Similarity-Agnostic Contrastive Learning With Alterable Self-Supervision

Shuo Chen, Chen Gong, *Senior Member, IEEE*, Jun Li, *Member, IEEE*, and Jian Yang, *Member, IEEE*

***Abstract*—Self-supervised contrastive learning (CL) seeks to learn generalizable feature representations via the self-supervision of *pairwise similarities*, where existing CL approaches usually build definite similarity labels (e.g., positive or negative) for model training. Yet in practice, the same pair of instances may have opposite similarity labels in different scenarios, e.g., two interclass images from CIFAR-100 can be a similar pair in CIFAR-20. Learning with definite similarities can hardly obtain an ideal representation that simultaneously characterizes the similar and dissimilar patterns (e.g., the contexts and details) between each two instances. Therefore, pairwise similarities used for CL should be *agnostic*, and we argue that *simultaneously considering* both the similarity and dissimilarity for each data pair could learn more generalizable representations. To this end, we propose similarity-agnostic CL (SACL), which generalizes the *instance discrimination* strategy of conventional CL to a new multiobjective programming (MOP) form. In SACL, we build multiple projection layers with corresponding regularizers to constrain the distance matrix to have *different sparsity in different objectives* so that we can obtain *alterable pairwise distances* to capture both the similarity and dissimilarity between each pair of instances. We show that SACL can be equivalently converted to a single learning objective, easily solved by stochastic optimization with convergence guarantees. Theoretically, we prove a *tighter* error bound than conventional CL approaches; empirically, our method improves the downstream task performance for image, text, and graph data.**

***Index Terms*—Agnostic similarity, contrastive learning (CL), model generalizability, self-supervised learning, similarity learning.**

I. Introduction

LEARNING representations without human annotations is a long-standing problem and has great significance in a lot of practical uses [4], [15], [26], [78]. Recently, contrastive learning (CL) successfully promotes the *unsupervised* representation learning and shows encouraging performance when compared with *fully supervised* learning approaches [32], [61]. As CL directly pretrains a generic feature representation by autonomously building the pseudo-supervision (i.e., the *self-supervision*) from the raw data, the learned representation can be applied to various downstream tasks such as classification [33], retrieval [38], and clustering [88].

Originally, the contrast information of CL was generated by regarding each pair of instances as a *negative pair* to conduct the *instance discrimination* [19], [73] so that the representations of each pair of instances can be pushed away. After that, some representative works such as *SimCLR* [11] and *MoCo* [27] introduced the *data augmentation* to further construct *positive pairs*, and empirical results also successfully demonstrated the great effectiveness of CL, especially on image and graph data [3], [63], [78] even compared with some fully supervised learning methods. Currently, the similarity-based loss functions that fully integrate negative pairs and positive pairs have already become a commonly used setting for a lot of self-supervised CL algorithms in different domains [67], [69], [84].

Accordingly, the most recent advances in CL mainly focus on two aspects: positive pair enrichment and negative pair correction. On the one hand, several perturbation techniques were introduced into CL, further enriching critical intraclass information for model training. For example, the multimodal coding [47], [49], [57] and adversarial generation [31], [43], [64] were applied to CL to build more plentiful augmentation results. Meanwhile, to avoid the improper positive pairs generated by excessive data augmentations, some filter mechanisms (e.g., the distribution divergence constraint [67] and sharpening distribution strategy [87]) were proposed to better control the generation of augmented/perturbed instances. On the other hand, as the straightforward instance discrimination may result in some false negatives (i.e., the semantically similar instances yet being pushed away), recent works adopted traditional approaches such as pseudo-labeling [85], positive-unlabeled learning [16], and metric learning [10], [17], to effectively reduce the impact of those false negatives. Moreover, considering that diverse negative pairs may have different influences on the learned representation, the hard-mining and importance weighting techniques [35], [51] were also deployed to further adjust the occurrence frequencies of negative pairs in a mini-batch optimization. In summary, existing CL approaches have been greatly promoted by the utilization of positive and negative pairs.

Although existing CL approaches have achieved very promising results in a lot of downstream tasks, most of them can usually consider either the *similar* or *dissimilar pattern*

Received 9 January 2025; revised 25 April 2025; accepted 8 May 2025. This work was supported by the National Science Fund of China under Grant U24A20330, Grant 62361166670, Grant 62336003, and Grant 12371510. *(Corresponding author: Shuo Chen.)*

Shuo Chen is with the School of Intelligence Science and Technology, Nanjing University, Nanjing 210093, China (e-mail: shuo.chen@nju.edu.cn).

Chen Gong, Jun Li, and Jian Yang are with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: chen.gong@njust.edu.cn; junli@njust.edu.cn; csjyang@njust.edu.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/TNNLS.2025.3570784, provided by the authors.

Digital Object Identifier 10.1109/TNNLS.2025.3570784

TABLE I

COMPARISON BETWEEN OUR PROPOSED METHOD AND VARIOUS EXISTING CL APPROACHES, WHERE DA AND ID STAND FOR DATA AUGMENTATION AND INSTANCE DISCRIMINATION, RESPECTIVELY

Methodology	Reducing False Similarity	Softening Similarity	Hierarchical Similarity	Agnostic Similarity	Model&Loss Independent	Main Philosophy
Original Methods [11], [73]	✗	✗	✗	✗	✓	DA and ID
Negative-Free Methods [13], [24]	✓	✗	✗	✗	✗	Discarding ID
Uncertainty/Smoothing Methods [65], [72]	✗	✓	✗	✗	✓	Similarity is continuous
Clustering/Prototype based Methods [7], [41]	✓	✗	✓	✗	✗	Having hierarchical clusters
Debiased/Denoised Methods [10], [16], [51]	✓	✗	✗	✗	✓	Correcting false ID
Similarity-Agnostic Contrastive Learning (ours)	✓	✓	✓	✓	✓	Similarity is agnostic

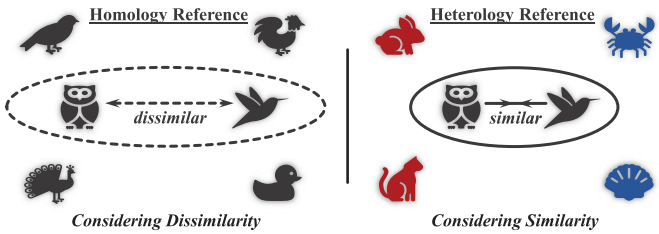


Fig. 1. Conceptual illustration for *similarity-agnostic* contrast. Due to the local context of data, the fineness degree of categorization, or any other reasons, the similarity between the same pair of instances can be opposite in different cases. We propose to characterize such an important property so that we can *enrich the self-supervisory information* for obtaining a better representation with stronger generalizability.

(e.g., the contexts and details) between a pair of instances, and they ignore the remaining pairwise relationship within such a data pair, which is also potentially useful (as shown in Fig. 1). The traditional clustering and prototype algorithms [7], [41], [89] were introduced into CL, successfully leveraging the hierarchical class structures of data to consider different pairwise similarities (i.e., multiple similarities) in different granularity scenarios. However, the clustering results usually depend on the reliability of the learned representations, and these approaches inevitably become weak or ineffective when the original data do not really have hierarchical structures. Some recent works further employed the uncertainty-aware [2], [65] and label-smoothing [72], [83] techniques to soften pairwise similarities to be continuous values, which allow us to better characterize the similarity degree for each data pair. Nevertheless, their softened similarities are usually *definite values*, and thus, each data pair still only receives the supervision signal composed of a single scalar. Therefore, the feature representations learned by most existing CL approaches can hardly capture the similar and dissimilar patterns simultaneously (which are both useful) between each pair of instances, so the generalizability of these representations would be limited.

To overcome the drawbacks of the existing methods mentioned above, in this article, we propose a novel method called similarity-agnostic CL (SACL) to construct alterable self-supervision, simultaneously exploring both the similarity and dissimilarity within each single data pair. Here, "agnostic" means that the learning algorithms and encoder models are agnostic to the intrinsic similarities between instances, and we call the changeable pairwise similarities captured in different learning tasks as "alterable" similarities. Our basic motivation is that the pairwise similarities used for CL are unknowable and changeable, so the learning algorithm should be agnostic to the similarity supervision, and we need a new hierarchy-free approach to utilize the potential multiple similarities of each data pair. This inspires us to generalize the instance discrimination of conventional CL to a multiobjective programming (MOP) form. The single learning objective of conventional CL enlarges the distance between each pair of instances, and the distance matrix of pairwise instances is usually *nonsparse*. We extend such a real-valued loss to a vector-valued loss to consider the changeable similarities between instances, where the pairwise similarities are alterable within several sparse distance matrices. Specifically, we constrain a series of distance matrices to have *different sparsity in different projection layers*, so we can obtain alterable distances to supervise the learning of both the similarity and dissimilarity for each data pair. In this way, the generalizability of the learned representation can be successfully improved. We prove that the proposed SACL (i.e., the MOP problem with a vector-valued loss) can be equivalently converted to a regular real-valued loss minimization. We design the stochastic algorithm to optimize the learning objective and also provide the corresponding theoretical analysis to guarantee the effectiveness and soundness of our method. The experiments on image, text, and graph benchmarks clearly demonstrate the superiority of our method when compared with existing CL approaches.

Our method is generic and effective, and it can be easily deployed in many existing CL approaches (marked as Model&Loss Independent in Table I), further improving the generalization ability of the learned representations with negligible additional computation burdens. Table I lists the detailed comparison of our method with existing CL approaches, and our main contributions are summarized as follows.

1) We provide a new viewpoint that pairwise similarities used for CL are agnostic, which suggests that both the

similarity and dissimilarity of each data pair are important for learning more generalizable representations.

2) We propose a new method with theoretical guarantees to generalize the conventional CL to a similarity-agnostic form, simultaneously exploring the similarity and dissimilarity between pairwise instances.
3) We conduct extensive experiments on real-world data to validate the effectiveness of our method, and results on multiple domain tasks consistently demonstrate the superiority of our method to the state-of-the-art CL approaches.

The rest of this article starts with a brief review of the background in Section II. Then, Section III details the SACL framework and the corresponding formulation. Section IV provides theoretical analyses on the optimization property, distance sparsity, and model generalizability of our method. Section V shows experimental results on real-world benchmark datasets. Finally, Section VI concludes this article.

II. Background and Related Work

In this section, we first introduce some necessary notations. Then, we briefly review the background of (supervised) metric learning and (self-supervised) CL.

Notations

Throughout this article, we write matrices, vectors, and three-order tensors as bold uppercase characters, bold lowercase characters, and bold calligraphic uppercase characters, respectively. We denote the training data $\mathscr{X} = \{\boldsymbol{x}_i \in \mathbb{R}^m | i = 1, 2, \ldots, N\}$, where m is the data dimensionality and N is the sample size. Here, operators $\|\cdot\|_1$ and $\|\cdot\|_2$ denote the ℓ_1-norm and ℓ_2-norm, respectively. $\boldsymbol{\mathcal{P}}_{::k}$ denotes the kth slice of the tensor $\boldsymbol{\mathcal{P}}$.

A. Supervised Metric Learning

Pairwise similarities of training data can be directly annotated by humans and generated by the class labels, e.g., the verification task related data [56] and the classification task related data [74], respectively. Based on the pairwise similarities, people can design some pair-based loss functions to learn distance metrics and the corresponding feature representations for several downstream recognition tasks. Such a fully supervised problem setting is usually called *metric learning* or *similarity learning* [6], [79], [80], [82].

In metric learning, the central problem is how to learn a distance metric or a feature representation that faithfully reflects the pairwise similarity between each pair of instances from a dataset. In the past decades, both linear and nonlinear metric learning approaches have been proposed to learn a generic feature representation $\boldsymbol{\varphi} : \mathbb{R}^m \to \mathbb{R}^h$ (where h is the feature dimensionality), and the corresponding learnable distance can be written as $d_{\boldsymbol{\varphi}}(\boldsymbol{x}, \widehat{\boldsymbol{x}}) = \|\boldsymbol{\varphi}(\boldsymbol{x})/\|\boldsymbol{\varphi}(\boldsymbol{x})\|_2 - \boldsymbol{\varphi}(\widehat{\boldsymbol{x}})/\|\boldsymbol{\varphi}(\widehat{\boldsymbol{x}})\|_2\|_2$, which measures the similarity between instances $\boldsymbol{x}$ and $\widehat{\boldsymbol{x}}$ sampled from the m-dimensional space. Since human annotations are available, the basic objective of metric learning is to enlarge the distance value $d_{\boldsymbol{\varphi}}(\boldsymbol{x}, \widehat{\boldsymbol{x}})$ if $\boldsymbol{x}$ and $\widehat{\boldsymbol{x}}$ are similar and meanwhile reduce $d_{\boldsymbol{\varphi}}(\boldsymbol{x}, \widehat{\boldsymbol{x}})$ if $\boldsymbol{x}$ and $\widehat{\boldsymbol{x}}$ are dissimilar [76]. For example, suppose that we are given $n+1$ data pairs $\{(\boldsymbol{x}_i, \boldsymbol{x}_k), (\boldsymbol{x}_i, \boldsymbol{x}_{b_1}), (\boldsymbol{x}_i, \boldsymbol{x}_{b_2}), \ldots, (\boldsymbol{x}_i, \boldsymbol{x}_{b_n})\}$ that uniformly sampled from the training data $\mathscr{X}$, and the widely used $(n+1)$-*tuplet loss* [55] can be written as

$$\mathcal{L}_{\mathrm{TUP}}(\boldsymbol{\varphi}) = \mathbb{E}\left[-\log \frac{\mathrm{e}^{-d_{\boldsymbol{\varphi}}(\boldsymbol{x}_i, \boldsymbol{x}_k)}}{\mathrm{e}^{-d_{\boldsymbol{\varphi}}(\boldsymbol{x}_i, \boldsymbol{x}_k)} + \sum_{j=1}^{n} \mathrm{e}^{-d_{\boldsymbol{\varphi}}\left(\boldsymbol{x}_i, \boldsymbol{x}_{b_j}\right)}}\right] \tag{1}$$

where $(\boldsymbol{x}_i, \boldsymbol{x}_k)$ is a similar pair and $(\boldsymbol{x}_i, \boldsymbol{x}_{b_j})$ is a dissimilar pair for $i, k = 1, 2, \ldots, N$ and $j = 1, 2, \ldots, n$. Besides the above well-known projected Euclidean distance $d_{\boldsymbol{\varphi}}(\boldsymbol{x}, \widehat{\boldsymbol{x}})$, some new similarity metrics such as the asymmetric metric [80] and relation alignment metric [90] were also proposed to enrich the similarity relationship among instances. Nevertheless, most existing metric learning approaches assume that we are given definite similarity annotations, so we can only consider either the similarity or dissimilarity between a given pair of instances from training data. To address this issue, recent works proposed to use more plentiful data annotations such as hierarchical similarity [21], [81], multihead similarity [59], and multilabel information [22], [75] such that learning more generalizable feature representations and the corresponding similarity metrics.

B. Self-Supervised CL

As a popular self-supervised representation learning approach, CL shares a very analogous training manner (i.e., considering the pairwise relationship) with metric learning. Since human annotations are not available anymore, CL usually builds pseudo-supervision, namely self-supervision.

Existing CL methods usually have two critical components: the *instance discrimination* for generating negative pairs [19], [73] and the *data augmentation* for generating positive pairs [11], [31]. Based on this common setting, we suppose that $n+1$ data pairs $\{(\boldsymbol{x}, \boldsymbol{x}^+), (\boldsymbol{x}, \boldsymbol{x}_{b_1}), (\boldsymbol{x}, \boldsymbol{x}_{b_2}), \ldots, (\boldsymbol{x}, \boldsymbol{x}_{b_n})\}$ are uniformly sampled from the training data $\mathscr{X}$, where $(\boldsymbol{x}, \boldsymbol{x}^+)$ is a positive pair and $(\boldsymbol{x}, \boldsymbol{x}_{b_j})$ is a negative pair for $j = 1, 2, \ldots, n$. Here, $\boldsymbol{x}^+$ is a random data augmentation of the instance $\boldsymbol{x}$ from the training data. Then, we can formulate the learning objective of CL as the following noise contrastive estimation (NCE) loss [73]:

$$\mathcal{L}_{\mathrm{NCE}}(\boldsymbol{\varphi}) = \mathbb{E}\left[-\log \frac{\mathrm{e}^{-d_{\boldsymbol{\varphi}}(\boldsymbol{x}, \boldsymbol{x}^+)/\gamma}}{\mathrm{e}^{-d_{\boldsymbol{\varphi}}(\boldsymbol{x}, \boldsymbol{x}^+)/\gamma} + \sum_{j=1}^{n} \mathrm{e}^{-d_{\boldsymbol{\varphi}}\left(\boldsymbol{x}, \boldsymbol{x}_{b_j}\right)/\gamma}}\right] \tag{2}$$

which still seeks to learn a generic feature representation $\boldsymbol{\varphi} : \mathbb{R}^m \to \mathbb{R}^h$ in h-dimensional space, and $\gamma > 0$ is an additional temperature parameter. In the above learning objective, it is obvious that $\boldsymbol{x}$ and $\boldsymbol{x}_{b_j}$ $(j = 1, 2, \ldots, n)$ might be semantically similar, whereas they are undeservedly pushed away from each other. Therefore, some recent works proposed to correct/reduce the false negative pairs by various conventional techniques such as positive-unlabeled learning [16], pseudo-labeling [85], and regularization approaches [10], [70]. Meanwhile, the classical prototype and clustering algorithms were adopted to consider the hierarchical class information for some cases with multiple granularities. Nevertheless, the effectiveness

of the clustering/prototype usually depends on the learned representation itself, and these methods inevitably become weak when the training data do not really have hierarchical structures. Furthermore, the uncertainty-aware [2] and label-smoothing [86] techniques were also introduced into CL to characterize pairwise similarities more precisely. However, the softened similarities are usually *definite values*, and thus, each data pair still only receives the supervision signal composed of a single scalar. It means that representations learned by most existing CL approaches can hardly capture the similar and dissimilar patterns concurrently (which are both useful) between each pair of instances. Therefore, we aim to propose a new method to simultaneously consider both the similarity and dissimilarity of each single data pair, further boosting the generalization ability of learned representations.

III. Methodology

In this section, we first discuss the necessity of considering agnostic similarities in self-supervised CL. After that, we propose a novel framework dubbed SACL by introducing multiple projection layers with the corresponding sparse regularization. The learning objective and the corresponding optimization algorithm are finally designed with convergence guarantees.

A. Motivation

In both supervised metric learning and self-supervised contrastive learning, the pairwise similarity plays a critical role in the model design. Most settings for both learning tasks assume that the real pairwise similarity between each two instances is clearly defined. That is to say, once two instances are given, their similarity will usually be specified. Nevertheless, in many real-world scenarios, the same pair of instances may have diverse similarities in different cases due to the local context, the target of the learning task, or any other reasons.

To be specific, here we take the recognition tasks on the CIFAR [39] and In-shop [44] datasets as examples. We adopt the $(n+1)$-tuplet loss as we discussed in (1) to train *ResNet50*-based [29] encoders and learn the similarity metrics supervised by different annotations. In Fig. 2(a), we record the distance values between intraclass instances in CIFAR-100, as well as the corresponding distance distribution of those same instances in CIFAR-20. For the In-shop dataset, we directly train two models based on the clothes annotation and pose annotation and visualize the distance values for some representative data pairs, which is denoted as "ClothesA-PoseU and ClothesB-PoseV" in Fig. 2(b). Note that in this figure, we only list a fraction of instance pairs from the training data for clarity, and the global average distances of all data pairs are plotted with colored dashed lines. In the above empirical results, the similarity between the same pair of instances *significantly changes* in different learning scenarios, where the values of red bars are *opposite* to the values of blue bars, and the distance ranks in different annotation cases are completely different from each other. This is actually a common issue in a lot of real-world recognition tasks, and thus, the pairwise similarity has to be agnostic if we want to learn a generalizable similarity metric and feature representation.

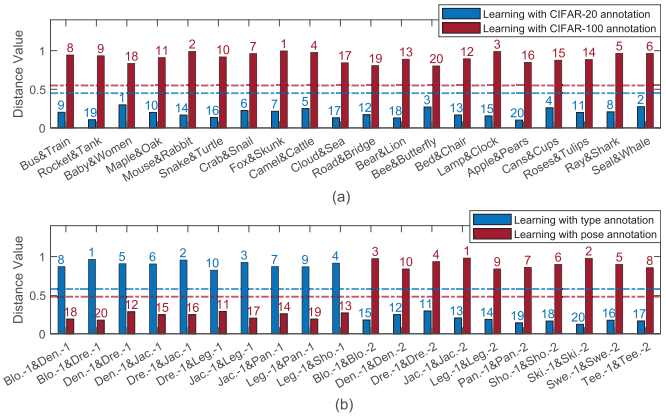


Fig. 2. Distributions of pairwise distances learned with different annotations (i.e., the same images divided into 20 and 100 classes in CIFAR-20 and CIFAR-100, respectively) and different learning targets (i.e., the clothes classification and pose classification for In-shop dataset). Here, the colored numbers on the bars indicate the distance ranks, and the dashed lines are the global average distances of all data pairs (including those distances not plotted in the figure). (a) Distance distribution of learned pairwise similarity on CIFAR dataset. (b) Distance distribution of learned pairwise similarity on In-shop dataset.

1) Similarity-Agnosticity Is a Trouble in Supervised Learning: In the supervised case, pairwise similarities are directly annotated by humans. Most learning algorithms aim to fit human annotations as much as possible based on some inductive biases. Yet, the human annotation for a single data pair might be changeable in different cases. As we illustrated in Fig. 2, if there are two images from "bus" and "train" in CIFAR-100, they should be regarded as a negative pair (dissimilar) because they come from two different classes. However, such two images are treated as a positive pair (similar) in CIFAR-20, which shares the completely same training instances with CIFAR-100, because the two images will belong to the same super-class "vehicles-1." There exist agnostic similarities in supervised learning, but this may lead to some troubles. It means that we need more complex annotations such as multiview, multilabel, and hierarchical information [22], [77], [81] to consider each possibility of the agnostic similarity. This would significantly increase the human cost, even if we could provide the accurate supervision.

2) Similarity-Agnosticity is a Blessing in Self-Supervised Learning: Interestingly, the above issue caused by agnostic similarities in supervised learning (i.e., metric learning) can be naturally solved in the self-supervised scenario (i.e., CL). Existing CL approaches construct the pseudo-supervision by combining each pair of instances as a negative pair. Such pseudo-supervision can be regarded as a manner of capturing low-level (e.g., fine-grained) similarity, and at this point, there will be no *false negatives* anymore. This further inspires us to naturally generalize the current instance discrimination to an MOP form, and each objective has its own independent distance supervision to learn a shared feature representation. As the distance between the same pair of instances may change with different objectives, such alterable supervision can adaptively capture both similarity and dissimilarity within each data pair. In this case, the agnostic similarity has actually

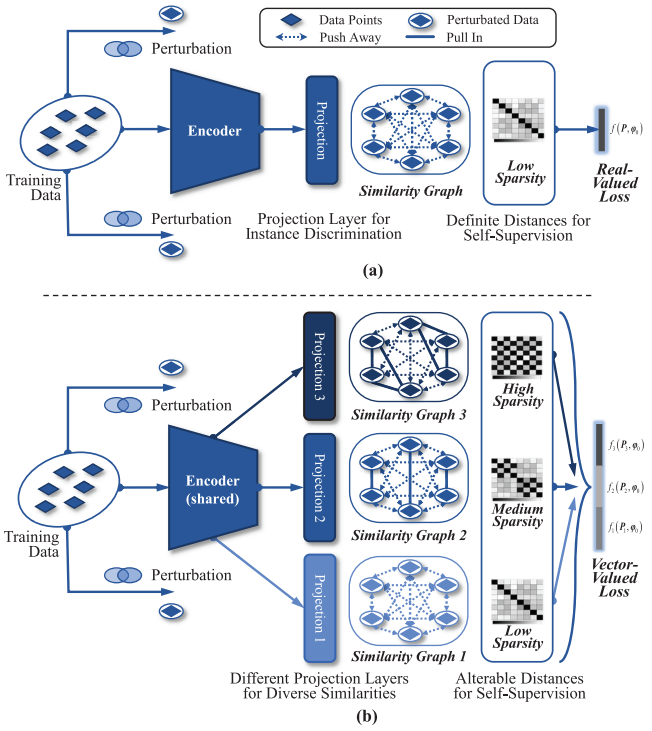


Fig. 3. Framework comparison between (a) conventional CL and (b) our proposed SACL, where we take three projection layers for illustration. Our SACL is a natural generalization of the conventional method. We regard the instance discrimination (which is the basic component of existing CL) as capturing low-level similarity, and we extend it to a multiobjective form to build alterable distances for more plentiful supervision.

become a good thing because it allows us to further consider the remaining (similar or dissimilar) pattern among instances, which is useful but ignored in conventional CL methods.

B. Formulation

Now, we discuss the technical details of our method. CL seeks to learn a generic network mapping $\boldsymbol{\varphi}_0 : \mathbb{R}^m \to \mathbb{R}^h$ [i.e., $\boldsymbol{\varphi}$ in (2)], which can be decomposed to $\boldsymbol{\varphi}_0 = \boldsymbol{P}_0 \cdot \boldsymbol{\Phi}$, where $\boldsymbol{\Phi} : \mathbb{R}^m \to \mathbb{R}^{\mathrm{H}}$ is the feature encoder and $\boldsymbol{P}_0 : \mathbb{R}^{\mathrm{H}} \to \mathbb{R}^h$ is a projection head as shown in Fig. 3(a). In previous works, $\boldsymbol{\Phi}(\cdot)$ is usually employed as the final features for downstream tasks, and the projection head $\boldsymbol{P}_0$ is used to perform the instance discrimination, pushing instances away from each other.

1) Distance Tensor: As the instance discrimination can be regarded as capturing the low-level similarity (i.e., the similarity calculated with the fine-grained features), we consider introducing multiple projection layers to enumerate the other potential similarities so that we can further enrich the self-supervisory information for model training. First, here we suppose that there are C projection layers with the corresponding parameters $\boldsymbol{\mathcal{P}}_{::1}, \boldsymbol{\mathcal{P}}_{::2}, \ldots, \boldsymbol{\mathcal{P}}_{::C} \in \mathbb{R}^{h \times h}$, where $\boldsymbol{\mathcal{P}}_{::k}$ is the kth slice of the tensor $\boldsymbol{\mathcal{P}}$ for $k = 1, 2, \ldots, C$. As shown in Fig. 3(b), the kth projection layer is the multiplicative result $\boldsymbol{\varphi}_k = \boldsymbol{\mathcal{P}}_{::k} \cdot \boldsymbol{\varphi}_0$. Then, for such C projection layers, we define that the distance tensor $\boldsymbol{\mathcal{D}}(\{\boldsymbol{\varphi}_k\}_{k=1}^C) = [\mathcal{D}_{ijk}] \in \mathbb{R}^{N \times N \times C}$, where $\mathcal{D}_{ijk}$ denotes the distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ in the kth layer, i.e.,

$$\mathcal{D}_{ijk} = d_{\boldsymbol{\varphi}_k}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left\| \frac{\boldsymbol{\varphi}_k(\boldsymbol{x}_i)}{\|\boldsymbol{\varphi}_k(\boldsymbol{x}_i)\|_2} - \frac{\boldsymbol{\varphi}_k(\boldsymbol{x}_j)}{\|\boldsymbol{\varphi}_k(\boldsymbol{x}_j)\|_2} \right\|_2 \tag{3}$$

for $i, j = 1, 2, \ldots, N$ and $k = 1, 2, \ldots, C$. After that, we would like to further investigate the value distribution of the distance tensor $\boldsymbol{\mathcal{D}}(\{\boldsymbol{\varphi}_k\}_{k=1}^C)$ in different projection layers.

2) Bottom-Up Sparsity: As the original projection $\boldsymbol{\varphi}_0$ mainly focuses on the instance-wise divergence, all data points should be pushed away from each other (i.e., the original instance discrimination). We use bottom projection layers to maintain such an objective to capture the variability between instances, and thus, most elements in matrix $\boldsymbol{\mathcal{D}}_{::k} \in \mathbb{R}^{N \times N}$ should be large values for a small k, which implies that $\boldsymbol{\mathcal{D}}_{::k}$ is *nonsparse* in the bottom layers. Meanwhile, the top projection layers focus on high-level divergences (calculated with the coarse-grained features), and thus, most pairwise distances should have small values to capture the commonality between instances. Therefore, the matrix $\boldsymbol{\mathcal{D}}_{::k}$ is *sparse* for a large k (e.g., $k = C$). In general, all slice matrices in the distance tensor $\boldsymbol{\mathcal{D}}(\{\boldsymbol{\varphi}_k\}_{k=1}^C)$ satisfy that $\|\boldsymbol{\mathcal{D}}_{::1}\|_1 > \|\boldsymbol{\mathcal{D}}_{::2}\|_1 > \cdots > \|\boldsymbol{\mathcal{D}}_{::C}\|_1$, although there is no strict hierarchy in SACL. Here, we further introduce a tolerance parameter $\tau > 0$ to allow the small bias, and then, we have that for any $1 \le u < v \le C$

$$\|\max(\boldsymbol{\mathcal{D}}_{::u} - \tau, 0)\|_1 > \|\max(\boldsymbol{\mathcal{D}}_{::v} - \tau, 0)\|_1 . \tag{4}$$

It is worth pointing out that the abovementioned small values are successfully considered as zero values by the operation $\max(\boldsymbol{\mathcal{D}}_{::k} - \tau, 0)$ (which ignores any elements that are smaller than τ) so that the calculated matrix in $\|\cdot\|_1$ can be really sparse, with lots of zero elements. Accordingly, when the traditional empirical risk $\mathcal{L}_{\mathrm{NCE}}(\boldsymbol{\varphi}_k)$ in (2) is introduced, the basic learning objective of the kth layer is formulated as

$$\min_{\boldsymbol{\varphi}_k} \mathcal{L}_{\mathrm{NCE}}(\boldsymbol{\varphi}_k) \quad \text{s.t.} \ \|\max(\boldsymbol{\mathcal{D}}_{::k} - \tau, 0)\|_1 \le r_k \tag{5}$$

where $k = 1, 2, \ldots, C$ and $r_1 > r_2 > \cdots > r_C > 0$ are expected sparsity degrees for the C projection layers. However, the above constrained optimization is hard to solve in practical implementations, so we consider to convert it to a regularized form to remove the constraint. Specifically, to obtain different sparsity degrees, we constrain $\|\max(\boldsymbol{\mathcal{D}}_{::1} - \tau, 0)\|_1, \|\max(\boldsymbol{\mathcal{D}}_{::2} - \tau, 0)\|_1, \ldots, \|\max(\boldsymbol{\mathcal{D}}_{::C} - \tau, 0)\|_1$ with different weights, and thus, we build the following weighted regularization term to constrain each projection layer:

$$\mathcal{R}_k(\boldsymbol{\varphi}_k) = \lambda_k \|\max(\boldsymbol{\mathcal{D}}_{::k} - \tau, 0)\|_1 \tag{6}$$

where $k = 1, 2, \ldots, C$ and the regularization parameters $\lambda_C > \lambda_{C-1} > \cdots > \lambda_1 > 0$ are tuned by users. Here, the largest parameter λ_C controls the upper bound of sparsity. In the limit case, $\lambda_C \to \infty$ will lead to a trivial solution $\boldsymbol{\mathcal{P}}_{::k} = \boldsymbol{0}$ and will regard all instances as similar ones. Such regularization terms actually play a role of *self-supervision in a bottom-up manner*. In this way, we roughly enumerate potential similarities for different scenarios in advance so that we can further enrich the self-supervisory information even though pairwise similarities are unknown to us.

3) Vector-Valued Learning Objective: When we integrate the above regularizers $\mathcal{R}_1(\boldsymbol{\varphi}_1), \mathcal{R}_2(\boldsymbol{\varphi}_2), \ldots, \mathcal{R}_C(\boldsymbol{\varphi}_C)$ into the conventional CL, we obtain C learning objectives, and the kth objective is $\mathcal{L}_{\mathrm{NCE}}(\boldsymbol{\varphi}_k)+\alpha\mathcal{R}_k(\boldsymbol{\varphi}_k)$ where $\alpha > 0$. Actually, this is a direct extension of the Karush-Kuhn–Tucker condition in the augmented Lagrangian optimization [5], and there always exits $\alpha > 0$ such that minimizing $\mathcal{L}_{\mathrm{NCE}}(\boldsymbol{\varphi}_k)+\alpha\mathcal{R}_k(\boldsymbol{\varphi}_k)$ is equivalent to (5). Now, we want to simultaneously minimize these C objectives to obtain a shared feature representation (i.e., $\boldsymbol{\varphi}_0$ and its corresponding $\boldsymbol{\Phi}$), and thus, we construct the following SACL:

$$\begin{aligned} &\min_{\boldsymbol{\mathcal{P}},\boldsymbol{\varphi}_0} \boldsymbol{\mathcal{F}}\left(\boldsymbol{\mathcal{P}},\boldsymbol{\varphi}_0\right) = \left(f_1\left(\boldsymbol{\mathcal{P}}_{::1},\boldsymbol{\varphi}_0\right),\ldots,f_C\left(\boldsymbol{\mathcal{P}}_{::C},\boldsymbol{\varphi}_0\right)\right)^\top \\ &\text{s.t. } f_k\left(\boldsymbol{\mathcal{P}}_{::k},\boldsymbol{\varphi}_0\right) = \mathcal{L}_{\mathrm{NCE}}\left(\boldsymbol{\varphi}_k\right)+\alpha\mathcal{R}_k\left(\boldsymbol{\varphi}_k\right) \end{aligned} \tag{7}$$

where the kth projection layer is $\boldsymbol{\varphi}_k = \boldsymbol{\mathcal{P}}_{::k}\cdot\boldsymbol{\varphi}_0$ for $k = 1,2,\ldots,C$, and $\alpha > 0$ is tuned by users to balance the importance of regularization and empirical risk. Note that all $\boldsymbol{\varphi}_0,\boldsymbol{\varphi}_1,\ldots,\boldsymbol{\varphi}_C$ depend on the encoder $\boldsymbol{\Phi}$, so any constraints on projection layers will finally affect the training of encoder $\boldsymbol{\Phi}$ (i.e., the feature representation). At this point, (7) actually aims to learn a generalizable feature representation $\boldsymbol{\Phi}$ that is able to classify data when facing diverse annotations in different scenarios, e.g., different fineness degrees of categorization and different learning targets. It is worth pointing out that (7) is a typical MOP problem, where the *absolute optimal solution*[1] that concurrently minimizes each objective f_k does not necessarily exist, but its *Pareto optimal solution* [2] can always exist [46], [54]. In Section III-C, we will equivalently convert the above vector-valued function to a real-valued function and also provide a stochastic algorithm to solve the converted problem.

C. Optimization

Now, we provide the detailed optimization algorithm to solve our proposed SACL in (7). As both $\mathcal{L}_{\mathrm{NCE}}(\boldsymbol{\varphi}_k)$ and $\mathcal{R}_k(\boldsymbol{\varphi}_k)$ are nonnegative for any $k = 1,2,\ldots,C$, we consider converting the vector-valued function $\boldsymbol{\mathcal{F}}(\boldsymbol{\mathcal{P}},\boldsymbol{\varphi}_0)$ to a summation of all elements. Specifically, we let

$$\begin{aligned} \mathcal{J}\left(\boldsymbol{\mathcal{P}},\boldsymbol{\varphi}_0\right) &= \frac{1}{C}\sum_{k=1}^{C}\mathcal{L}_{\mathrm{NCE}}\left(\boldsymbol{\varphi}_k\right)+\alpha\frac{1}{C}\sum_{k=1}^{C}\mathcal{R}_k\left(\boldsymbol{\varphi}_k\right) \\ &= \mathcal{L}\left(\left\{\boldsymbol{\mathcal{P}}_{::k}\cdot\boldsymbol{\varphi}_0\right\}_{k=1}^{C}\right)+\alpha\mathcal{R}\left(\left\{\boldsymbol{\mathcal{P}}_{::k}\cdot\boldsymbol{\varphi}_0\right\}_{k=1}^{C}\right) \end{aligned} \tag{8}$$

which can also be regarded as optimizing the average of all learning objectives in (7). To be more rigorous, we provide the following theorem to guarantee the equivalence between the optimal solutions of (8) and (7).

Theorem 1: Assume that $(\boldsymbol{\mathcal{P}}^*,\boldsymbol{\varphi}_0^*) \in \arg\min_{\boldsymbol{\mathcal{P}},\boldsymbol{\varphi}_0}\mathcal{J}(\boldsymbol{\mathcal{P}},\boldsymbol{\varphi}_0)$, and then, we have that $(\boldsymbol{\mathcal{P}}^*,\boldsymbol{\varphi}_0^*)$ is always a Pareto optimal solution to $\boldsymbol{\mathcal{F}}(\boldsymbol{\mathcal{P}},\boldsymbol{\varphi}_0)$. Furthermore, if $\boldsymbol{\mathcal{F}}(\boldsymbol{\mathcal{P}},\boldsymbol{\varphi}_0)$ has absolute optimal solutions, then we have that $(\boldsymbol{\mathcal{P}}^*_{::k},\boldsymbol{\varphi}_0^*) \in \arg\min_{\boldsymbol{\mathcal{P}}_{::k},\boldsymbol{\varphi}_0} f_k(\boldsymbol{\mathcal{P}}_{::k},\boldsymbol{\varphi}_0)$ for any $k = 1,2,\ldots,C$.

[1]Here, $(\widetilde{\boldsymbol{\mathcal{P}}},\widetilde{\boldsymbol{\varphi}}_0)$ is an absolute optimal solution to $\boldsymbol{\mathcal{F}}$ if and only if $f_k(\widetilde{\boldsymbol{\mathcal{P}}}_{::k},\widetilde{\boldsymbol{\varphi}}_0) \le f_k(\boldsymbol{\mathcal{P}}_{::k},\boldsymbol{\varphi}_0)$ $(k = 1,2,\ldots,C)$ for any given $(\boldsymbol{\mathcal{P}},\boldsymbol{\varphi}_0)$.

[2]Here, $(\boldsymbol{\mathcal{P}}^*,\boldsymbol{\varphi}_0^*)$ is a Pareto optimal solution to $\boldsymbol{\mathcal{F}}$ if and only if there does not exist $(\boldsymbol{\mathcal{P}},\boldsymbol{\varphi}_0)$ such that $f_k(\boldsymbol{\mathcal{P}}_{::k},\boldsymbol{\varphi}_0) < f_k(\boldsymbol{\mathcal{P}}^*_{::k},\boldsymbol{\varphi}_0^*)$ for all $k = 1,2,\ldots,C$.

Algorithm 1 Solving (8) via SGD

Input: training data $\mathscr{X} = \{\boldsymbol{x}_i\}_{i=1}^N$; step size $\eta > 0$; regularization parameter $\alpha > 0$; batch size $n \in \mathbb{N}_+$.

Initialize: iteration number $t = 0$; sparsity parameters $\lambda_C > \lambda_{C-1} > \cdots > \lambda_1$; random $\boldsymbol{\mathcal{P}}^{(0)}$ and $\boldsymbol{\varphi}_0^{(0)}$.

For t from 1 to T:

1). Uniformly pick $(n+1)$ instances $\{\boldsymbol{x}_{b_j}\}_{j=1}^{n+1}$ from $\mathscr{X}$;

2). Compute the gradients of $\mathcal{L}(\boldsymbol{\mathcal{P}},\boldsymbol{\varphi}_0;\{\boldsymbol{x}_{b_j}\}_{j=1}^{n+1})$ and $\mathcal{R}(\boldsymbol{\mathcal{P}},\boldsymbol{\varphi}_0;\{\boldsymbol{x}_{b_j}\}_{j=1}^{n+1})$ (denoted as $\{\nabla_{\boldsymbol{\mathcal{P}}}\mathcal{L}_B,\nabla_{\boldsymbol{\varphi}_0}\mathcal{L}_B\}$ and $\{\nabla_{\boldsymbol{\mathcal{P}}}\mathcal{R}_B,\nabla_{\boldsymbol{\varphi}_0}\mathcal{R}_B\}$, respectively) via Eq. (9);

3). Update the learning parameters:

$$\begin{cases} \boldsymbol{\mathcal{P}}^{(t)} = \boldsymbol{\mathcal{P}}^{(t-1)} - \eta\left(\nabla_{\boldsymbol{\mathcal{P}}}\mathcal{L}_B + \alpha\nabla_{\boldsymbol{\mathcal{P}}}\mathcal{R}_B\right), \\ \boldsymbol{\varphi}_0^{(t)} = \boldsymbol{\varphi}_0^{(t-1)} - \eta\left(\nabla_{\boldsymbol{\varphi}_0}\mathcal{L}_B + \alpha\nabla_{\boldsymbol{\varphi}_0}\mathcal{R}_B\right), \end{cases} \tag{10}$$

End.

Output: the converged $\boldsymbol{\mathcal{P}}^{(T)}$ and $\boldsymbol{\varphi}_0^{(T)}$.

The proof is given in the Supplementary Material. The above result clearly reveals that (7) and (8) actually share the same optimal solution $\boldsymbol{\varphi}_0^*$, which means that we can directly optimize (8) to obtain a final encoder for the downstream tasks. Now, we provide a stochastic algorithm to solve it.

Stochastic Loss: Minimizing the objective function $\mathcal{J}(\boldsymbol{\mathcal{P}},\boldsymbol{\varphi}_0)$ in (8) is a typical batch optimization problem [91], where both the empirical risk $\mathcal{L} = 1/C\sum_{k=1}^{C}\mathcal{L}_{\mathrm{NCE}}(\boldsymbol{\varphi}_k)$ and the regularizer $\mathcal{R} = 1/C\sum_{k=1}^{C}\mathcal{R}_k(\boldsymbol{\varphi}_k)$ involve all training data. Therefore, we adopt the stochastic gradient descent (SGD) method [34] to solve it, and here, we demonstrate the stochastic gradient for the objective function $\mathcal{J}(\boldsymbol{\mathcal{P}},\boldsymbol{\varphi}_0)$. Specifically, for $n+1$ (i.e., the batch size) randomly selected data points $\{\boldsymbol{x}_{b_j}|\boldsymbol{x}_{b_j} \in \mathscr{X}\}_{j=1}^{n+1}$ ($\boldsymbol{b}$ is the index vector of the mini-batch), the NCE loss defined by (2) already has a stochastic form,[3] so here we only need to demonstrate the stochastic regularizer for $\boldsymbol{b}$, namely

$$\begin{aligned} &C\mathcal{R}\left(\left\{\boldsymbol{\mathcal{P}}_{::k}\cdot\boldsymbol{\varphi}_0\right\}_{k=1}^{C};\left\{\boldsymbol{x}_{b_j}\right\}_{j=1}^{n+1}\right) \\ &= \sum_{k=1}^{C}\lambda_k\left\|\max\left(\boldsymbol{\mathcal{D}}_{::k}\left(\boldsymbol{\mathcal{P}}_{::k}\cdot\boldsymbol{\varphi}_0;\left\{\boldsymbol{x}_{b_j}\right\}_{j=1}^{n+1}\right)-\tau,0\right)\right\|_1 \\ &= \sum_{k=1}^{C}\sum_{i,j=1}^{n+1}\lambda_k\max\left(d_{\boldsymbol{\mathcal{P}}_{::k}\boldsymbol{\varphi}_0}\left(\boldsymbol{x}_{b_i},\boldsymbol{x}_{b_j}\right)-\tau,0\right) \end{aligned} \tag{9}$$

which merely depends on the mini-batch data $\{\boldsymbol{x}_{b_j}\}_{j=1}^{n+1}$. It means that our method can be easily implemented in most existing CL methods by only introducing very little computational overhead. Based on the above stochastic loss, we further provide the SGD iteration steps in Algorithm 1 to optimize (8).

Here, we further investigate the convergence behavior of iteration points $\{\boldsymbol{\mathcal{P}}^{(1)},\boldsymbol{\varphi}_0^{(1)}\}$, $\{\boldsymbol{\mathcal{P}}^{(2)},\boldsymbol{\varphi}_0^{(2)}\}$, $\ldots$, $\{\boldsymbol{\mathcal{P}}^{(T)},\boldsymbol{\varphi}_0^{(T)}\}$ obtained by Algorithm 1. We prove that the gradients of iteration points of our final learning objective $\mathcal{J}(\boldsymbol{\mathcal{P}},\boldsymbol{\varphi}_0)$ will

[3]For a mini-batch, the stochastic loss $\mathcal{L}_{\mathrm{NCE}}(\boldsymbol{\varphi}_0;\{\boldsymbol{x}_{b_j}\}_{j=1}^{n+1}) = -\log\left[\exp(-d_{\boldsymbol{\varphi}_0}(\boldsymbol{x}_{b_{n+1}},\boldsymbol{x}_{b_{n+1}}^+))/\left(\exp(-d_{\boldsymbol{\varphi}_0}(\boldsymbol{x}_{b_{n+1}},\boldsymbol{x}_{b_{n+1}}^+))+\sum_{j=1}^{n}\exp(-d_{\boldsymbol{\varphi}}(\boldsymbol{x}_{b_j},\boldsymbol{x}_{b_{n+1}}))\right)\right]$. The mini-batch index vector $\boldsymbol{b} = (b_1,b_2,\ldots,b_{n+1})^\top$, where $b_i, = 1,2,\ldots,N$, $b_i \neq b_j$, $i,j = 1,2,\ldots,n+1$.

gradually converge to 0 even though it has two independent variables $\mathcal{P}$ and $\boldsymbol{\varphi}_0$.

Theorem 2: We assume that function $\mathcal{J}(\mathcal{P}, \boldsymbol{\varphi}_0)$ has a δ-bounded gradient ($\|\nabla \mathcal{J}(\mathcal{P}, \boldsymbol{\varphi}_0)\|_2 < \delta$) and let $\eta = (2(\mathcal{J}(\mathcal{P}^{(0)}, \boldsymbol{\varphi}_0^{(0)}) - \mathcal{J}(\mathcal{P}^*, \boldsymbol{\varphi}_0^*))/(S\delta^2 T))^{1/2}$. Then, for iteration points in Algorithm 1, we have $\min_{0\le t\le T} \mathbb{E}[\,\|\nabla \mathcal{J}(\mathcal{P}^{(t)}, \boldsymbol{\varphi}_0^{(t)})\|_2] \le \sqrt{2S\Delta\mathcal{J}/T}\delta$, where $\Delta\mathcal{J} = \mathcal{J}(\mathcal{P}^{(0)}, \boldsymbol{\varphi}_0^{(0)}) - \mathcal{J}(\mathcal{P}^*, \boldsymbol{\varphi}_0^*)$ and $S > 0$ is a Lipschitz constant such that $\|\nabla \mathcal{J}(\mathcal{P}, \boldsymbol{\varphi}_0) - \nabla \mathcal{J}(\mathcal{P}', \boldsymbol{\varphi}_0')\|_2 \le S\,\|[\mathcal{P}, \boldsymbol{\varphi}_0] - [\mathcal{P}', \boldsymbol{\varphi}_0']\|_2$.

The stationary point of the learning objective can be found during the training phase by setting a sentry to save the minimum gradient and the corresponding iteration results. However, in practice, we usually do not need to set such a sentry but just simply use the finally iterated point. Note that in the above theorem, variables $S, \Delta\mathcal{J}$, and δ are all independent of T. It means that the iteration points $\{\mathcal{P}^{(1)}, \boldsymbol{\varphi}_0^{(1)}\}, \ldots, \{\mathcal{P}^{(T)}, \boldsymbol{\varphi}_0^{(T)}\}$ will converge to a stationary point of the learning objective $\mathcal{J}$ with a convergence rate $\mathcal{O}(1/\sqrt{T})$, where T is the number of iterations.

IV. Theoretical Analyses

In this section, we further provide in-depth theoretical analyses. We investigate the bottom-up sparsity of learned distances and the generalization ability of our learning algorithm to demonstrate its soundness and effectiveness. All proofs are given in the Supplementary Material.

A. Bottom-Up Sparsity of the Pairwise Distance

In Section III-B, we have built a weighted regularization term [i.e., (6)] to provide the critical self-supervision for pairwise distances. We expect that pairwise distances will gradually become sparse from the bottom projection layers to the top projection layers. Now, we would like to investigate whether the learning objective can really result in such bottom-up sparsity.

We suppose that $\{\mathcal{P}^*, \boldsymbol{\varphi}_0^*\}$ is an optimal solution to $\mathcal{J}(\mathcal{P}, \boldsymbol{\varphi}_0)$ and we have the following theorem to reveal the sparsity of the distance matrix in each projection layer.

Theorem 3: For any given constants $\alpha, \tau > 0$, and $\lambda_C > \lambda_{C-1} > \cdots > \lambda_1 > 0$, we assume that $\{\mathcal{P}^*, \boldsymbol{\varphi}_0^*\} \in \arg\min_{\mathcal{P}, \boldsymbol{\varphi}_0} \mathcal{J}(\mathcal{P}, \boldsymbol{\varphi}_0)$ and the corresponding distance tensor $\mathcal{D}(\{\boldsymbol{\varphi}_k^*\}_{k=1}^C) = [\mathcal{D}_{ijk}^*] \in \mathbb{R}^{N\times N\times C}$. Then, we have that

$$\left\|\max\left(\mathcal{D}_{::k}^* - \tau, 0\right)\right\|_1 > \left\|\max\left(\mathcal{D}_{::(k+1)}^* - \tau, 0\right)\right\|_1 \tag{11}$$

where $\boldsymbol{\varphi}_k^* = \mathcal{P}_{::k}^* \cdot \boldsymbol{\varphi}_0^*$ for $k = 1, 2, \ldots, C-1$.

The above theorem clearly reveals that the optimal solution to the learning objective of SACL will necessarily lead to an effect of bottom-up sparsity with the increase of k from 1 to C. It means that different projection layers can effectively provide different supervisions for pairwise similarities. In the bottom projection layers, pairwise distances are nonsparse, and this makes each data point tend to form itself as a cluster to learn the difference between instances. In the top projection layers, pairwise distances are sparse, and the potential commonality between instances will be learned. As different layers can successfully capture different types of similarities, the generalization ability of learned embedding $\boldsymbol{\varphi}_0$ can be improved.

B. Generalization Error Bound

Compared with the conventional CL approach with a single encoder $\boldsymbol{\Phi}$ and projection head $\boldsymbol{P}_0$, our SACL introduces more learning parameters. However, here we would like to prove that our learning algorithm actually tightens the generalization error bound (GEB) of the conventional CL approach.

As we know, the GEB [10] usually has a convergence rate of $\mathcal{O}(1/\sqrt{N})$ for an empirical risk minimization model, where N is the sample size. Here, we are not going to investigate the convergence rate with respect to the sample size but show a tightened GEB result benefited from the regularization term $\mathcal{R}$ for validating the effectiveness of our method. Specifically, for the underlying data distribution $\mathscr{D}$, we denote the expected risk $\widetilde{\mathcal{L}}(\{\mathcal{P}_{::k}\boldsymbol{\varphi}_0\}_{k=1}^C; \mathscr{D}) = \mathbb{E}_{\{z_i|z_i\sim\mathscr{D}\}_{i=1}^N}[\mathcal{L}(\{\mathcal{P}_{::k}\boldsymbol{\varphi}_0\}_{k=1}^C; \{z_i\}_{i=1}^N)]$ and discuss how far it is from the empirical risk $\mathcal{L}(\{\mathcal{P}_{::k}\boldsymbol{\varphi}_0\}_{k=1}^C; \mathscr{X})$ on the training data. The error bound is described as follows.

Theorem 4: For any $(\mathcal{P}, \boldsymbol{\varphi}_0)$ learned from the objective $\mathcal{J}(\mathcal{P}, \boldsymbol{\varphi}_0)$ and any given constant $\delta \in (0, 1)$, we have that with probability at least $1 - \delta$

$$\begin{aligned}&\left|\mathcal{L}\left(\left\{\mathcal{P}_{::k}\cdot\boldsymbol{\varphi}_0\right\}_{k=1}^C; \mathscr{X}\right) - \widetilde{\mathcal{L}}\left(\left\{\mathcal{P}_{::k}\cdot\boldsymbol{\varphi}_0\right\}_{k=1}^C; \mathscr{D}\right)\right| \\ &\quad\le \xi(\alpha)\,\omega(n)\max\left(\mathcal{D}\left(\left\{\mathcal{P}_{::k}\cdot\boldsymbol{\varphi}_0\right\}_{k=1}^C\right)\right)\sqrt{\frac{\ln(2/\delta)}{2N}}\end{aligned} \tag{12}$$

where $\xi(\alpha) = (C+2/C)/\alpha$ is monotone decreasing with respect to α and $\omega(n) = \log\left(e^2/n + 1\right)$ is monotone decreasing with respect to n.

From the above result in (12), it is easy to observe that the error bound is dominated by two main aspects. First, the error bound gradually decreases with the increase of the sampling number N as well as the batch size n. This is consistent with the traditional GEB and empirical observations in existing CL [53]. More importantly, we can find that such an error bound becomes *tighter* with the increase of α, and thus, the regularization term $\mathcal{R}$ can assist the expected risk in converging to the empirical risk. Therefore, Theorem 4 demonstrates that our method effectively improves the generalization ability of conventional CL algorithms.

C. Equivalence to the Stacked Form

Here, we further consider a stacked form of our method, where the kth projection layer is a multiplicative result of the first k projection matrices, i.e., $\boldsymbol{\varphi}_k = \left(\Pi_{l=1}^k \mathcal{P}_{::l}\right)\cdot\boldsymbol{\varphi}_0$, and the corresponding learning objective becomes to $\widehat{\mathcal{J}}(\mathcal{P}, \boldsymbol{\varphi}_0) = \mathcal{L}\left(\left\{\left(\Pi_{l=1}^k \mathcal{P}_{::l}\right)\cdot\boldsymbol{\varphi}_0\right\}_{k=1}^C\right) + \alpha\mathcal{R}\left(\left\{\left(\Pi_{l=1}^k \mathcal{P}_{::l}\right)\cdot\boldsymbol{\varphi}_0\right\}_{k=1}^C\right)$, which can also be regarded as another straightforward implementation of our method. Now, we would like to prove that this stacked form is actually equivalent to our original parallel form in (8).

To be more specific, we would like to investigate whether the optimal solution to $\mathcal{J}(\mathcal{P}, \boldsymbol{\varphi}_0)$ can also minimize the stacked form loss $\widehat{\mathcal{J}}(\mathcal{P}, \boldsymbol{\varphi}_0)$.

Theorem 5: Assume that $\{\mathcal{P}^*, \boldsymbol{\varphi}_0^*\}$ is an optimal solution to the learning objective $\mathcal{J}(\mathcal{P}, \boldsymbol{\varphi}_0)$ in (8). Then, we have that there exists a mapping $\boldsymbol{\Theta} : \mathbb{R}^{h\times h} \to \mathbb{R}^{h\times h}$ such that $\mathcal{J}(\mathcal{P}^*, \boldsymbol{\varphi}_0^*) = \widehat{\mathcal{J}}(\boldsymbol{\Theta}(\mathcal{P}^*), \boldsymbol{\varphi}_0^*)$ and $\{\boldsymbol{\Theta}(\mathcal{P}^*), \boldsymbol{\varphi}_0^*\} \in \arg\min_{\mathcal{P}, \boldsymbol{\varphi}_0} \widehat{\mathcal{J}}(\mathcal{P}, \boldsymbol{\varphi}_0)$, where the mapping $\boldsymbol{\Theta}$ is independent of $\mathcal{P}^*$.

From the above theorem, we can clearly find that both two learning objectives $\mathcal{J}(\boldsymbol{\mathcal{P}}, \boldsymbol{\varphi}_0)$ and $\widehat{\mathcal{J}}(\boldsymbol{\mathcal{P}}, \boldsymbol{\varphi}_0)$ actually have the same optimal representation $\boldsymbol{\varphi}_0^*$, and thus, the formulation in (8) can also provide a stacked mechanism to consider the hierarchical information within data. It also means that the effectiveness of our method is independent of the architectural style of those C projection layers, which makes our method very easy to implement in practical uses.

V. Experimental Results

In this section, we show experimental results on real-world datasets to validate the effectiveness of our proposed method. In detail, we first conduct ablation studies to reveal the usefulness of our newly introduced block/regularizer. Then, we compare our proposed learning algorithm with existing state-of-the-art models on vision, language, and graph-related tasks. The pretraining process is implemented on Pytorch [48] with NVIDIA TeslaV100 GPUs. We adopt the encoder result $\boldsymbol{\Phi}(\cdot)$ for feature extraction, where the regularization parameter α is fixed to 0.5 and we use five projection layers for capturing similarities. Parameters λ_1–λ_5 are fixed to 2, 4, 8, 16, and 32, respectively. The dimensionality of projection space (i.e., h) and the temperature parameter [i.e., γ in (2)] are set to 512 and 0.2, respectively. The hyperparameters of compared methods are set to the recommended values according to their original papers. More detailed settings are discussed in each subsection.

A. Ablation Studies and Visualization Results

In this section, we conduct ablation studies on the effectiveness of the introduced components in our method. We use the STL-10 and CIFAR-10 datasets to train two representative baseline methods (including *SimCLR* [11], the negative-free method *BYOL* [24], and the adversarial method *CaCo* [30], [66]) and different implementations of SACL with different numbers of projection layers. We train all models with 400 epochs with the same batch size and learning rate, and we record the test accuracy by fine-tuning a linear *softmax* [23]. The loss functions of the two types of SACL (w/ neg and adv neg) are InfoNCE loss, and the loss function of SACL (w/o neg) is mean-squared loss, as recommended by the baseline methods. It is noteworthy that most negative-free methods focus on how to construct informative and reliable positive pairs (e.g., BYOL aligns the outputs of two independent networks input with the same example). In contrast, our SACL focuses on how to explore useful information from the negative pairs by introducing the similarity agnosticity (merely for negative pairs). Meanwhile, most adversarial learning-based methods aim to generate critical positive pairs or unearth hard negative pairs, so they actually enrich the training data for learning their encoders. In contrast, our SACL aims to explore the potential self-supervisory information by only using the existing training data.

We record the test accuracy (mean $\pm$ std, five random trials) of compared methods at the 400th epoch in Table II. First, we can observe that our method can collaborate very well with two representative baseline methods. Our SACL obtains relatively stable performance in various cases with different batch sizes and numbers of projection layers. Then, we can clearly find that introducing the regularization term $\mathcal{R}$ consistently improves both baseline methods on two datasets. Meanwhile, another interesting phenomenon is that, with the increase in layer number (from $C = 1$ to $C = 5$), the classification accuracy also correspondingly increases. This successfully validates that it is indeed useful to introduce alternative supervision via multiple projection layers. Furthermore, we also perform the t-test at a significance level 0.05 in the last column, and "✓" indicates that our method is significantly better than the best baseline result. Since SimCLR is a negative sample-based method, while BYOL is a negative-free method, the consistently improved results demonstrate that our SACL is applicable to different types of CL approaches. SACL remains competitive with negative-free methods such as BYOL by enriching the supervision signal beyond what traditional instance discrimination provides. One key reason is that SACL leverages a multiobjective framework

TABLE II

Classification Accuracy Rates (Mean $\pm$ std) of Baseline Methods and Our Method on STL-10 and CIFAR-10 Datasets With 400 Epochs and Batch Size (Negative Sample Size) = 256 and 512

Method	STL-10		CIFAR-10	
	256	512	256	512
SimCLR (w/ neg., $C = 0$)	76.2 $\pm$ 1.1	79.2 $\pm$ 0.5	89.3 $\pm$ 2.1	92.3 $\pm$ 0.4
BYOL (w/o neg., $C = 0$)	75.2 $\pm$ 0.1	79.2 $\pm$ 0.1	90.3 $\pm$ 0.6	92.2 $\pm$ 0.6
CaCo (adv. neg., $C = 0$)	75.3 $\pm$ 0.2	79.5 $\pm$ 0.3	90.2 $\pm$ 0.5	92.6 $\pm$ 0.2
SACL (w/ neg., $C = 1$)	76.2 $\pm$ 0.1	80.2 $\pm$ 0.1	89.3 $\pm$ 0.1	92.5 $\pm$ 0.1
SACL (w/o neg., $C = 1$)	76.5 $\pm$ 0.5	80.5 $\pm$ 0.5	90.5 $\pm$ 1.2	92.5 $\pm$ 1.2
SACL (adv. neg., $C = 1$)	76.4 $\pm$ 0.4	80.5 $\pm$ 0.2	91.1 $\pm$ 0.7	92.9 $\pm$ 0.8
SACL (w/ neg., $C = 3$)	77.1 $\pm$ 0.3	81.1 $\pm$ 0.3 ✓	91.4 $\pm$ 2.6	93.4 $\pm$ 0.1 ✓
SACL (w/o neg., $C = 3$)	77.3 $\pm$ 0.5 ✓	82.3 $\pm$ 0.5 ✓	92.5 $\pm$ 0.2 ✓	93.5 $\pm$ 0.1 ✓
SACL (adv. neg., $C = 3$)	77.9 $\pm$ 0.4 ✓	82.7 $\pm$ 0.3 ✓	92.7 $\pm$ 0.1 ✓	93.8 $\pm$ 0.1 ✓
SACL (w/ neg., $C = 5$)	79.5 $\pm$ 0.2 ✓	**84.5 $\pm$ 0.2** ✓	93.5 $\pm$ 0.3 ✓	**95.6 $\pm$ 0.3** ✓
SACL (w/o neg., $C = 5$)	79.6 $\pm$ 0.5 ✓	**84.6 $\pm$ 0.5** ✓	94.1 $\pm$ 0.3 ✓	95.1 $\pm$ 0.3 ✓
SACL (adv. neg., $C = 5$)	**79.9 $\pm$ 0.2** ✓	83.2 $\pm$ 0.5 ✓	**94.4 $\pm$ 0.3** ✓	95.4 $\pm$ 0.1 ✓

TABLE III

Investigation of Our Method Influenced by Different Regularization Weights λ_1–λ_5 on STL-10 and CIFAR-10 Datasets With 400 Epochs and Batch Size (Negative Sample Size) = 256 and 512

Parameters $(\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5)$	STL-10		CIFAR-10	
	256	512	256	512
$(4, 4, 4, 4, 4)$, w/ neg.	75.2 $\pm$ 0.2	79.2 $\pm$ 0.4	88.3 $\pm$ 1.2	93.3 $\pm$ 1.4
$(4, 4, 4, 4, 4)$, w/o neg.	75.8 $\pm$ 0.2	79.8 $\pm$ 1.8	90.5 $\pm$ 0.6	92.1 $\pm$ 0.6
$(8, 8, 8, 8, 8)$, w/ neg.	76.7 $\pm$ 0.5	82.1 $\pm$ 0.5	92.4 $\pm$ 1.6	93.3 $\pm$ 0.4
$(8, 8, 8, 8, 8)$, w/o neg.	77.2 $\pm$ 0.5	81.3 $\pm$ 0.7	92.1 $\pm$ 0.9	92.5 $\pm$ 0.1
$(2, 4, 8, 16, 32)$, w/ neg.	**79.5 $\pm$ 0.2**	**84.5 $\pm$ 0.2**	**93.5 $\pm$ 0.3**	**95.6 $\pm$ 0.3**
$(2, 4, 8, 16, 32)$, w/o neg.	**79.6 $\pm$ 0.5**	**84.6 $\pm$ 0.5**	**94.1 $\pm$ 0.3**	**95.1 $\pm$ 0.3**
Setting zero values for $(\lambda_1, \lambda_2, \lambda_3)$, $C = 3$				
$(2, 4, 8)$, w/ neg.	**77.1 $\pm$ 0.3**	**81.1 $\pm$ 0.3**	**91.4 $\pm$ 2.6**	**93.4 $\pm$ 0.1**
$(0, 4, 8)$, w/ neg.	76.5 $\pm$ 0.2 ↓	80.5 $\pm$ 0.4 ↓	90.4 $\pm$ 1.6 ↓	92.8 $\pm$ 0.2 ↓
$(2, 0, 8)$, w/ neg.	76.3 $\pm$ 0.2 ↓	80.6 $\pm$ 0.3 ↓	90.6 $\pm$ 1.8 ↓	93.0 $\pm$ 0.3 ↓
$(2, 4, 0)$, w/ neg.	76.7 $\pm$ 0.4 ↓	80.5 $\pm$ 0.2 ↓	90.7 $\pm$ 2.8 ↓	92.6 $\pm$ 0.3 ↓
Setting zero values for $(\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5)$, $C = 5$				
$(2, 4, 8, 16, 32)$, w/ neg.	**79.5 $\pm$ 0.2**	**84.5 $\pm$ 0.2**	**93.5 $\pm$ 0.3**	**95.6 $\pm$ 0.3**
$(0, 4, 8, 16, 32)$, w/ neg.	77.8 $\pm$ 0.8 ↓	82.5 $\pm$ 0.4 ↓	92.3 $\pm$ 0.3 ↓	94.6 $\pm$ 0.2 ↓
$(2, 0, 8, 16, 32)$, w/ neg.	78.2 $\pm$ 0.5 ↓	82.4 $\pm$ 0.3 ↓	92.5 $\pm$ 0.4 ↓	94.2 $\pm$ 0.2 ↓
$(2, 4, 0, 16, 32)$, w/ neg.	78.5 $\pm$ 0.3 ↓	83.1 $\pm$ 0.5 ↓	92.8 $\pm$ 0.3 ↓	94.6 $\pm$ 0.3 ↓
$(2, 4, 8, 0, 32)$, w/ neg.	79.1 $\pm$ 0.3 ↓	82.8 $\pm$ 0.3 ↓	92.1 $\pm$ 0.5 ↓	94.2 $\pm$ 0.5 ↓
$(2, 4, 8, 16, 0)$, w/ neg.	78.4 $\pm$ 0.4 ↓	82.5 $\pm$ 0.3 ↓	91.9 $\pm$ 0.1 ↓	93.8 $\pm$ 0.2 ↓

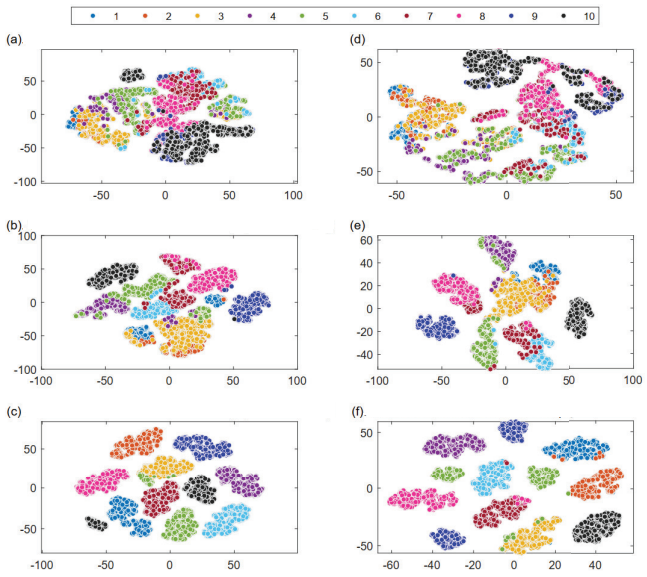


Fig. 4. T-SNE visualizations of our SACL on CIFAR-10 and STL-10 datasets. We can observe that SACL (layer number >1) can successfully obtain the better separability than the baseline result (layer number = 1), and this clearly demonstrates the effectiveness of the alterable supervision component in our method. (a) Visualizations of SACL on CIFAR-10 (layer number = 1). (b) Visualizations of SACL on CIFAR-10 (layer number = 3). (c) Visualizations of SACL on CIFAR-10 (layer number = 5). (d) Visualizations of SACL on STL-10 (layer number = 1). (e) Visualizations of SACL on STL-10 (layer number = 3). (f) Visualizations of SACL on STL-10 (layer number = 5).

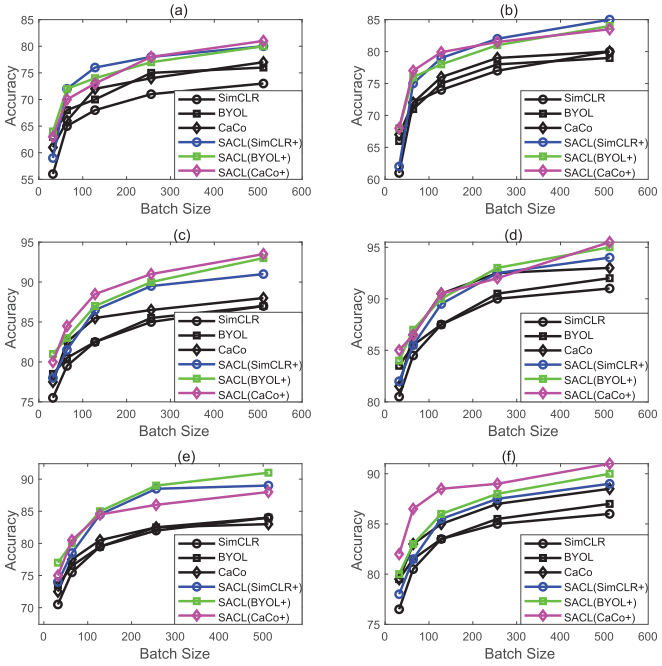


Fig. 5. Accuracy rates of compared methods on STL-10, CIFAR-10, and CIFAR-100 datasets, where the negative sample size is from 32 to 512. (a) STL-10 with 100 epochs. (b) STL-10 with 400 epochs. (c) CIFAR-10 with 100 epochs. (d) CIFAR-10 with 400 epochs. (e) CIFAR-100 with 100 epochs. (f) CIFAR-100 with 400 epochs.

that deploys multiple projection layers with distinct regularization strengths. This design allows the model to capture a richer spectrum of pairwise relationships; while negative-free methods like BYOL focus on pulling together augmented views of the same instance and rely on implicit mechanisms to avoid collapse, SACL explicitly enforces both similarity and dissimilarity constraints across different projection layers. Moreover, we provide Table III by changing the regularization parameters λ_1–λ_5 to different values, where we set the five parameters to the same values or the increasing values in the corresponding rows. Then, we find that the implementations with increasing regularization parameters achieve the stably better results than the other settings. This is because such increasing parameters can successfully obtain the bottom-up sparsity for pairwise distance matrices, and this also clearly shows the effectiveness of our vector-valued learning objective in (7), which constrains the sparsity degrees of different projection results with different weights.

Visualization Results: We further conduct more detailed investigations on the effectiveness of our method on different baseline frameworks with diverse training epochs and batch sizes (i.e., Fig. 4 and 5). Specifically, in Fig. 5, we visualize the classification accuracy rates of all compared methods on CIFAR-10, CIFAR-100, and STL-10 datasets, where we can observe that our method consistently improves the corresponding baseline results in all scenarios. To be more intuitive, we also conduct the t-SNE embedding [62] to obtain the 2-D data points to better understand the usefulness of our introduced new component. In Fig. 4, SACL (layer number > 1) can successfully obtain the better separability than the baseline result (layer number = 1), where the results of five layers achieve very satisfactory separability. Therefore, it is critically important to maintain the alterable supervision via such multiple projection layers in CL. In Fig. 6, we also visualize the distance matrices of different projection layers learned on STL-10, where the distance matrix of the high-level layer indeed has more zero (or infinitesimal) elements. This new visualization result is also well consistent with our theoretical finding in Theorem 3, and this empirical evidence can further guarantee the soundness of our method.

B. Experiments on Image Data

Here, we first evaluate the effectiveness and superiority of SACL on the image classification task. We implement our method on both the ResNet [29] and vision transformer (ViT) [25] backbones to validate the generalizability of our method. After that, we further investigate the performance of SACL on two transfer learning tasks including *object detection* and *instance segmentation*, where both two tasks focus on how to locate and discriminate visual objects in an image.

1) Image Classification: Specifically, for the image classification task, we first employ *ResNet-50* as our backbone network and implement our method based on the loss functions of SimCLR (negative-used) and BYOL (negative-free), respectively. We train our method on ImageNet-100 and ImageNet-1K datasets [52] and compare it with existing representative approaches including contrastive multiview coding (CMC) [60], SwAV [7], CMC [60], prototypical CL (PCL) [41], hard negative based CL (HCL) [51], meta augmentation (MetAug) [40], low rank promoting CL (LORAC) [69], and

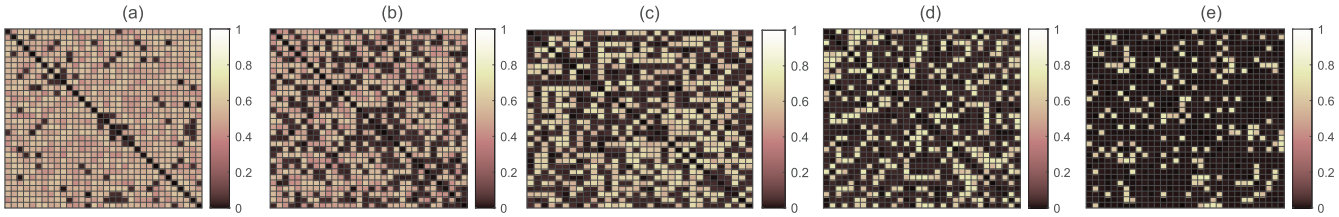


Fig. 6. Distance matrices (32 examples are randomly selected for visualizations) calculated with different layers of our SACL learned on STL-10, where we can clearly observe that the high-level layers exhibit a higher sparsity. (a). Distance matrix of the first layer. (b) Distance matrix of the second layer. (c) Distance matrix of the third layer. (d) Distance matrix of the fourth layer. (e) Distance matrix of the fifth layer.

TABLE IV

CLASSIFICATION ACCURACY (%) OF ALL METHODS ON IMAGENET-100 AND IMAGENET-1K DATASETS. THE BATCH SIZES ARE SET TO 1024 AND 512 FOR RESNET-50 AND VIT-B/16 BACKBONES, RESPECTIVELY. THE OPTIMIZATION RELATED PARAMETERS OF COMPARED METHODS ARE SET ACCORDING TO THEIR RECOMMENDED VALUES. HERE, THE BEST AND SECOND-BEST RESULTS ARE BOLDED AND UNDERLINED, RESPECTIVELY

METHOD	ImageNet-100						ImageNet-1K						#Arch., #Total-Pars.
	100 epochs			400 epochs			300 epochs			800 epochs			
	k-NN	Top-1	Top-5	k-NN	Top-1	Top-5	k-NN	Top-1	Top-5	k-NN	Top-1	Top-5	
SimCLR [11]	55.9	61.3	78.6	70.6	75.2	92.1	64.2	67.4	87.9	66.1	69.3	89.6	ResNet-50, 23M
BYOL [24]	56.3	65.5	77.8	69.2	73.2	90.1	66.9	71.2	90.5	67.2	73.2	91.5	ResNet-50, 23M
CMC [60]	57.7	60.2	79.2	71.6	73.6	92.1	63.2	68.2	87.2	67.2	71.2	89.9	ResNet-50, 47M
PCL [41]	55.9	60.2	77.2	71.5	76.1	93.2	59.5	66.5	86.7	62.2	70.5	90.5	ResNet-50, 23M
SwAV [7]	58.2	61.0	79.4	72.1	75.8	92.9	65.4	74.1	91.2	65.7	75.3	91.5	ResNet-50, 23M
HCL [35]	55.9	60.8	79.3	70.2	74.6	92.3	64.2	71.2	91.2	67.2	71.7	90.7	ResNet-50, 23M
MetAug [40]	59.2	61.1	79.4	69.8	75.6	93.2	65.4	73.2	91.1	67.8	76.0	92.9	ResNet-50, 23M
LORAC [69]	60.1	66.5	78.1	69.5	76.3	92.8	67.2	73.5	91.7	65.8	75.2	91.7	ResNet-50, 23M
INTL [71]	61.2	65.8	78.9	72.9	**79.2**	91.5	67.2	74.2	91.2	67.8	75.9	92.5	ResNet-50, 23M
SACL (neg.-used)	**61.5**	**67.2**	79.9	**74.2**	77.8	93.8	68.3	**74.8**	**92.4**	68.1	**76.6**	92.9	ResNet-50, 23M
SACL (neg.-free)	60.5	66.2	**80.1**	73.5	76.5	**94.5**	**68.9**	73.7	91.9	**69.1**	76.5	**93.2**	ResNet-50, 23M
BYOL [24]	57.2	62.8	77.9	72.1	76.9	93.8	66.6	71.4	91.2	68.2	74.2	92.8	ViT-B/16, 85M
SwAV [7]	60.1	62.5	80.5	74.2	77.8	94.2	64.7	71.8	91.1	69.2	75.6	91.8	ViT-B/16, 85M
Mugs [89]	61.2	66.5	81.5	76.2	78.2	95.2	72.1	75.8	92.1	77.9	80.4	94.5	ViT-B/16, 85M
DINO [8]	61.5	67.5	81.8	78.2	79.2	95.5	72.3	76.1	92.4	76.2	78.2	94.2	ViT-B/16, 85M
SACL (neg.-used)	62.5	66.4	**82.1**	**80.2**	**82.1**	**96.8**	72.9	**77.8**	**92.9**	**79.2**	81.9	95.5	ViT-B/16, 85M
SACL (neg.-free)	**63.4**	**68.8**	81.8	78.9	80.1	96.1	**73.5**	76.8	92.5	78.2	**82.2**	**96.2**	ViT-B/16, 85M

iternorm with trace loss (INTL) [71]. Then, we also implement our method on the popular ViT-B/16 backbone and compare it with two more methods, including *Mugs* [89] and *DINO* [8]. We conduct comprehensive evaluations by recording the classification accuracy rates of all methods obtained with three popular protocols, including the fine-tuning linear softmax (i.e., the Top-1 score and Top-5 score of *linear probing*) and the k-NN classification (here $k = 8$). From Table IV, we can clearly observe that our method SACL successfully improves the SimCLR and BYOL (with ResNet-50 backbone) by at least 4% in different cases of batch size on the two datasets. Our method also consistently outperforms the best baseline methods MetAug, LORAC, and INTL on most scores, which demonstrates the effectiveness of our method. Similarly, based on the powerful ViT-B/16 feature encoder, our method consistently improves the baseline methods and outperforms the state-of-the-art methods in most cases. Since SACL is implemented on different baselines and different backbones, our method has good compatibility with existing CL algorithms on the image classification task.

2) Detection and Segmentation: Now, we would like to further investigate the transferability of our method on the objective detection and instance segmentation tasks. We first pretrain the model (with ResNet-50 backbone) on ImageNet-1K and then fine-tune the pretrained backbone on the new dataset. Specifically, we select COCO [42] as our target dataset and follow the common setting (as discussed in *MoCo-v3* [14]) to fine-tune *all layers* of the pretrained model over the *train2017* set while evaluating the performance on the *val2017* set. We employ *Faster R-CNN* [50] and *Mask R-CNN* [28] as our backbone for detection and segmentation, respectively. As listed in Table V, our SACL shows considerable improvement over MoCo-v3 and DINO on both two recognition tasks. This indicates that our method not only works well on classification-oriented tasks but also on more natural image-related recognition tasks.

TABLE V

PERFORMANCE OF ALL METHODS FOR TWO TRANSFER LEARNING TASKS: OBJECT DETECTION AND INSTANCE SEGMENTATION ON COCO DATASET

METHOD	Object Detection			Instance Segmentation		
	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{mk}	AP^{mk}_{50}	AP^{mk}_{75}
Supervised	38.2	59.1	41.5	35.4	56.5	38.1
BYOL [24]	39.9	60.2	43.3	36.5	58.4	39.1
SwAV [7]	40.3	61.5	44.4	36.3	58.7	39.4
MoCo-v2 [12]	37.6	57.9	40.8	35.3	55.9	37.9
MoCo-v3 [14]	39.9	61.2	43.2	36.5	58.1	38.8
DenseCL [68]	40.3	59.9	44.3	36.4	57.0	39.2
INTL [71]	40.7	60.9	43.7	35.4	57.3	37.6
DINO [8]	40.3	62.0	44.1	36.8	58.8	39.2
SACL (neg.-used)	**42.2**	**62.5**	44.8	**37.2**	**59.4**	**40.5**
SACL (neg.-free)	41.3	62.3	**45.2**	36.9	58.5	40.2

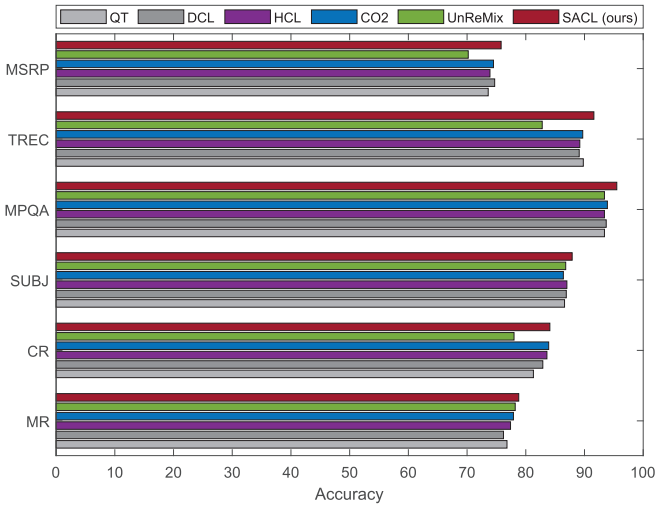


Fig. 7. Accuracy rates (%) of all methods on the BookCorpus dataset, including six text classification tasks.

In summary, our method has shown stable performance on the above three popular visual tasks. It is notable that although our method introduces new projection layers in the overall framework, the quantity of additional parameters (i.e., the tensor $\mathcal{P}$) can be actually ignored (as the final parameter listed in Table IV), and thus, our method is computation-friendly in practical uses.

C. Experiments on Text Data

In this section, we evaluate the performance of our method on the popular text classification task. Our experimental data include the *BookCorpus* benchmark dataset [36] and the semantic text similarity (STS) benchmark dataset (from STS12 to STS16) [1].

Sentence Embedding: For the BookCorpus dataset, which includes six subtasks movie review (MR) sentiment, product reviews (CR), subjectivity classification (SUBJ), opinion polarity (MPQA), question type classification (TREC), and paraphrase identification (MSRP), we follow the experimental settings in the baseline method quick-thought (QT) [45] to choose the neighboring sentences as positive pairs. Then, we further compare our SACL with DCL, HCL, *consistent contrast* (CO_2) [70], and uncertainty and representativeness mixing (UnReMix) [58], and the corresponding average classification accuracy rates are shown in Fig. 7. For the STS dataset, we follow the common practice in *SimCSE* [20] to use the pretrained checkpoints of *BERT* [18], and we train all methods on randomly sampled sentences from English Wikipedia. Then, we use the *Spearman correlation* to measure the correlation between the ranks of predicted scores and the ground truth. We also further compare our method with three more existing methods, including information minimization CL (InforMin-CL) [9], misCSE [37], and smoothed CL (SCL) [72].

For the six classification tasks in Fig. 7, our method improves the classification accuracy of baseline method QT for at least two percentage points on most classification benchmarks. With the default settings, our method can consistently obtain the better performance on all six tasks. Furthermore, as we can observe from Table VI, SACL obtains considerable improvements on the baseline method SimCSE. Meanwhile, our method can outperform the other three representative methods misCSE, InforMin-CL, and SCL in most cases. Our SACL also achieves the best average score in all compared methods. This clearly demonstrates that the similarity-agnostic property not only exists in the image data but also in the text data, and our method is a good solution to utilize such a property to enrich the self-supervisory information for model training.

TABLE VI
ACCURACY RATES (%) OF ALL METHODS ON STS DATASET, INCLUDING FIVE TASKS AND THE CORRESPONDING AVERAGE SCORES

METHOD	STS12	STS13	STS14	STS15	STS16	Aver.
SimCSE [20]	68.69	82.05	72.91	81.15	79.39	76.84
PCL [41]	72.74	83.36	76.05	83.07	79.26	78.90
InforMin-CL [9]	70.22	83.48	75.51	81.72	79.88	78.16
miCSE [37]	71.71	83.09	75.46	83.13	80.22	78.72
SCL [72]	**72.86**	**84.91**	76.79	84.35	81.74	80.13
SACL (ours)	72.80	84.12	**77.92**	**85.42**	**82.60**	**80.57**

VI. CONCLUSION AND FUTURE WORK

In this article, we first investigated the issue of agnostic similarity existing in traditional CL approaches. After that, we proposed a novel framework SACL, which generalizes the instance discrimination strategy of conventional CL to a new vector-valued loss form. We built different projection layers to capture diverse potential similarities based on a gradually sparse regularization so that we can successfully consider both the similar and dissimilar patterns between pairwise instances. To the best of our knowledge, this is the first work in CL that proposes agnostic similarity and simultaneously considers both the similarity and dissimilarity between each pair of instances. We conducted intensive theoretical analyses to guarantee the effectiveness of our method. Comparison experiments on real-world datasets across multiple domains indicated that our learning algorithm acquires more reliable feature representations than state-of-the-art methods.

However, there are also several limitations in the proposed work. A potential risk is that the dynamic self-supervisory information, which depends on alterable pairwise distances, might degrade when the quality of negative samples is poor, thereby diminishing the richness of the self-supervision. Moreover, although experimental results on standard benchmark datasets across images, text, and graphs are promising, the approach has not been thoroughly evaluated in more difficult or domain-specific tasks, such as cross-domain adaptation and low-resource language scenarios, leaving open questions about its generalizability and robustness in more complex real-world settings. As we used hyperparameters to control the sparsity of each projection layer, exploring the automatic determination of the sparsity parameters would be interesting for future work.

REFERENCES

[1] E. Agirre et al., "SemEval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation," in *Proc. Int. Workshop Semantic Eval.*, 2016, pp. 1–15.

[2] S. Ardeshir and N. Azizan, "Uncertainty in contrastive learning: On the predictability of downstream performance," 2022, *arXiv:2207.09336*.

[3] Y. Bai et al., "MSR: Making self-supervised learning robust to aggressive augmentations," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2022, pp. 1–14.

[4] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[5] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[6] Y. Cao, L. Feng, Y. Xu, B. An, G. Niu, and M. Sugiyama, "Learning from similarity-confidence data," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2021, pp. 1272–1282.

[7] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2020, pp. 1401–1413.

[8] M. Caron et al., "Emerging properties in self-supervised vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9630–9640.

[9] S. Chen, J. Zhou, Y. Sun, and L. He, "An information minimization based contrastive learning model for unsupervised sentence embeddings learning," in *Proc. Int. Conf. Comput. Linguistics (COLING)*, 2022, pp. 1–11.

[10] S. Chen, G. Niu, C. Gong, J. Li, J. Yang, and M. Sugiyama, "Large-margin contrastive learning with distance polarization regularizer," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2021, pp. 1673–1683.

[11] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.

[12] X. Chen, H. Fan, R. Girshick, and K. He, "Improved baselines with momentum contrastive learning," 2020, *arXiv:2003.04297*.

[13] X. Chen and K. He, "Exploring simple Siamese representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 15745–15753.

[14] X. Chen, S. Xie, and K. He, "An empirical study of training self-supervised vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9620–9629.

[15] X. Chen et al., "Self-PU: Self boosted and calibrated positive-unlabeled training," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2020, pp. 1510–1519.

[16] C.-Y. Chuang, J. W. Robinson, Y.-C. Lin, A. Torralba, and S. Jegelka, "Debiased contrastive learning," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, 2020, pp. 1–11.

[17] J. Denize, J. Rabarisoa, A. Orcesi, R. Hérault, and S. Canu, "Similarity contrastive estimation for self-supervised soft contrastive learning," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2023, pp. 2705–2715.

[18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[19] A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, and T. Brox, "Discriminative unsupervised feature learning with convolutional neural networks," in *Proc. NeurIPS*, vol. 27, 2014, pp. 766–774.

[20] T. Gao, X. Yao, and D. Chen, "SimCSE: Simple contrastive learning of sentence embeddings," 2021, *arXiv:2104.08821*.

[21] W. Ge, W. Huang, D. Dong, and M. R. Scott, "Deep metric learning with hierarchical triplet loss," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Jan. 2018, pp. 272–288.

[22] X. Gong, D. Yuan, and W. Bao, "Online metric learning for multi-label classification," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2020, pp. 1–8.

[23] I. Goodfellow and Y. Bengio, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[24] J.-B. Grill et al., "Bootstrap your own latent—A new approach to self-supervised learning," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, 2020, pp. 21271–21284.

[25] K. Han et al., "A survey on vision transformer," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 87–110, Jan. 2023.

[26] D. He et al., "Analyzing heterogeneous networks with missing attributes by unsupervised contrastive learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 4, pp. 4438–4450, Apr. 2024.

[27] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9729–9738.

[28] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2961–2969.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[30] Q. Hu, X. Wang, W. Hu, and G.-J. Qi, "AdCo: Adversarial contrast for efficient learning of unsupervised representations from self-trained negative adversaries," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 1074–1083.

[31] Z. Jiang, T. Chen, T. Chen, and Z. Wang, "Robust pre-training by adversarial contrastive learning," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, 2020, pp. 1–12.

[32] Z. Jiang, T. Chen, B. Mortazavi, and Z. Wang, "Self-damaging contrastive learning," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2021, pp. 1–13.

[33] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 11, pp. 4037–4058, Nov. 2021.

[34] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 26, 2013, pp. 315–323.

[35] Y. Kalantidis, M. B. Sariyildiz, N. Pion, P. Weinzaepfel, and D. Larlus, "Hard negative mixing for contrastive learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 1–12.

[36] R. Kiros et al., "Skip-thought vectors," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 28, 2015, pp. 3294–3302.

[37] T. Klein and M. Nabi, "MiCSE: Mutual information contrastive learning for low-shot sentence embeddings," 2022, *arXiv:2211.04928*.

[38] X. Kou, C. Xu, X. Yang, and C. Deng, "Attention-guided contrastive hashing for long-tailed image retrieval," in *Proc. 31st Int. Joint Conf. Artif. Intell.*, Jul. 2022, pp. 1017–1023.

[39] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Tech. Rep., Apr. 2009.

[40] J. Li, W. Qiang, C. Zheng, B. Su, and H. Xiong, "MetAug: Contrastive learning via meta feature augmentation," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2022, pp. 12964–12978.

[41] J. Li, P. Zhou, C. Xiong, and S. C. H. Hoi, "Prototypical contrastive learning of unsupervised representations," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021, pp. 1–16.

[42] T.-Y. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Jan. 2014, pp. 740–755.

[43] R. Liu, Z. Jiang, S. Yang, and X. Fan, "Twin adversarial contrastive learning for underwater image enhancement and beyond," *IEEE Trans. Image Process.*, vol. 31, pp. 4922–4936, 2022.

[44] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "DeepFashion: Powering robust clothes recognition and retrieval with rich annotations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1096–1104.

[45] L. Logeswaran and H. Lee, "An efficient framework for learning sentence representations," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–16.

[46] R. T. Marler and J. S. Arora, "Survey of multi-objective optimization methods for engineering," *Struct. Multidisciplinary Optim.*, vol. 26, no. 6, pp. 369–395, 2004.

[47] B. Mustafa, C. Riquelme, J. Puigcerver, R. Jenatton, and N. Houlsby, "Multimodal contrastive learning with limoe: The language-image mixture of experts," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2022, pp. 1–13.

[48] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 32, 2019, pp. 1–12.

[49] N. Pielawski et al., "CoMIR: Contrastive multimodal image representation for registration," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 33, 2020, pp. 1–12.

[50] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 28, 2015, pp. 1–9.

[51] J. Robinson, C. Y. Chuang, S. Sra, and S. Jegelka, "Contrastive learning with hard negative samples," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021, pp. 1–28.

[52] O. Russakovsky et al., "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[53] N. Saunshi, O. Plevrakis, S. Arora, M. Khodak, and H. Khandeparkar, "A theoretical analysis of contrastive unsupervised representation learning," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 5628–5637.

[54] O. Sener and V. Koltun, "Multi-task learning as multi-objective optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–12.

[55] K. Sohn, "Improved deep metric learning with multi-class N-pair loss objective," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 29, 2016, pp. 1857–1865.

[56] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2014, pp. 1–9.

[57] Y. Sun, H. Xue, R. Song, B. Liu, H. Yang, and J. Fu, "Long-form video-language pre-training with multimodal temporal contrastive learning," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2022, pp. 1–14.

[58] A. Tabassum, M. Wahed, H. Eldardiry, and I. Lourentzou, "Hard negative sampling strategies for contrastive representation learning," 2022, *arXiv:2206.01197*.

[59] R. Tan, M. Vasileva, K. Saenko, and B. Plummer, "Learning similarity conditions without explicit supervision," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 10372–10381.

[60] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 1–18.

[61] Y. Tian, C. Sun, B. Poole, D. Krishnan, C. Schmid, and P. Isola, "What makes for good views for contrastive learning?," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, 2020, pp. 1–13.

[62] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.

[63] S. Wan et al., "Boosting graph contrastive learning via adaptive sampling," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 11, pp. 15971–15983, Nov. 2024.

[64] N. Wang, P. Feng, Z. Ge, Y. Zhou, B. Zhou, and Z. Wang, "Adversarial spatiotemporal contrastive learning for electrocardiogram signals," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 10, pp. 13845–13859, Oct. 2024.

[65] T. Wang, J. Lu, Z. Lai, J. Wen, and H. Kong, "Uncertainty-guided pixel contrastive learning for semi-supervised medical image segmentation," in *Proc. 31st Int. Joint Conf. Artif. Intell.*, Jul. 2022, pp. 1444–1450.

[66] X. Wang, Y. Huang, D. Zeng, and G.-J. Qi, "CaCo: Both positive and negative samples are directly learnable via cooperative-adversarial contrastive learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 9, pp. 10718–10730, Sep. 2023.

[67] X. Wang and G.-J. Qi, "Contrastive learning with stronger augmentations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 5, pp. 5549–5560, May 2023.

[68] X. Wang, R. Zhang, C. Shen, T. Kong, and L. Li, "Dense contrastive learning for self-supervised visual pre-training," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 3023–3032.

[69] Y. Wang et al., "A low rank promoting prior for unsupervised contrastive learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 2667–2681, Mar. 2023.

[70] C. Wei, H. Wang, W. Shen, and A. Yuille, "Co2: Consistent contrast for unsupervised visual representation learning," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021, pp. 1–12.

[71] X. Weng et al., "Modulate your spectrum in self-supervised learning," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2024, pp. 1–28.

[72] X. Wu, C. Gao, Y. Su, J. Han, Z. Wang, and S. Hu, "Smoothed contrastive learning for unsupervised sentence embedding," in *Proc. 29th Int. Conf. Comput. Linguistics (ACL)*, 2022, pp. 1–5.

[73] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3733–3742.

[74] E. Xing, M. Jordan, S. J. Russell, and A. Ng, "Distance metric learning with application to clustering with side-information," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 15, 2002, pp. 1–8.

[75] C. Xu, Z. Chai, Z. Xu, C. Yuan, Y. Fan, and J. Wang, "HyP^2 loss: Beyond hypersphere metric space for multi-label image retrieval," in *Proc. 30th ACM Int. Conf. Multimedia*, 2022, pp. 3173–3184.

[76] J. Xu, L. Luo, C. Deng, and H. Huang, "Bilevel distance metric learning for robust image recognition," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2018, pp. 1–10.

[77] J. Xu, L. Luo, C. Deng, and H. Huang, "Multi-level metric learning via smoothed Wasserstein distance," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 2919–2925.

[78] X. Xu, C. Deng, Y. Xie, and S. Ji, "Group contrastive self-supervised learning on graphs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 3169–3180, Mar. 2023.

[79] X. Xu, Z. Wang, C. Deng, H. Yuan, and S. Ji, "Towards improved and interpretable deep metric learning via attentive grouping," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 1189–1200, Jan. 2023.

[80] X. Xu, Y. Yang, C. Deng, and F. Zheng, "Deep asymmetric metric learning via rich relationship mining," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4071–4080.

[81] J. Yan, L. Luo, C. Deng, and H. Huang, "Unsupervised hyperbolic metric learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 12465–12474.

[82] J. Yan, E. Yang, C. Deng, and H. Huang, "MetricFormer: A unified perspective of correlation exploring in similarity learning," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2022, pp. 1–14.

[83] X. Yang, X. Hu, S. Zhou, X. Liu, and E. Zhu, "Interpolation-based contrastive learning for few-label semi-supervised learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 2, pp. 2054–2065, Feb. 2024.

[84] Y. You, T. Chen, Y. Shen, and Z. Wang, "Graph contrastive learning automated," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Jul. 2021, pp. 12121–12132.

[85] M. Zheng et al., "Weakly supervised contrastive learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 10022–10031.

[86] M. Zheng, S. You, L. Huang, F. Wang, C. Qian, and C. Xu, "SimMatch: Semi-supervised learning with similarity matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 14451–14461.

[87] M. Zheng et al., "ReSSL: Relational self-supervised learning with weak augmentation," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2021, pp. 1–13.

[88] H. Zhong, C. Chen, Z. Jin, and X.-S. Hua, "Deep robust clustering by contrastive learning," 2020, *arXiv:2008.03030*.

[89] P. Zhou, Y. Zhou, C. Si, W. Yu, T. Khim Ng, and S. Yan, "Mugs: A multi-granular self-supervised learning framework," 2022, *arXiv:2203.14415*.

[90] P. Zhu, R. Qi, Q. Hu, Q. Wang, C. Zhang, and L. Yang, "Beyond similar and dissimilar relations: A kernel regression formulation for metric learning," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 3242–3248.

[91] M. Zinkevich, M. Weimer, L. Li, and A. J. Smola, "Parallelized stochastic gradient descent," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 4, 2010, pp. 1–9.

Shuo Chen received the Ph.D. degree from Nanjing University of Science and Technology, Nanjing, China, in 2020.

He was a CSC Visiting Student at the University of Pittsburgh, Pittsburgh, PA, USA, from 2018 to 2019. He was a Post-Doctoral Researcher and a Research Scientist at the RIKEN National Science Institute, Wako, Japan, from 2020 to 2024. He is currently an Associate Professor with the School of Intelligence Science and Technology, Nanjing University, Nanjing. He has published more than 50 technical papers at top-tier conferences such as NeurIPS, ICML, ICLR, and CVPR, and prominent journals such as IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (TPAMI), IEEE TRANSACTIONS ON IMAGE PROCESSING (TIP), and IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (TNNLS). His research interests include machine learning and pattern recognition.

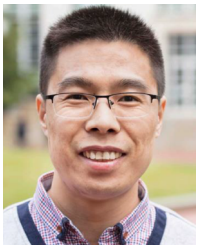
Dr. Chen won the "Excellent Achievement Award" of RIKEN, the "Excellent Doctoral Dissertation Award" of Chinese Institute of Electronics, and the "Excellent Doctoral Dissertation Nomination" of Chinese Association for Artificial Intelligence. He has served as the Area Chair for NeurIPS, ICML, ICLR, CVPR, and AAAI over ten times.

Chen Gong (Senior Member, IEEE) received the Ph.D. degrees from Shanghai Jiao Tong University, Shanghai, China, and the University of Technology Sydney, Ultimo, NSW, Australia, in 2016 and 2017, respectively.

He is currently a Professor with Nanjing University of Science and Technology, Nanjing, China. He has published more than 100 technical papers at prominent journals and conferences such as JMLR, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (TPAMI), IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (TNNLS), IEEE TRANSACTIONS ON IMAGE PROCESSING (TIP), ACM T-IST, CVPR, ICML, NeurIPS, AAAI, IJCAI, and ICDM. His research interests mainly include machine learning and data mining.

Dr. Gong won the "Excellent Doctorial Dissertation Award" of Chinese Association for Artificial Intelligence, "Young Elite Scientists Sponsorship Program" of China Association for Science and Technology, the "Wu Wen-Jun AI Excellent Youth Scholar Award," and the Science Fund for Distinguished Young Scholars of Jiangsu Province. He was also selected as the "Global Top Chinese Young Scholars in AI" released by Baidu. He also serves as a reviewer for more than 20 international journals, such as JMLR, AIJ, IEEE T-PAMI, IJCV, IEEE TNNLS, and IEEE TIP, and also an (S)PC Member of several top-tier conferences, such as ICML, NeurIPS, ICLR, CVPR, ICCV, AAAI, IJCAI, and ICDM.

He has served as an SPC/PC Member for CVPR, ICCV, ICML, NeurIPS, ICLR, and AAAI and a reviewer for many international journals, such as IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (TNNLS), IEEE TRANSACTIONS ON IMAGE PROCESSING (TIP), IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING (TKDE), IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY (TIFS), and IEEE TRANSACTIONS ON CYBERNETICS (TCYB).



Jun Li (Member, IEEE) received the Ph.D. degree in pattern recognition and intelligence systems from Nanjing University of Science and Technology, Nanjing, China, in 2015.

From October 2012 to July 2013, he was a Visiting Student at the Department of Statistics, Rutgers University, Piscataway, NJ, USA. From December 2015 to October 2018, he was a Post-Doctoral Associate with the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA, USA. From November 2018 to October 2019, he was a Post-Doctoral Associate with the Institute of Medical Engineering and Science, Massachusetts Institute of Technology, Cambridge, MA, USA. He is currently a Professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology, since 2019. His research interests are computer vision and creative machine learning.

Jian Yang (Member, IEEE) received the Ph.D. degree in pattern recognition and intelligence systems from Nanjing University of Science and Technology (NUST), Nanjing, China, in 2002.

From 2003 to 2007, he was a Post-Doctoral Fellow at the University of Zaragoza, Zaragoza, Spain; The Hong Kong Polytechnic University, Hong Kong; and New Jersey Institute of Technology, Newark, NJ, USA. Since 2007, he has been a Professor with the School of Computer Science and Technology, NUST. His articles have been cited over 50 000 times in Google Scholar. His research interests include pattern recognition and computer vision.

Dr. Yang is a fellow of IAPR. He is/was an Associate Editor of *Pattern Recognition*, *Pattern Recognition Letters*, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, and *Neurocomputing*.