the theoretical perspective. Inspired by an experimental observation that when adding sufficient visually less hard negatives, contrastive learning could still learn comparable representations to those with visually hard negatives. This work attempts to better interpret the hardness of negative data in contrastive learning. Instead of considering positive and negative pairs in the literature, we provide a new insight to investigate the effect of triplet (x, x^+, x^-) in contrastive learning. We show theoretically that the gradients of a family of contrastive loss functions could be interpreted as approximate primal solutions for one-class SVMs with specific NTKs. In our analysis and empirical experiments, we demonstrate that the Lagrange multipliers of SVMs decide the importance of triplet in every learning epoch. The deep contrastive learning thus can be viewed approximately as an ensemble of one-class SVMs with NTKs.

The empirical bottlenecks in our method are related to the per-sample gradients, which prevent us from large-scale settings. This results in two limitations: (1) Computational bottleneck. Extracting per-sample gradients takes much longer than the conventional approaches. This problem can be mitigated by rewriting/optimizing the Cuda code. The quadratic programming solver for the dual in one-class SVMs is another barrier due to its complexity, especially when the number of triplets is large. However, since our kernel is linear, we can employ optimized SVM solvers such as Liblinear (Fan et al., 2008) for faster computation. (2) Storage bottleneck. Saving all the gradients needs large memory that is proportional to the model size as well as the number of triplets. To mitigate this problem, we could move the gradients from GPU to CPU memory.

Such limitations inspire us to think about how to better approximate the one-class SVM primal solution without explicitly extracting per-sample contrastive gradients. Similar to InfoNCE and DCL, one way will be embedding all the computations into a suitable loss function. In terms of applications, our current method may be very useful in few-shot learning with contrastive learning, due to the superior performance with small numbers of data samples.

REFERENCES

Relja Arandjelovic and Andrew Zisserman. Objects that sound. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 435–451, 2018.

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

Federico Augugliaro, Angela P Schoellig, and Raffaello D'Andrea. Generation of collision-free trajectories for a quadrocopter fleet: A sequential convex programming approach. In *2012 IEEE/RSJ international conference on Intelligent Robots and Systems*, pp. 1917–1922. IEEE, 2012.

Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. *Advances in neural information processing systems*, 32, 2019.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In *International conference on machine learning*, pp. 531–540. PMLR, 2018.

Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pp. 92–100, 1998.

Riccardo Bonalli, Abhishek Cauligi, Andrew Bylard, and Marco Pavone. Gusto: Guaranteed sequential trajectory optimization via sequential convex programming. In *2019 International conference on robotics and automation (ICRA)*, pp. 6741–6747. IEEE, 2019.

Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924, 2020.

Varun Chandola. Anomaly detection: A survey varun chandola, arindam banerjee, and vipin kumar, 2007.

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, 2020a.

Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33:22243–22255, 2020b.

Ting Chen, Calvin Luo, and Lala Li. Intriguing properties of contrastive losses. *Advances in Neural Information Processing Systems*, 34, 2021.

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020c.

Sumit Chopra, Raia Hadsell, and Yann 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. IEEE, 2005.

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.

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 215–223. JMLR Workshop and Conference Proceedings, 2011.

Juan M Coria, Hervé Bredin, Sahar Ghannay, and Sophie Rosset. A comparison of metric learning loss functions for end-to-end speaker verification. In *International Conference on Statistical Language and Speech Processing*, pp. 137–148. Springer, 2020.

Jia Deng. A large-scale hierarchical image database. *Proc. of IEEE Computer Vision and Pattern Recognition, 2009*, 2009.

MSC. John Duchi. Sequential Convex Programming . `https://web.stanford.edu/class/ee364b/lectures/seq_notes.pdf`, 2018. Accessed: 2022-09-16.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2): 303–338, 2010.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *Journal of machine learning research*, 9(Aug):1871–1874, 2008.

RCNN Faster. Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 9199(10.5555):2969239–2969250, 2015.

Songwei Ge, Shlok Mishra, Chun-Liang Li, Haohan Wang, and David Jacobs. Robust contrastive learning using negative samples with diminished semantics. *Advances in Neural Information Processing Systems*, 34, 2021.

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.

Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7297–7306, 2018.

Raia Hadsell, Sumit Chopra, and Yann 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. IEEE, 2006.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

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.

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

Qianjiang Hu, Xiao Wang, Wei Hu, and Guo-Jun Qi. Adco: Adversarial contrast for efficient learning of unsupervised representations from self-trained negative adversaries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1074–1083, 2021.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 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.

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

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.

Sungnyun Kim, Gihun Lee, Sangmin Bae, and Se-Young Yun. Mixco: Mix-up contrastive learning for visual representation. *arXiv preprint arXiv:2010.06300*, 2020.

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

Kibok Lee, Yian Zhu, Kihyuk Sohn, Chun-Liang Li, Jinwoo Shin, and Honglak Lee. i-mix: A domain-agnostic strategy for contrastive representation learning. *arXiv preprint arXiv:2010.08887*, 2020.

Chunyuan Li, Xiujun Li, Lei Zhang, Baolin Peng, Mingyuan Zhou, and Jianfeng Gao. Self-supervised pre-training with hard examples improves visual representations. *arXiv preprint arXiv:2012.13493*, 2020a.

Junnan Li, Pan Zhou, Caiming Xiong, and Steven CH Hoi. Prototypical contrastive learning of unsupervised representations. *arXiv preprint arXiv:2005.04966*, 2020b.

Zhaowen Li, Yousong Zhu, Fan Yang, Wei Li, Chaoyang Zhao, Yingying Chen, Zhiyang Chen, Jiahao Xie, Liwei Wu, Rui Zhao, et al. Univip: A unified framework for self-supervised visual pre-training. *arXiv preprint arXiv:2203.06965*, 2022.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Lajanugen Logeswaran and Honglak Lee. An efficient framework for learning sentence representations. *arXiv preprint arXiv:1803.02893*, 2018.

Florian Messerer, Katrin Baumgärtner, and Moritz Diehl. Survey of sequential convex programming and generalized gauss-newton methods. *ESAIM. Proceedings and Surveys*, 71:64, 2021.

Shlok Mishra, Anshul Shah, Ankan Bansal, Abhyuday Jagannatha, Abhishek Sharma, David Jacobs, and Dilip Krishnan. Object-aware cropping for self-supervised learning. *arXiv preprint arXiv:2112.00319*, 2021.

Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6707–6717, 2020.

Daniel Morgan, Soon-Jo Chung, and Fred Y Hadaegh. Model predictive control of swarms of spacecraft using sequential convex programming. *Journal of Guidance, Control, and Dynamics*, 37(6):1725–1740, 2014.

Mary M Moya and Don R Hush. Network constraints and multi-objective optimization for one-class classification. *Neural networks*, 9(3):463–474, 1996.

Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4004–4012, 2016.

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

Xiangyu Peng, Kai Wang, Zheng Zhu, Mang Wang, and Yang You. Crafting better contrastive views for siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16031–16040, 2022.

Marco AF Pimentel, David A Clifton, Lei Clifton, and Lionel Tarassenko. A review of novelty detection. *Signal processing*, 99:215–249, 2014.

Michael JD Powell. Direct search algorithms for optimization calculations. *Acta numerica*, 7: 287–336, 1998.

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.

Gunnar Ratsch, Sebastian Mika, Bernhard Scholkopf, and K-R Muller. Constructing boosting algorithms from svms: An application to one-class classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(9):1184–1199, 2002.

Sucheng Ren, Huiyu Wang, Zhengqi Gao, Shengfeng He, Alan Yuille, Yuyin Zhou, and Cihang Xie. A simple data mixing prior for improving self-supervised learning. In *CVPR*, 2022.

Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. *arXiv preprint arXiv:2010.04592*, 2020.

Joshua David Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. In *International Conference on Learning Representations*, 2021.

Stephan R Sain. The nature of statistical learning theory, 1996.

Bernhard Schölkopf, Robert C Williamson, Alex Smola, John Shawe-Taylor, and John Platt. Support vector method for novelty detection. *Advances in neural information processing systems*, 12, 1999.

Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 815–823, 2015.

Ramprasaath R Selvaraju, Karan Desai, Justin Johnson, and Nikhil Naik. Casting your model: Learning to localize improves self-supervised representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11058–11067, 2021.

Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, Sergey Levine, and Google Brain. Time-contrastive networks: Self-supervised learning from video. In *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 1134–1141. IEEE, 2018.

Zhiqiang Shen, Zechun Liu, Zhuang Liu, Marios Savvides, Trevor Darrell, and Eric Xing. Unmix: Rethinking image mixtures for unsupervised visual representation learning. *arXiv preprint arXiv:2003.05438*, 2020.

Abhishek Sinha, Kumar Ayush, Jiaming Song, Burak Uzkent, Hongxia Jin, and Stefano Ermon. Negative data augmentation. In *International Conference on Learning Representations*, 2021.

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

Aravind Srinivas, Michael Laskin, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. *arXiv preprint arXiv:2004.04136*, 2020.

David Martinus Johannes Tax. One-class classification: Concept learning in the absence of counter-examples. 2002.

David MJ Tax and Robert PW Duin. Support vector domain description. *Pattern recognition letters*, 20(11-13):1191–1199, 1999.

David MJ Tax and Robert PW Duin. Support vector data description. *Machine learning*, 54(1):45–66, 2004.

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

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? *Advances in Neural Information Processing Systems*, 33:6827–6839, 2020b.

Yuandong Tian. Deep contrastive learning is provably (almost) principal component analysis. *arXiv preprint arXiv:2201.12680*, 2022.

Yuandong Tian, Lantao Yu, Xinlei Chen, and Surya Ganguli. Understanding self-supervised learning with dual deep networks. *arXiv preprint arXiv:2010.00578*, 2020c.

Michael Tschannen, Josip Djolonga, Marvin Ritter, Aravindh Mahendran, Neil Houlsby, Sylvain Gelly, and Mario Lucic. Self-supervised learning of video-induced visual invariances. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13806–13815, 2020.

Lieven Vandenberghe. The cvxopt linear and quadratic cone program solvers. *Online: http://cvxopt. org/documentation/coneprog. pdf*, 2010.

Vikas Verma, Thang Luong, Kenji Kawaguchi, Hieu Pham, and Quoc Le. Towards domain-agnostic contrastive learning. In *International Conference on Machine Learning*, pp. 10530–10541. PMLR, 2021.

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

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pp. 9929–9939. PMLR, 2020.

Zhenbo Wang and Michael J Grant. Constrained trajectory optimization for planetary entry via sequential convex programming. *Journal of Guidance, Control, and Dynamics*, 40(10):2603–2615, 2017.

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

Chang Xu, Dacheng Tao, and Chao Xu. A survey on multi-view learning. *arXiv preprint arXiv:1304.5634*, 2013.

Mang Ye, Xu Zhang, Pong C Yuen, and Shih-Fu Chang. Unsupervised embedding learning via invariant and spreading instance feature. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6210–6219, 2019.

Chun-Hsiao Yeh, Cheng-Yao Hong, Yen-Chi Hsu, Tyng-Luh Liu, Yubei Chen, and Yann LeCun. Decoupled contrastive learning. *arXiv preprint arXiv:2110.06848*, 2021.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.