Semi-Supervised Contrastive Learning With Similarity Co-Calibration

Yuhang Zhang, Xiaopeng Zhang, Jie Li, Senior Member, IEEE, Robert C. Qiu, Fellow, IEEE, Haohang Xu, and Qi Tian, Fellow, IEEE

Abstract—Semi-supervised learning acts as an effective way to leverage massive unlabeled data. In this paper, we propose a novel training strategy, termed Semi-Supervised Contrastive Learning (SsCL), which combines the well-known contrastive loss in self-supervised learning with the cross entropy loss in semi-supervised learning, and jointly optimizes the two objectives in an end-to-end way. The highlight is that different from self-training based semi-supervised learning that conducts prediction and retraining over the same model weights, SsCL interchanges the predictions over the unlabeled data between the two branches, and thus formulates a co-calibration procedure, which we find is beneficial for better prediction and avoids being trapped in local minimum. Towards this goal, the contrastive loss branch models pairwise similarities among samples, using the pseudo labels generated from the cross entropy branch, and in turn calibrates the prediction distribution of the cross entropy branch with the contrastive similarity. We show that SsCL produces more discriminative representation and is beneficial to semi-supervised learning. Notably, on ImageNet with ResNet50 as the backbone, SsCL achieves 60.2% and 72.1% top-1 accuracy with 1% and 10% labeled samples respectively, which significantly outperforms the baseline, and is better than previous semi-supervised and self-supervised methods.

Index Terms—Semi-supervised learning, contrastive learning, similarity co-calibration.

I. INTRODUCTION

Semi-Supervised learning (SSL) has attracted more attention over the past years, as it allows for training network with limited labeled data to improve the model performance with the availability of large scale unlabeled data. SSL plays a great role in different application scenarios and data types, including image data [1]–[6], time-series data [7], [8] and multi-modal data [9], [10]. An effective way to use unlabeled data is automatic annotate unlabeled data [11]–[13], while another general methods are based on pseudo-labeling and consistency regularization [1], [3], [14]. The key idea is to train the model with limited labeled data and predict the pseudo labels for the unlabeled samples. In this pipeline, the images with high-confidence prediction are selected for retraining, using either one-hot hard labels or the predicted distribution of soft labels. In order to avoid homogenization in SSL, i.e., predicting and retraining with the same samples, data augmentation is widely used, and has been demonstrated to be particularly effective when combined with consistency regularization, which penalizes the discrepancy between different augmentations of the same samples. In SSL, the network is usually optimized by minimizing the cross entropy loss between the prediction and the target distribution as fully supervised learning. However, as pointed out by [15], [16], such cross entropy loss is not robust to noisy labels, which is inevitable in semi-supervised learning. This is especially true for large scale dataset such as ImageNet, where limited labeled samples cannot represent the specific class properly, and cross entropy based semi-supervised methods [2], [4], [6], [14] suffer limited performance gain.

On the other hand, recent advances in self-supervised learning have demonstrated powerful performance in semi-supervised learning scenarios, especially with the achievements of contrastive learning [17]. In this setting, the model is first pretrained with large scale unlabeled data, and then fine-tuned with few shot labeled samples. In the pretraining stage, a typical solution is to...
treat each image as well as its augmentations as a separate class, and the features among different transformations of an image are pulled closer together, while all other instances are treated as negatives and pushed away. However, due to the lack of labels, the self-supervised pretraining stage is task agnostic, which is not optimal for specific tasks.

In this paper, we propose a novel training strategy that incorporates contrastive loss into semi-supervised learning, and jointly optimize the two objectives in an end-to-end way. The motivation is that it should be better for specific task when we incorporate class specific priors into the contrastive pretraining stage, and hope that the two different feature learning strategies would benefit each other for better representation. Towards this goal, we introduce a co-calibration mechanism to interchange predictions between the two branches: one branch is based on cross entropy loss and the other one is based on contrastive loss. In particular, the pseudo labels generated from the cross entropy loss term are used to search for nearest neighbors for contrastive learning, while the similarity embedding learned by the contrastive term is used in turn to adjust the prediction in the pseudo labeling branch. In this way, the information between the two branches is interflowed, which is complementary with each other and thus avoids being trapped in the local minimum course by self-training.

In order to facilitate co-calibration between the two branches, we extend the contrastive loss [17] to allow for multiple positive samples during each forward propagation, and pulling samples that are similar to the specified class prototype for discriminative representation. In order to tackle the noisy samples especially at the initial training stage, we propose a self-paced learning strategy which adaptively adjusts the loss that the mined samples belonging to the assigned class. In this way, the model is gradually evolving as the similarity computation becomes more accurate. Furthermore, to avoid the biased representation of each class with limited labeled data, we introduce a data mix strategy to enrich samples of each class. Specifically, we periodically select similar samples during the semi-supervised learning process, and apply data mixing [18] operation over the randomly selected two samples from the ground truth and similar sample pools, respectively. In this way, we are able to enrich the class specific samples in a robust and smooth way.

Integrating contrastive loss into semi-supervised learning produces a much more powerful framework, and experiments demonstrate that the enriched representation is beneficial for data mining and results in improved representation. Notably, using the standard ResNet-50 as the backbone, we can achieve 60.2% and 72.1% top-1 accuracy on the large scale ImageNet dataset with only 1% and 10% labeled samples respectively, which consistently surpasses the semi-supervised and self-supervised methods.

II. RELATED WORK

Our approach is related with recent advances covering semi-supervised learning and self-supervised learning. We briefly review related works and clarify the differences between them and our method.

A. Semi-supervised Learning

Semi-supervised Learning makes uses of few shot labeled data in conjunction with a large number of unlabeled data [2], [5], [14], [19]–[26]. Pseudo labeling is a widely used method in early works, which uses the predictions of the model to label images and retrain the model with these predictions [1], [3]. After that, exponential moving average of model parameters is found to be useful to further improve the performance of model [23]. In the follow-up development, [2] uses the distribution of ground truth to align the pseudo labels. Then [14] combines consistency regularization and pseudo-labeling to align the predictions between weakly and strongly augmented unlabeled images. S^4 L [6] is the first unified work that combines self-supervised learning with semi-supervised learning, but it only trains the model by directly combining previous unsupervised loss and supervised loss and no changes have been made on them. As to our approach, we not only use pseudo labeling method to build a part of our loss function, but also insert the pseudo label into our multi-positive contrastive learning, and contrastive learning representations will also help the model obtain more accurate pseudo labels.

B. Contrastive Learning

Contrastive Learning in recent works mainly benefits from instance discrimination [27], which regards each image and its augmentations as one separate class and others as negatives [17], [28]–[34]. The researchers used different ways to maintain a training queue, for example, [27] used a memory bank to store the pre-computed representations from which positive examples are retrieved given a query. Based on this, [17] used a momentum updating mechanism to maintain a long queue of negative examples for contrastive learning, while [28] used a large batch of training data to produce enough negative samples. [34] move away negative samples, through predicting previous version of the sample itself can also achieve competitive results. These works prove that contrastive learning reaches better performance on learning data characteristics. Previous contrastive based semi-supervised learning works are almost two-stage ones, i.e., using contrastive learning to pretrain a backbone model and then using few shot labeled data to fine-tune it. In contrast, our method trains the model in an end-to-end way, which is able to make better use of the advantages of the features learned by different loss functions and class specific priors.

III. METHOD

We first give an overview of the proposed Semi-supervised Contrastive Learning framework. SsCL includes a combination of two well-known approaches, i.e., the pseudo labeling strategy with cross entropy loss, and the instance discrimination with contrastive loss. The highlight is that we jointly optimize the two losses with a shared backbone in an end-to-end way, and importantly, the pseudo labels between the two branches are calibrated in a co-training mechanism, which we find is beneficial for better prediction over the unlabeled data.
Specifically, denote $\mathcal{X} = \{x_b, y_b : b \in (1, \ldots, B)\}$ as a batch of $B$ labeled samples, where $y_b$ denotes labels, and denote $\mathcal{U} = \{u_b : b \in (1, \ldots, \mu B)\}$ as a batch of $\mu B$ unlabeled samples, where $\mu$ determines the relative size of $\mathcal{X}$ and $\mathcal{U}$. $DA_a$ and $DA_b$ represent the different data augmentation conducted over an image. Overall, SsCL targets at optimizing three losses: 1) the supervised loss $L_{sup}$ optimized over the labeled data; 2) the pseudo labeling loss $L_{pl}$ penalized on the unlabeled data and 3) the contrastive loss $L_{ctr}$ that enforces pairwise similarity among neighborhood samples. The whole framework is shown in Fig. 2, and we omit the supervised loss term for brevity.

The supervised loss is simply conducted over the labeled data $x_{\text{b}}$ with cross-entropy minimization, using the ground truth labels $y_{\text{b}}$:

$$L_{sup} = \frac{1}{B} \sum_{b=1}^{B} H(y_{\text{b}}, p(y|DA(x_{\text{b}})))$$  \hspace{1cm} (1)

Similarly, the pseudo labeling loss is penalized over the unlabeled data $u_{\text{b}}$, using the pseudo labels $\hat{p}_{\text{b}}$:

$$L_{pl} = \frac{1}{\mu B} \sum_{b=1}^{\mu B} 1(\max \hat{p}_{\text{b}} \geq \tau) H(\hat{p}_{\text{b}}, p(y|DA(u_{\text{b}})))$$  \hspace{1cm} (2)

where $\hat{p}_{\text{b}}$ denotes the model’s prediction on the unlabeled samples, which is obtained by calibrating the pseudo labels $p_{\text{b}}$ predicted from the cross entropy loss term with similarity distribution $\tilde{p}_{\text{b}}$ from the contrastive loss term, and it will be elaborated in the following Section III. 1 is an indicator function and we only retain $p_{\text{b}}$ whose largest class probability is above a certain threshold $\tau$ for optimization.

For contrastive loss, different from [17], we adjust the contrastive loss term to allow for multiple positive samples during each forward propagation so that similar images that belong to the corresponding class prototype are pulled together for more discriminative representation. We defer the definition of contrastive loss term in the following section. The overall losses can be formulated as:

$$L = L_{sup} + \lambda_{pl} L_{pl} + \lambda_{ctr} L_{ctr},$$  \hspace{1cm} (3)

where $\lambda_{pl}$ and $\lambda_{ctr}$ are the balancing factors that control the weights of the two losses. In the following sections, we describe the terms $L_{pl}$ and $L_{ctr}$ in detail and elaborate the similarity co-calibration procedure between the two branches.

A. Pseudo Label Calibration With Contrastive Similarity

For conventional semi-supervised learning with cross entropy loss, the pseudo label $p_{\text{b}}$ of an unlabeled sample $u_{\text{b}}$ is simply derived by predicting distribution on $DA_a(u_{\text{b}})$ using the current model, and enforces the cross-entropy loss against the model’s output for $DA_b(u_{\text{b}})$ in the following training procedure. However, the pseudo labeling and re-training is conducted over the same network, which suffers from model homogenization issue and is easy to be trapped in local minimum. In this section, we propose a pseudo label calibration strategy that refines the prediction $p_{\text{b}}$ via the similarity distribution $p_{\text{b}}$ from the contrastive loss term.

a) Similarity Distribution: The similarity distribution $p_{\text{b}}$ over the classes is obtained as follows. Given a list of $n$ few shot labeled samples $X_c = \{x_{i}\}_{i=1}^{n}$ for class $c$, we compute the feature representation $v_{i}$ of $x_{i}$ with current encoder $E_q$ where $v_{i} = E_q(x_{i})$. The feature representation of class $c$ is simply obtained by averaging the features, i.e., $\tilde{v}_{c} = \sum v_{i}/nZ$, where $Z$ is a normalizing constant value. Given an unlabeled sample $x'$ with normalized feature $\tilde{v}'$, we can get its cosine similarity distribution $p_{\text{b}} = \{s_{1}, s_{2}, \ldots, s_{K}\}$ with $C$ predefined prototypes, where $s_{k}$ denotes the similarity between $(\tilde{v}', \tilde{v}_{c})$.

Inspired by [2], we introduce a form of fairness prediction $\hat{p}_{\text{b}}$ by maintaining a running average of distribution $p_{\text{b}}$ (average...
vector of $p_b$ over the last $t$ batches, here we simply set $t = 128$) over the course of training. Overall, $p_b$ can be treated as a global adjustment among the scores for each class, and thus avoid biased prediction $p_b$ over a single image $u_b$. Then the calibration is simply conducted by scaling $p_b$ with $\hat{p}_b$ and then re-normalizing to form a valid probability distribution $\hat{p}_b$. We use $\hat{p}_b$ as calibrated pseudo labels for optimization following (2).

b) Prototype Refinement: The similarity distribution $p_b$ requires a predefined class specific prototype generated by the few labeled samples, and each prototype $\tilde{\mathbf{v}}_c$ is simply computed using the available ground truth labels belonging to class $c$. However, this representation is easily biased to the few shot samples, which is insufficient to model the real distribution. To solve this issue, we propose a prototype mixture strategy to refine the feature representation of each class in a robust way. The new samples are generated via a simple mixup operation, conducted between the mined samples and those with ground truth labels. In this way, based on the continuity of feature space, we can ensure that the generated samples do not drift away too much from the real distribution of that class, while enriching the feature representation in a robust way. Specifically, given the few shot labeled sample set $\mathbf{X}_c$ for class $c$ and a list of nearest neighborhood unlabeled samples $\Omega_c$ that belong to class $c$, we randomly select a ground truth sample $x_c \in \mathbf{X}_c$, and a nearest sample $x_N \in \Omega_c$, and mix them to generate a new sample $\hat{x}$:

$$\hat{x} = \lambda \cdot x_c + (1 - \lambda) \cdot x_N,$$

where $\lambda$ is the combination factor and is sampled from beta distribution $\text{Beta}(\alpha, \alpha)$ with parameter $\alpha$. In our implementation, we simply set $\alpha = 1$, and $\hat{x}$ will be added to $\mathbf{X}_c$ only used for calculating prototype for class $c$. The purpose of this step is to smooth the noise caused during the sample selection process. We visualize the benefits of this step in Fig. 3. Benefiting from the mixed samples, the feature representation is refined to better represent the class prototype. We would validate its effectiveness in the experimental section.

B. Contrastive Learning With Pseudo Label Propagation

In contrastive learning, the positive samples are simply constrained within a single image with different data transformations, and all other images are treated as negatives and pushed away. In this section, we incorporate the few shot labeled priors into contrastive learning, and target at pursuing more discriminative representation with the help of the mined samples from the pseudo labeling branch. Specifically, we extend the original contrastive learning loss in [17] and pull the mined samples closer together with the predefined prototypes. Benefiting from the class specific priors, the feature representation is more discriminative, which is beneficial for semi-supervised learning that encourages class separability and better representation.

a) Loss Function: In order to efficiently leverage the unlabeled samples in a robust way, inspired by [35], we adjust the original contrastive loss from [17] in three aspects. First, we extend it to allow for multiple positives for each forward propagation; second, we introduce a margin value $m$ which enforces class separability for better generalization, which we find is robust for data mining; third, considering that the mined samples may not be reliable, especially during the initial training stage, we introduce a softer factor $\alpha_p$ for each mined positive sample, and adaptively penalize the discrepancy according to its similarity to the corresponding class. For ease of expression, we make a deformation of original contrastive loss by using $s_p$ to represent the similarity of positive samples while $s_n$ represents the similarity of negative samples, and denote $\gamma = 1/\tau$. After introducing margin $m$ and soften factor $\alpha_p$, the loss $\mathcal{L}_{ctr}$ can be reformulated as:

$$\mathcal{L}_{ctr} = -\log \left( \sum_{k=1}^{k^*} \exp(\gamma(\alpha_p s_p - m)) \over \sum_{k=1}^{k^*} \exp(\gamma(\alpha_p s_p - m)) + \sum_{k=1}^{k^*} \exp(\gamma s_n) \right)$$

$$= -\log \left( \sum_{k=1}^{k^*} \exp(\gamma(\alpha_p s_p)) \over \sum_{k=1}^{k^*} \exp(\gamma s_n) + \sum_{k=1}^{k^*} \exp(\gamma s_n + m) \right)$$

$$= \log \left( 1 + \sum_{k=1}^{k^*} \exp(\gamma(s_n + m)) \over \sum_{k=1}^{k^*} \exp(-\gamma \alpha_p s_p) \right).$$

(5)

Designed with the above three properties, the loss is well suited for semi-supervised learning. In the following parts, we would analyze its advantages in detail.

b) Positive Sample Selection: For this branch, the positive samples are simply selected from the pseudo label loss branch. Specifically, given an unlabeled sample $u_b$ with pseudo labels $p_b$, which is predicted by the $fc$ layers of the cross entropy term. We assign it to class $c$ if $\text{argmax}_c p_b = c$, denoting that this sample is most similar with class $c$. Then we pull $u_b$ together with those samples that have ground truth label $c$. In this way, the unlabeled samples are pulled towards the specific instances corresponding to that class, and the representation is more compact. In practice, all the positive samples are maintained in the key encoder $E_k$. Benefit from the moment contrast mechanism in Moco [17], the
encoder $E_k$ is asynchronously updated and add more positive keys doesn’t increase the computation complexity.

c) Self-paced Weighting Mechanism: A critical issue when using model’s prediction on unlabeled data for positive sample selection is that at the initial training stage, the model is under-fitted, and the accuracy of the pseudo labels is very low. As a result, the incorrect positive samples would be assigned to a class, causing the model to update in the wrong direction, which is harmful for generalization. To solve this issue, we propose a self-paced learning method to enhance the optimization flexibility and it is robust to the noisy samples. This is achieved by assigning each selected positive sample a weight factor $\alpha_p$, which is determined by the similarity between the current sample and the predefined most similar class prototype. It is equivalent to adding a penalty term to the loss function, and with the increased accuracy of positive sample selection, the penalty becomes larger, and in turn enforces the compact representation.

4) Analysis of Our Loss Function: We now step deep into the optimization objective Eq. (5) to better understand its properties. Towards this goal, we first make some approximate transformations using the following formulations:

$$L_{LSE}(x; \gamma) = \frac{1}{\gamma} \log \sum_i \exp(\gamma x_i) \approx \max(x)$$ \hspace{1cm} (6)

$$L_{-LSE}(x; \gamma) = -\frac{1}{\gamma} \log \sum_i \exp(-\gamma x_i) \approx \min(x)$$ \hspace{1cm} (7)

$$L_{Softplus} = \log(1 + \exp(x)) \approx \max(x, 0)$$ \hspace{1cm} (8)

From above formulas, we can transform the (5) to:

$$L_{ctr} = \log \left(1 + \sum_{k^-} \exp(\gamma s_n + m) \sum_{k^+} \exp(-\gamma \alpha_p s_p)\right)$$

$$\approx \max \left(\log \sum_{k^-} \exp(\gamma s_n + m) \sum_{k^+} \exp(-\gamma \alpha_p s_p), 0\right)$$

$$= \max \left(\log \sum_{k^-} \exp(\gamma s_n + m) \log \sum_{k^+} \exp(-\gamma \alpha_p s_p), 0\right)$$

$$= \max(\gamma (L_{LSE}(s_n) + m - L_{-LSE}(\alpha_p s_p)), 0)$$

$$\approx \max(\gamma (\max(s_n) - \min(\alpha_p s_p) + m), 0)$$ \hspace{1cm} (9)

From the above equations, we have the following observations: (1) According to (9), the objective is to optimize the upper bound of positive pairs and lower bound of negative pairs, and introducing of $\alpha_p$ makes this process more flexible; (2) Based on the properties of (6) and (7), whose gradient is the $\text{softmax}$ function, the positive and negative pairs obtain equal gradient, regardless of the number of positive and negative sample pairs. This gradient equilibrium ensures the success of our multi-positive sample training.

IV. Experiment

In this section, we evaluate our method on several standard image classification benchmarks, including CIFAR-10 [36] and ImageNet [37] when few shots labels are available. We also evaluate our representations by transferring to other classification benchmarks and downstream tasks.

For the contrastive branch, we use two encoders following [17], one for training (named as $E_q$) and the other one for momentum updating (named as $E_k$) to produce negative keys. For prediction co-calibration, the contrastive similarity is computed via features obtained by $E_q$ and the prediction on unlabeled data is achieved by the $fc$ head. For efficiency, we conduct these operations offline, and update the pseudo labels (for selecting positive samples) and prototype for each category every 5 epochs during the training process. We find that the results are robust as the updating frequency ranges from 1 to 20. Besides, in practice, the similarity computation and pseudo label inference can be done within only a few minutes, and bring about negligible costs comparing with the contrastive learning iteration.

As for prototype refinement, we periodically update the feature representation of each class using the mixed samples for every 5 epochs, the same frequency as the similarity computation, and add fixed number of samples to each class. The number of samples is simply set equal to the labeled number of each class, hoping that the mixed samples do not overwhelm the ground truth samples, and we find that the result is robust for a range of mixed samples when they are at the same scale with the ground truth labels.

A. CIFAR-10

We first compare our approach with various methods on CIFAR-10, which contains 50000 images with size of $32 \times 32$. We conduct the experiments with different number of available labels, and each setting is evaluated on 5 folds.

a) Implementation Details: As recommended by [14], we use Wide ResNet-28-2 [38] as backbone, as well as training protocol, $\mu = 4$, $B = 64$, $\tau = 0.95$, and all loss weights (e.g. $\lambda_{ml}$) are set as 1. The projection head is a 2-layer MLP that outputs 64-dimensional embedding. The models are trained using SGD with a momentum of 0.9 and a weight decay of 0.0005. We train our model for 200 epochs, using learning rate of 0.03 with a cosine decay schedule. For the additional hyperparameters, we set the number of negative samples $K = 4096$, $\gamma = 5$. Besides, we use two different augmentations on unlabeled data, the $DA_{soft}$ here is the standard crop-and-flip, and RandAugment [39] combined with the augmentation strategy in [28] is used as another augmentation $DA_{rand}$.

b) Result: We report the results of different label settings with our method, and compare it with some well-known approaches. As shown in Table I, The improvement is more substantial when fewer labeled samples are available. We achieve 10.29% error rate with only 40 labels, which is 3.5% better than the previous best performed method Fixmatch [14] with more robust capability.
B. Imagenet

In this section, we evaluate our method on the large scale ImageNet dataset, which is challenging especially when extremely few shot labels are available. Following previous semi-supervised learning works on ImageNet [17], [28], [44], we conduct experiments with 1% and 10% labels for semi-supervised learning, using the same split of available labels for fair comparisons. Besides, different from the previous contrastive learning based methods that first conduct pretraining and then few shot fine tuning, our method is an end-to-end framework.

a) Implementation Details: For model training, we use ResNet-50 [45] as the backbone, and use the SGD optimizer with weight decay of 0.0001 and momentum as 0.9. The model is trained on 32 Tesla V100 GPUs with batch size of 1024 for unlabeled data and 256 for labeled data, and the initial learning rate is 0.4 with cosine decay schedule. The data augmentation follows the same strategy as [28].

b) Result: We compare our method with several well-known semi-supervised learning approaches, which can be roughly categorized into two aspects, i.e., the conventional semi-supervised learning strategy that use cross entropy loss for data mining, as well as contrastive learning based methods that first pre-train the models without using any image-level labels, and then directly fine-tuning with few shot labeled data. As shown in Table II, we achieve top-1 accuracies of 60.2% and 72.1% with 1% and 10% labeled data respectively, which significantly improves the baseline Moco v2 [30]. Under 1% setting, our method is also better than previous best performed method simCLR v2 [44]. It should be noted that simCLR v2 makes use of extra tricks such as more MLP layers, which has been demonstrated to be effective for semi-supervised learning, while we simply follow the Moco v2 baseline that consists of two MLP layers. The reason is that the extremely few labeled data is too biased to represent the corresponding class, and such error can be magnified during the data mining procedure using cross entropy loss, while our method is robust to the bias, and especially effective when only few shot labels are available.

C. Transfer Learning

1) Transfer to Other Dataset: We evaluated the quality of our representation for transfer learning in two settings: linear evaluation, where a logistic regression classifier is trained to classify a new dataset based on the representation learned by ScCL on ImageNet; and fine-tuning, where we allow all weights to vary during training. In both cases, we follow the settings of [28].

a) Dataset: We use four tiny scale datasets in this experiment, including CIFAR-10/100 [36], FGVC Aircraft [46] and Stanford Cars [47]. We report top-1 accuracy for these datasets with our 10% labels 800 epochs pre-trained model.

b) Implemental Detail: For linear evaluation, we simply follow the settings used in Moco v2 [30], where learning rate is set as 30 and training epochs is 100. While in the fine-tuning task, we use the SGD optimizer to fine-tune all network parameters. The learning rate is set as $10^{-3}$ for backbone and 1 for the classifier, and the total epoch is 120. The learning rate is decayed by 0.1 in 50, 70 and 90 epoch.

c) Baseline: We compare our approach with the supervised and unsupervised methods. The supervised method is trained on ImageNet with standard cross-entropy loss, while we use well-known methods simCLR [28] and Moco v2 [30] as our unsupervised baseline. Their results are all taken from [28] except for Moco v2, which is based on our re-implementation. Besides, in the fine-tuning task, we add a setting by training a model from scratch for comparison.

---

**TABLE I**

| Method       | CIFAR-10  |
|--------------|-----------|
|              | 40 labels | 250 labels | 4000 labels |
| π-model      | -         | 54.26 ± 3.97 | 14.01 ± 0.38 |
| Pseudo-Labeling | -         | 49.78 ± 0.38 | 16.09 ± 0.28 |
| Mean Teacher | -         | 32.21 ± 2.30 | 9.19 ± 0.19 |
| MixMatch     | 47.54 ± 11.50 | 11.05 ± 0.86 | 6.42 ± 0.10 |
| UDA          | 29.05 ± 5.93 | 8.82 ± 1.08 | 4.88 ± 0.18 |
| ReMixMatch   | 19.10 ± 9.64 | 5.44 ± 0.05 | 4.72 ± 0.13 |
| FixMatch w.RA| 13.81 ± 3.37 | 5.07 ± 0.65 | 4.26 ± 0.05 |
| SsCL         | 10.29 ± 2.61 | 5.12 ± 0.41 | 4.51 ± 0.13 |

**TABLE II**

| Method       | 1% labels | 10% labels |
|--------------|-----------|------------|
|              | top-1(%) | top-5(%)   | top-1(%) | top-5(%) |
| Supervised   | 25.4     | 48.4       | 56.4     | 80.4     |

**Semi-supervised Based Method**

| Method       | 1% labels | 10% labels |
|--------------|-----------|------------|
| Pseudo-label[3] | -         | 51.6       | -         | 82.4     |
| VAT[4]       | -         | 47         | -         | 83.4     |
| UDA[40]      | -         | 68.8       | -         | 88.5     |
| Fixmatch(w. RA)[14] | -     | 71.5       | -         | 89.1     |
| S4L-Rotation[6] | -         | 53.4       | -         | 83.8     |

**Self-supervised Based Method**

| Method       | 1% labels | 10% labels |
|--------------|-----------|------------|
| PCL [41]     | -         | 75.3       | -         | 85.6     |
| PIRL[42]     | 30.7      | 57.2       | 60.4      | 83.8     |
| simCLR[28]   | 48.3      | 75.5       | 65.6      | 87.8     |
| Moco v2[30]  | 52.4      | 78.4       | 65.3      | 86.6     |
| BYOL[34]     | 53.2      | 78.4       | 68.8      | 89.0     |
| SwAV[43]     | 53.9      | 78.5       | 70.2      | 89.9     |
| simCLR v2[44] | 57.9     | 82.5       | 68.4      | 89.9     |

**Semi-supervised Contrastive Learning**

| Model       | 1% labels | 10% labels |
|-------------|-----------|------------|
| SsCL (200)  | 54.7      | 79.3       | 70.0      | 89.8     |
| SsCL (800)  | 60.2      | 82.8       | 72.1      | 90.9     |
d) Result: It can be seen from Table III that the supervised method has a clear advantage over self-supervised and semi-supervised training in linear evaluation. By introducing more label prior information, which is beneficial for the separation of representation, we outperform the self-supervised based method on almost all datasets with both linear evaluation and fine-tuning task. Besides, we achieve superior performance to the supervised in almost all datasets with fine-tuning, which also demonstrates the transfer ability of our model.

2) Downstream Tasks: We further evaluate our learned representation on downstream object detection task to uncover the transferability. Following [17], we choose Mask R-CNN [48] with FPN [49] as the backbone, and fine-tune all parameters on the training set, then evaluate it on the validation set of COCO 2017. We report our performance under 1× schedules in Table IV. It can be observe that our method consistently outperforms the supervised method and Moco v2.

D. Study and Analysis

In this section, we give an in-depth analysis of our design for semi-supervised contrastive learning, and demonstrate the benefits of them through comparative experiments. Unless specified, the results are based on ImageNet validation set after 100 epochs.

a) Distinguish with clustering method: Our approach targets at pulling nearest class samples together with calculated prototypes, which is similar to the idea of some clustering based methods. So we use intra-class similarity as an indicator to measure our learned features. We compare our method with two classic clustering-based learning methods, SwAV [43] and Deepcluster v2 [50], and the results are shown in Table V. The intra-class similarity is calculated by averaging cosine distance among all intra-class pairwise samples, and we report the average of 1000 classes on the ImageNet validation set. Our method consistently outperforms these two methods both in 1% and 10% labels available, which validates that it is better for discrimination when more class priors are available.

b) Co-calibration helps for better representation: Table VI provides some interesting results. If we use pretrained backbone and fc head to predict unlabeled training data, we can get top-1 accuracy of 53.2%, while the accuracy drops to 45.3% if we use the nearest class in feature space as its label. However, the overlap of their correct prediction is only 60.1%, in other words, most of the labels that one method predicts wrong are correct in another method, which is the motivation of our co-calibration mechanism. We believe this will help the model learn complementary information during training, and it can be seen that our method gains 2.4% benefits with this operation from the results reported in Table VI.

c) Visualization of Feature Representation: We visualize the embedding feature to better understand the advantage of our semi-supervised contrastive training method. We randomly choose ten classes from the validation set and provide the t-sne visualization of feature representation generated by ScCL, supervised training, unsupervised training (Moco v2 [17]) and clustering based training (Deep cluster v2 [50] and SwAV [43]). The results are shown in Fig. 4, and the same color denotes features with the same label. It can be seen that ScCL takes on higher aggregation property compared with Moco v2, and is slightly worse than the supervised method because it leverages all labels. Besides, we can observe that our method also surpasses the clustering based methods Deep Cluster v2 and SwAV on the feature representation, which is further demonstrated in Table V.

d) Prototype mixture helps for better prediction: We calculate the average accuracy of the training samples which can get the correct positive samples by assigning the nearest category before and after adding the mixture module. The results are shown in Fig. 5. We can see that this module brings about 8% accuracy
improvement at most. In addition, the model with prototype mixture always keeps high accuracy in selecting positive samples, and more correct positive samples are undoubtedly beneficial to improve the performance.

V. ABLATION STUDY

In this section, we conduct detailed ablation studies to inspect how each hyperparameter affects the performance. For efficiency, unless specified, all experiments are conducted with 100 epochs using a randomly selected subset of 100 categories in ImageNet with 1% labels. We report top-1 accuracy in each experiment, where we achieve 63.3% with our default settings.

A. The Effectiveness of Each Module

We first conduct ablation experiments to reveal the effectiveness of each module in our experiments. The modules used for evaluation include integrating contrastive loss into semi-supervised learning (the second row), adding calibration with contrastive similarity (cwcs) in pseudo label loss introduced in Section III (the third row), and adding contrastive learning with pseudo label propagation (plp) introduced in Section III (the fourth row). It can be seen that combining contrastive learning with conventional semi-supervised loss can greatly improve the model’s performance (+13.9%). Besides, our designed strategies in the process of pseudo label loss and contrastive loss clearly improve the performance (+16.1% and +17.5%), and the co-training operation can further benefit the convergence of the model, which brings 22.4% enhancement.

B. Hyper-Parameters of Loss Function

We first analyze the impact of several hyper-parameters, which includes pseudo label loss weight $\lambda_{pl}$, contrastive loss weight $\lambda_{ctr}$, scale factor $\gamma$, weight factor $\alpha_p$, and the margin value $m$.

a) The scale factor $\gamma$: The effect of this coefficient is equal to $\tau$ in [28], [30]. The larger $\gamma$ is, the more it will penalize on the discrepancy, which will break the semantic structure of the embedding distribution, as is shown in Fig. 6. We set it as 10 in our ImageNet experiment.

b) Loss weight: We observe the influence of two weights $\lambda_{pl}$ and $\lambda_{ctr}$ on the performance and report the result in Fig. 6, where $\lambda_{pl} = 5$ and $\lambda_{ctr} = 10$ lead to the best performance. But in the 10% labels setting, we need to increase $\lambda_{pl}$ to 10 and decrease $\lambda_{ctr}$ to 5 to get the best result. We consider it is because the fewer labeled data may cause worse performance on predicting pseudo labels, which needs a larger $\lambda_{ctr}$ to strengthen the contrastive regularization.

c) The weight factor $\alpha_p$: The comparison results of introducing $\alpha_p$ are shown in Table VIII. It can be seen that introducing adaptive weight $\alpha_p$ is beneficial for improving classification accuracy. It brings about 4.9% gains with fixed $\alpha_p$ (set as 1). This is mainly due to the reason that $\alpha_p$ assigns a lower penalty to the positive samples that are less similar to the assigned class, which reduces the influence of these relatively noisy samples during model training.

d) The margin value $m$: Inspired by [51], we set $m$ as negative because the pseudo labels are not always correct during training. Fig. 6 shows the result of different settings, and it is observed that both positive and negative margins can improve the quality of learned features, so it is beneficial to design a less stringent decision boundary for better class separability.

C. Positive Sample Selection Strategy

In this section, we discuss the impact of different positive sample selection strategies. Three alternatives of sample selection methods are listed as follows:

- Random selection: randomly select samples as positives.
- Nearest class: use the nearest category to produce positives.
- Top-3 nearest classes: find three categories closest to the sample, and randomly select a sample from each category as a positive sample.
- Pseudo label: using the model’s prediction on unlabeled data to select positives.

As is shown in Table IX, we can see that if we select additional positive samples randomly or from top-3 classes, the performance is inferior to the baseline Moco v2, which shows that the additional positive samples added by these two methods

---

**Fig. 4.** t-sne visualization of feature representation learned by SsCL, supervised, unsupervised and clustering based methods.

**Fig. 5.** The top-1 accuracy comparisons with and without using prototype mixture.
have negative effects on the model. Moreover, selecting positives from the nearest three categories performs even worse than the randomly selection method. The reason is that although selecting from the top-3 nearest classes has higher recall, it will carry two incorrect samples. These “hard examples” have high similarity with the training samples, but they actually belong to different categories. Besides, being selected by pseudo label is superior to being selected by nearest category by $\sim 2\%$, the reason may be that former can find the correct positives with higher accuracy according to the results shown in Table VI.

**D. The Number of Positive Samples**

Table X shows the results of adding different number of positive samples. 0 here means the original contrastive loss [30] that does not pull together semantically similar images. It can be seen that adding extra positive samples benefits class separability, which significantly boosts the performance.

**E. The Number of Batch Size and GPUs**

Table XI shows the results of training with different batch size and GPUs. The batch size here denotes the batch size for unlabeled data, and the batch size for the labeled data is simply set to a quarter of it. According to the results, we have the following observations: first, reducing the GPU numbers when the total batch size is fixed, which is equivalent to increasing the number of batch size on each GPU, will bring a negative impact on performance; second, too small batch size (e.g., 32) will also cause great damage to the results; finally, if we keep the number of batch sizes on each GPU unchanged, the performance of the model can remain stable. Besides, it is worth mentioning that these problems can be alleviated by adjusting some training hyperparameters, such as learning rate.
TABLE XI

| GPUs | Batch Size | Acc. (%) |
|------|------------|----------|
| 32   | 1024       | 62.7     |
| 8    | 256        | 63.3     |

Fewer GPUs

| 4    | 256        | 56.7     |
| 2    | 256        | 58.3     |

Smaller Batch Size

| 8    | 128        | 64.6     |
| 8    | 64         | 61.9     |
| 8    | 32         | 46.1     |

Fewer GPUs and Smaller Batch Size

| 4    | 128        | 63.9     |

VI. CONCLUSION

This paper proposed a semi-supervised contrastive learning framework, which incorporates contrastive loss into semi-supervised learning, and jointly optimizes the network in an end-to-end way. The main contribution is that we introduce a co-calibration mechanism to interchange predictions between the two branches, which we find is complementary and thus avoid trapped in local minimum. Experiments conducted on several datasets demonstrate the superior performance of the proposed method, especially when extremely few labeled samples are available.

REFERENCES

[1] E. Arazo, D. Ortego, P. Albert, N. E. O’Connor, and K. McGuinness, “Pseudo-labeling and confirmation bias in deep semi-supervised learning,” in Proc. IEEE Int. Joint Conf. Neural Netw., 2020, pp. 1–8.

[2] D. Berthelot et al., “Remixmatch: Semi-supervised learning with distribution alignment and augmented data,” 2019, arXiv:1911.00785.

[3] D.-H. Lee, “Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks,” in Proc. Workshop Challenges Representation Learn., 2013, vol. 3, p. 896.

[4] T. Miyato, S. I. Maeda, M. Koyama, and S. Ishii, “Virtual adversarial training: A regularization method for supervised and semi-supervised learning,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 41, no. 8, pp. 1979–1993, Aug. 2019.

[5] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, and T. Raiko, “Semi-supervised learning with ladder networks,” in Proc. Neural Inf. Process. Syst., 2015, pp. 3546–3554.

[6] X. Zhai, A. Oliver, A. Kolesnikov, and L. Beyer, “S4I: Self-supervised semi-supervised learning,” in Proc. Int. Conf. Comput. Vis., 2019, pp. 1476–1485.

[7] S. Jawed, J. Grabocka, and L. Schmidt-Thieme, “Self-supervised learning for semi-supervised time series classification,” Adv. Knowl. Discov. Data Mining, vol. 12084, pp. 499–511, 2020.

[8] K. Marusy and K. Buza, “Success: A new approach for semi-supervised classification of time-series,” in Proc. Int. Conf. Artif. Intell. Soft Comput., 2013, pp. 437–447.

[9] C. Gan, C. Sun, L. Duan, and B. Gong, “Webly-supervised video recognition by mutually voting for relevant web images and web video frames,” in Proc. Eur. Conf. Comput. Vis., 2016, pp. 849–866.

[10] C. Gan, T. Yao, K. Yang, Y. Yang, and T. Mei, “You lead, we exceed: Labor-free video concept learning by jointly exploiting web videos and images,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 923–932.

[11] F. Zou et al., “Semi-supervised cross-modal learning for cross-modal retrieval and image annotation,” World Wide Web, vol. 22, pp. 825–841, 2018.

[12] Z. Li et al., “A semi-supervised learning approach based on adaptive weighted fusion for automatic image annotation,” ACM Trans. Multimedia Comput., Commun. Appl., vol. 17, pp. 1–23, 2021.

[13] C. Tang et al., “Adaptive hypergraph embedded semi-supervised multi-label image annotation,” IEEE Trans. Multimedia, vol. 21, pp. 2837–2849, 2019.

[14] K. Sohn et al., “Fixmatch: Simplifying semi-supervised learning with consistency and confidence,” 2020, arXiv:2001.07685.

[15] S. Sukhbaatar, J. Bruna, M. Paluri, L. Bourdev, and R. Fergus, “Training convolutional networks with noisy labels,” 2014, arXiv:1406.2080.

[16] Z. Zhang and M. Sabuncu, “Generalized cross entropy loss for training deep neural networks with noisy labels,” in Proc. Neural Inf. Process. Syst., 2018, pp. 8778–8788.

[17] 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., 2020, pp. 9729–9738.

[18] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyond empirical risk minimization,” in Proc. Int. Conf. Learn. Representation, 2018.

[19] M. Guillaumin, J. Verbeek, and C. Schmid, “Multimodal semi-supervised learning for image classification,” in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., 2010, pp. 902–909.

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

[21] D. P. Kingma, S. Mohamed, D. Jimenez Rezende, and M. Welling, “Semi-supervised learning with deep generative models,” Neural Inf. Process. Syst., vol. 27, pp. 3581–3589, 2014.

[22] A. Oliver, A. Odena, C. A. Raffel, E. D. Cubuk, and I. Goodfellow, “Realistic evaluation of deep semi-supervised learning algorithms,” in Proc. Neural Inf. Process. Syst., 2018, pp. 3235–3246.

[23] A. Tarvainen and H. Valpola, “Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results,” in Proc. Neural Inf. Process. Syst., 2017, pp. 1195–1204.

[24] S. Wu, Q. Ji, S. Wang, H.-S. Wong, Z. Yu, and Y. Xu, “Semi-supervised image classification with self-paced cross-task networks,” IEEE Trans. Multimedia, vol. 20, pp. 851–865, 2018.

[25] C. Zhang, J. Cheng, and Q. Tian, “Unsupervised and semi-supervised image classification with weak semantic consistency,” IEEE Trans. Multimedia, vol. 21, pp. 2482–2491, 2019.

[26] M. Jian and C. Jung, “Semi-supervised bi-dictionary learning for image classification with smooth representation-based label propagation,” IEEE Trans. Multimedia, vol. 18, pp. 458–473, 2016.

[27] 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., 2018, pp. 3733–3742.

[28] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” 2020, arXiv:2002.05709.

[29] A. Dosovitskiy, P. Fischer, J. T. Springenberg, M. Riedmiller, and T. Brox, “Discriminative unsupervised feature learning with exemplar convolutional neural networks,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 38, no. 9, pp. 1734–1747, Sep., 2015.

[30] X. Chen, H. Fan, R. Girshick, and K. He, “Improved baselines with momentum contrastive learning,” 2020, arXiv:2003.04297.

[31] R. D. Hjelm et al., “Learning deep representations by mutual information estimation and maximization,” 2018, arXiv:1808.06670.

[32] A. v. d. Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” 2018, arXiv:1707.03748.

[33] Y. Tian, D. Krishnan, and P. Isola, “Contrastive multiview coding,” 2019, arXiv:1906.05849.

[34] J.-B. Grill et al., “Bootstrap your own latent—a new approach to self-supervised learning,” Neural Inf. Process. Syst., vol. 33, pp. 21271–21284, 2020.

[35] Y. Sun et al., “Circle loss: A unified perspective of pair similarity optimization,” in Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit., 2020, pp. 6398–6407.

[36] A. Krizhevsky, “Learning multiple layers of features from tiny images,” Tech. Rep., 2009.

[37] J. Deng, W. Dong, R. Socher, L. J. Li, and F. F. Li, “ImageNet: A large-scale hierarchical image database,” in Proc. Int. Conf. Comput. Vis., 2009, pp. 248–255.
[38] S. Zagoruyko and N. Komodakis, “Wide residual networks,” in Proc. Brit. Mach. Vis. Conf., British Machine Vision Association, 2016, pp. 87.1–87.12.

[39] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, “Randaugment: Practical automated data augmentation with a reduced search space,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops, 2020, pp. 702–703.

[40] Q. Xie, Z. Dai, E. Hovy, M.-T. Luong, and Q. V. Le, “Unsupervised data augmentation for consistency training,” 2019, arXiv:1904.12848.

[41] J. Li, P. Zhou, C. Xiong, and S. C. Hoi, “Prototypical contrastive learning of unsupervised representations,” in Proc. Int. Conf. Learn. Representation, 2021.

[42] M. Ishan and L. van der Maaten, “Self-supervised learning of pretext-invariant representations,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 6707–6717.

[43] M. Caron et al., “Unsupervised learning of visual features by contrasting cluster assignments,” Neural Inf. Process. Syst., vol. 33, pp. 9912–9924, 2020.

[44] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. Hinton, “Big self-supervised models are strong semi-supervised learners,” 2020, arXiv:2006.10029.

[45] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2016, pp. 770–778.

[46] S. Maji, E. Rahtu, J. Kannala, M. Blaschko, and A. Vedaldi, “Fine-grained visual classification of aircraft,” 2013, arXiv:1306.5151.

[47] J. Krause, J. Deng, M. Stark, and L. Fei-Fei, “Collecting a large-scale dataset of fine-grained cars,” in Proc. 2nd Workshop Fine-Grained Vis. Categorization, 2013, pp. 1–2.

[48] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 42, no. 2, pp. 387–397, 2017.

[49] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 2117–2125.

[50] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, “Deep clustering for unsupervised learning of visual features,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 132–149.

[51] J. Xie, X. Zhan, Z. Liu, Y. S. Ong, and C. C. Loy, “Delving into inter-image invariance for unsupervised visual representations,” 2020, arXiv:2008.11702.

Yuhang Zhang received the bachelor’s degree in electrical engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2017. Since 2017, he has been working toward the Ph.D. degree in computer science with Shanghai Jiao Tong University, Shanghai, China. His research interests include manifold learning, semi-supervised/unsupervised learning in classification and detection.

Xiaopeng Zhang received the Ph.D. degree from Shanghai Jiao Tong University, Shanghai, China, in 2017. He was a Research Fellow with the Department of Electrical & Computer Engineering, National University of Singapore, Singapore, from 2017 to 2019, and a Visiting Researcher with the Department of Computer Science, University of Texas at San Antonio, San Antonio, TX, USA, from 2015 to 2016. He is currently a Senior Researcher with CLOUD&AI, Huawei Technologies. His research interests include weakly supervised detection and self-supervised learning.

Jie Li (Senior Member, IEEE) received the Ph.D. degree from the University of Electro-Communications, Tokyo, Japan. He has been with the University of Tsukuba, Japan, where he is currently a Professor with the Faculty of Engineering, Information and Systems. He is also with the Department of Computer Science, Shanghai JiaoTong University, Shanghai, China. He was also on the program committees for several international conferences. He is a Senior Member of the ACM, and a Member of Information Processing Society of Japan. His research interests include mobile distributed multimedia computing and networking, Big Data and cloud computing, network security, and performance evaluation of information systems.

Robert C. Qiu (Fellow, IEEE) received the Ph.D. degree in electrical engineering from New York University (former Polytechnic University), Brooklyn, NY, USA. Since 2015, he has been with the Department of Electrical Engineering, Research Center for Big Data Engineering and Technologies, State Energy Smart Grid R & D Center, Shanghai Jiaotong University, Shanghai, China. He has 15 contributions to 3GPP and IEEE standards bodies. He holds more than six patents and authored more than 70 journal papers or book chapters and 90 conference papers. His research interests include random matrix theory-based theoretical analysis for deep learning and wireless communication applications.

Hashang Xu received the B.S. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2017. He is currently working toward the Ph.D. degree with the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai, China. His research interests include unsupervised learning, few shot learning, and federated learning.

Qi Tian (Fellow, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign, Champaign, IL, USA, in 2002. He is currently the Chief Scientist of artificial intelligence with Huawei Cloud & AI. From 2018 to 2020, he was the Chief Scientist of computer vision with Huawei Noah’s Ark Laboratory. From 2002 to 2019, he was a Full Professor with the Department of Computer Science, University of Texas at San Antonio, San Antonio, TX, USA. He was listed in the Top 10 of the 2016 Most Influential Scholars in Multimedia by Aminer.org. He is an Academician of International Eurasian Academy of Sciences, 2021. He is an Associate Editor for the IEEE TMM, TCSVT, ACM TOMM, MMSJ, and MVA.