FedCL: An Efficient Federated Unsupervised Learning for Model Sharing in IoT

Chen Zhao[1], Zhipeng Gao[1(✉)], Qian Wang[2], Zijia Mo[1], and Xinlei Yu[1]

[1] State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China
gaozhipeng@bupt.edu.cn

[2] Beijing University of Technology, Beijing, China

Abstract. Federated Learning (FL) continues to make significant advances, solving model sharing under privacy-preserving. However, these existing methods are only of limited utility in the Internet of Things (IoT) scenarios, as they either heavily depend on high-quality labeled data or only perform well under idealized conditions, which typically cannot be found in practical applications. As such, a natural problem is how to leverage unlabeled data among multiple clients to optimize sharing model. To address this shortcoming, we propose Federated Contrastive Learning (FedCL), an efficient federated learning method for unsupervised image classification. The proposed FedCL can be summarized in three steps: distributed federated pretraining of the local model using contrastive learning, supervised fine-tuning on a server with few labeled data, and distillation with unlabeled examples on each client for refining and transferring the personalized-specific knowledge. Extensive experiments show that our method outperforms all baseline methods by large margins, including 69.32% top-1 accuracy on CIFAR-10, 85.75% on SVHN, and 74.64% on Mini-ImageNet with the only use of 1% labels.

Keywords: Federated learning · Internet of things · Self-supervised learning · Unsupervised learning

1 Introduction

With the ubiquity of smart devices, federated learning [1] has become one of the most-used types of privacy-preserving model sharing method, and has been popularly applied in many scenarios, such as user habits prediction [2], personalized recommendation [3] and wireless network optimization [4]. Existing federated learning methods typically only consider supervised training settings, where the client data are fully labeled. Yet local data including sophisticated annotations is not realistic for IoT applications since users always have different habits and usage frequencies, which inspires the recent work to combine semi-supervised learning with federated learning to optimize sharing model [3,5,6]. However,

H. Gao et al. (Eds.): CollaborateCom 2022, LNICST 460, pp. 115–134, 2022.
https://doi.org/10.1007/978-3-031-24383-7_7

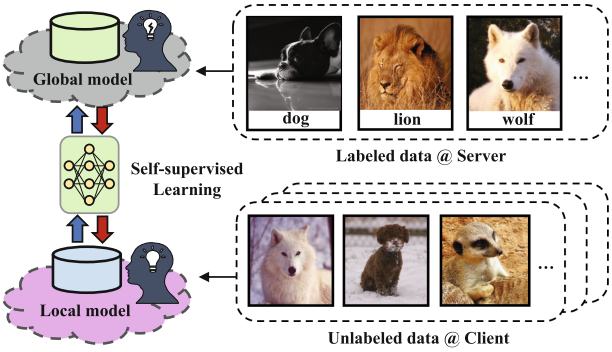

Fig. 1. Illustration of practical IoT scenario in federated unsupervised learning. Many unlabeled data are available at clients, few labeled data are available at the server.

these works normally adopt knowledge transfer techniques between labeled and unlabeled data, which limits the applications when clients are complete without available labeled data. For example, suppose that we have a photo classifier app that automatically categorizes pictures in albums. In this case, the app users may reluctant to annotate these private and sensitive pictures by themself, this leads the service providers can only use limited public on the central server. Thus, in many realistic IoT scenarios, clients' data may completely *unlabeled* and with few labeled data only available at the server. This leads to practical challenges of FL with deficiency of labels, namely, *Federated Unsupervised Learning* (FUL).

Therefore, a universal FL method should work in both supervised and unsupervised scenarios, which inspired the recent work to integrate semi-supervised techniques into the FL framework [5,7] (i.e. employing domain confusion technique to train labeled and unlabeled data). Such research directions are extremely active and have been shown to yield significant accuracy improvements in semi-supervised settings, which can be crucial when making such techniques available in realistic applications. However, different from idealized distribution conditions, data among IoT devices are normally non-independent and identically distributed (non-IID), leading to sharing model performance degradation.

There have been some studies trying to address these issues and provide convergence guarantees for sharing model. To achieve this, researchers use dissimilarity measurement [8], move distance [9], model-contrastive learning [10], and so on. Despite these efforts, these methods fail to achieve good performance when ground-truth annotations are absent. Although some works both address the unsupervised and non-IID problems, such as FedCA [11] and FedU [12], they fail to consider personalization requirements for each client.

In this paper, employing the self-supervised learning technique, we propose an efficient method, named *Federated Contrastive Learning (FedCL)*, to optimize sharing model using unlabeled data among multiple clients, as shown in Fig. 1. Although the recent Federated Self-Supervised Learning (FSSL) works (e.g. [13,14]) have made great progress on label-limited problems, however, for

model sharing in an unsupervised setting, due to the different behavior preferences (e.g., some user like take pet pictures, while others prefer to take life pictures), a big extra challenge is to guarantee local models are personalized after collaboration training, these existing methods are only of limited utility as they either heavily depends on high quality labeled data on clients or ignore the personalized requirement for optimizing sharing models.

By contrast, our FedCL is an effective method that optimizes sharing model when clients come without available labeled data while preserving clients' model personalized. Motivated by recent advances in unsupervised learning [15,16] and self-supervised learning [5,7], we follow unsupervised pre-train at client and supervised fine-tune at server for model training at IoT scenarios. Specifically, we employ RandAugment as data augmentation on each client and compute cosine similarity to learning visual representations on distributed unlabeled data. Then, fine-tuning the sharing model using supervised data at the server to adapt a task-agnostically model for a specific task. Finally, we distill sharing model at each client on local unlabeled data for personalized-preserved and lightweight use.

In summary, our main contributions are three-fold:

- We propose a novel federated self-supervised learning method to pre-train sharing model from clients' unlabeled data, which follows the pretext task of unsupervised learning and can learn distributed network representations by maximizing agreement between augmented images.
- We propose a central fine-tune and personalized distillation in network optimization, which can balance the sharing model consensus and personalization respectively.
- Based on the above two contributions, we propose FedCL, an efficient method to collaborate optimize sharing model under unsupervised settings, where clients' data is completely unlabeled and only a few labeled at the server. Experiments prove that FedCL significantly outperforms related works under both semi-supervised and unsupervised settings.

The remaining of this article is organized as follows. In Sect. 2, we introduce the related work. The design details of the FedCL method, especially the federated self-supervised learning, supervised fine-tuning and personalized distillation are described in Sect. 3. In Sect. 4, we present the experimental results on several commonly used semi-supervised benchmarks. Finally, conclusions are drawn in Sect. 6.

2 Related Work

Below we summarize the related work that involves three main topics, federated semi-supervised learning, federated self-supervised learning, and federated unsupervised learning.

Federated Semi-supervised Learning. Recently, interests of tackling scarcity of labels are discussed [3,17,18]. The motivation of federated semi-supervised

learning methods is to optimize a sharing model iteratively by knowledge transfer between devices. One family of highly relevant methods is based on knowledge transfer [3,5,6] or inter-client consistency [19,20], which is followed by supervised fine-tuning on a few labeled datasets. Aside from the transfer learning paradigm, there is a large and diverse set of approaches for practical federated learning. Another family of methods are based on contrast learning [21] or self-learning [15,16]. The main difference between these methods and ours is that FedCL can efficiently optimize sharing model when clients are without available labels.

Federated Self-supervised Learning. The idea of methods based on self-supervised learning is first to pre-train distributed local model using contrastive learning, and then fine-tune sharing model for the downstream task, where the representative works are FCL [22] and FLESD [14]. FCL [22] proposes a two-staged method, i.e., each client exchanges the features of its local data with other clients and then leverage structural similarity of image examples to align similar features among clients for better model performance. FLESD [14] gathers a fraction of the clients' inferred similarity matrices on a public dataset, and then ensembles the similarity matrices via similarity distillation. Although these methods can optimize a generalized model, the local model personalization cannot be well preserved due to there is no developed personalized model for each client.

Federated Unsupervised Learning. the motivation of federated unsupervised learning methods is to learn a generic representation from decentralized data. Jin et al. [23] first point out the advantages that leverage unlabeled data for unsupervised training. Then, researchers propose a series of unsupervised works to improve the performance on data privacy [24], specific application [25,26], and non-IID distribution [11]. Zhuang et al. [12] present a new framework to leverage unlabeled data while dynamically updating network predictors. Although these works study federated learning in unsupervised scenarios, they bypass the non-IID problems.

Unlike the above methods, our work combines representation learning and knowledge distillation, considering the model personalized requirement caused by data non-IID, and can automatically classify image samples when clients without labels.

3 Method

3.1 Federated Unsupervised Problem Definition

Given a set of IoT clients $C = \{c_1, c_2, ..., c_n\}$ and a global server G, each client possesses a local unlabeled dataset $\mathbf{x}_U^n$. Our method is modeled as a function Φ to optimize a sharing model $N_G = \Phi(\mathbf{x}_U, \mathbf{x}_L)$, where $\mathbf{x}_U = \{\mathbf{x}_U^1, \mathbf{x}_U^2, ..., \mathbf{x}_U^n\}$ are clients' local unlabeled data that used to learn model representation and specific-task, $\mathbf{x}_L$ is server's labeled data that are used to fine-turn the sharing model. The total objective function $\mathcal{L}_\Phi$ of N_G can be represented as

$$\mathcal{L}_\Phi = \mathcal{L}_{self}(\mathbf{x}_U) + \mathcal{L}_{fine}(\mathbf{x}_L) + \mathcal{L}_{distill}(\mathbf{x}_U), \tag{1}$$

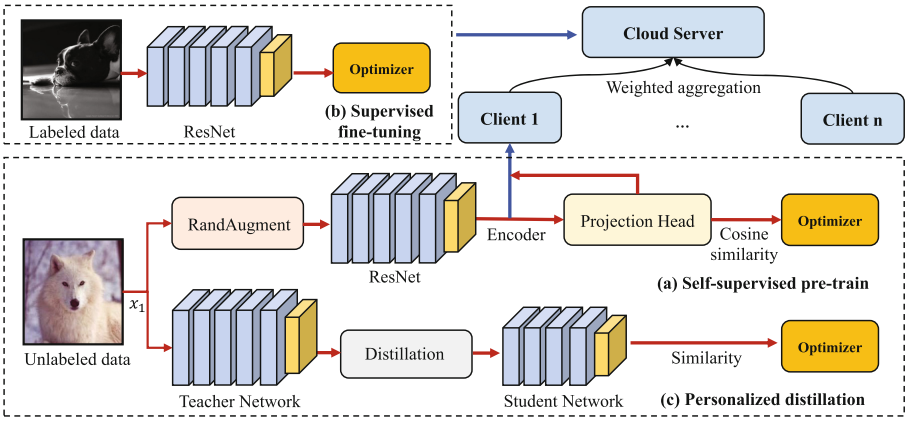

Fig. 2. Overall pipeline. Given a set of IoT clients $C = \{c_1, c_2, ..., c_n\}$ and a cloud server G, each client possesses a local unlabeled dataset $\mathbf{x}_U^n$. Our method is modeled as a function Φ to optimize a sharing model $N_G = \Phi(\mathbf{x}_U, \mathbf{x}_L)$. Our method can be summarized in three steps: federated self-supervised learning, supervised fine-tuning, and personalized distillation.

where self-supervised loss $\mathcal{L}_{self}$ and distillation loss $\mathcal{L}_{fine}$ are used for self-supervised learning and fine-tuning model respectively, and network distillation loss $\mathcal{L}_{distill}$ are used to improve the network for local specific-task.

A key challenge of federated unsupervised learning is that data among clients are always non-IID since users always have different habits and usage frequencies, both data size and distribution may also vary heavily on different devices [27]. These independent distributions are almost unable to be learned and optimized by a simple weight average and could result in a poor representation. In this work, we aim to leverage these unlabeled data from clients to learn a generic representation and personalized model without violating users' privacy.

3.2 FedCL Overview

Inspired by the recent successes of SimCLRv2 [16], the proposed FedCL leverages local unlabeled data in both sharing model optimization and the local model distillation process. As shown in Fig. 2, the first time the local unlabeled data is used for learning visual representations via federated unsupervised pretraining. Then the general sharing model is adapted for the downstream task via central labeled data fine-tuning, to further improve classification performance. To this end, we train the student model from the fine-tuned model with unlabeled data on each client to further improve the model for clients specific-task. Our method can be summarized in three steps: federated self-supervised learning, supervised fine-tuning, and personalized distillation.

Before presenting technique details, we introduce the training pipeline of FedCL with the following steps: (a) Self-supervised pre-train: each client conducts a pretext task through representation learning that can be used to obtain

Fig. 3. Illustration of the RandAugment operators. In these examples, the strength of the augmentation increases as the distortion magnitude M increases. Each augmentation is transformed stochastically with some internal parameters (e.g. rotation degree, cutout region, color distortion).

semantically meaningful features. (b) Supervised fine-tuning: the server fine-tunes the sharing model to further improve model performance and then aggregates the client's model parameters to update sharing model. (c) Personalized distillation: to improve the local model for specific tasks, we train the student model for each client based on sharing model. We will detail the three steps in Sect. 3.3 and Sect. 3.4, respectively.

3.3 Federated Self-supervised Learning

Inspired by contrastive learning [28,29], FedCL learns distributed model representations by maximizing agreement between augmented image examples on each client, as shown in Fig. 2 (a), and we will detail the two components as follows.

Data RandAugment. The goal of data augmentation is to constrain model predictions to be invariant to noise and through the similarity between augmentation data to learn visual representation. However, one obstacle of existing augmentation methods is a separate search of optimal parameters which significantly increases the clients' training complexity and is computationally expensive. Considering the limited computing and storage resources of the clients. We use a data augmentation called RandAugment [21] which removes the search phase and requires no labeled data.

Specifically, in clients data augmentation phase, we consider several common augmentation including {*identity, rotate, posterize, sharpness, translate-x, translate-y, shear-x, shear-y, autoContrast, solarize, contrast, equalize, color, brightness*} and stochastically choosing $T = 14$ available transformations to apply each augmentation. To reduce the parameter space while still preserving image diversity, we replace the learned policies for applying each transformation with a parameter-free procedure of selecting a transformation with probability $1/T$ [30]. As shown in Fig. 3, Randaugment contains two parameters R and M, which are used to control the transformation numbers and distortion magnitudes, we observe that the larger values of R and M will increase regularization strength

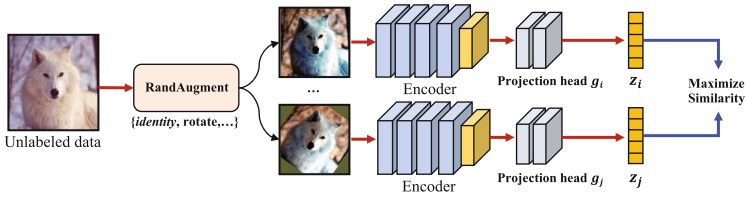

Fig. 4. Pipeline of self-supervised pre-train.

and the two hyperparameters may suffice for parameterizing all transformations. The goal of the pretext task is to minimize the distance between image samples x and their augmentations $T[x]$, expressed as min $d(f(x), T(f(x)))$, we observe that the samples with similar features are assigned to semantically similar [31], any pretext task that satisfies the equation above can be used for representation learning. We refer to Sect. 4.2 for a concrete experiment.

Self-supervised Pre-train. Inspired by the SimCLR [15,16] which introduces a learnable nonlinear transformation between the representation and the contrastive loss. The local pre-train is shown in Fig. 4, Our self-supervised pre-train is designed according to each client's unlabeled dataset. Specifically, given a set of local data $\mathbf{x}_U^n$, each image examples $x \in \mathbf{x}_U^n$ is augmented twice using RandAugment, creating two views x_{2k} and x_{2k-1}, and encode two images via encoder network ResNet to generate representations g_{2k} and g_{2k-1}. Then we transform the representation via a non-linear function to generate z_{2k} and z_{2k-1} that are used to compute the contrastive loss. We adopt the normalized temperature-scaled cross-entropy loss instead of cross-entropy loss as contrastive loss, represent as

$$\mathcal{L}(i,j) = -\log \frac{\exp(\text{sim}(z_i, z_j)/\tau)}{\sum_{k=1}^{2K} \mathbb{1}_{[k \neq i]} \exp(\text{sim}(z_i, z_j)/\tau)}, \tag{2}$$

where i, j is the augmented examples from the same image, $\text{sim}(z_i, z_j) = \frac{z_i^{\mathrm{T}} z_j}{\tau \|z_i\| \|z_j\|}$ is cosine similarity between two images, $\mathbb{1}_{[k \neq i]}$ is indicator function evaluating to 1 if $k \neq i$ and τ is a temperature scalar.

Finally, the self-supervised loss of client n can be represented as

$$\mathcal{L}_{self}^n = \frac{1}{2K} \sum_{k=1}^{K} [\mathcal{L}(2k-1, 2k) + \mathcal{L}(2k, 2k-1)], \tag{3}$$

where K is the local batch size. In each communication round, we choose p fraction of clients to train local unsupervised networks and then update model parameters to the server for weighted aggregation. Accordingly, the sharing model parameters can be updated as

$$\omega_{t+1} \leftarrow \sum_{i=1}^{S_t} \frac{|\mathbf{x}_U^n|}{|\mathbf{x}_U|} \omega_t^i, \tag{4}$$

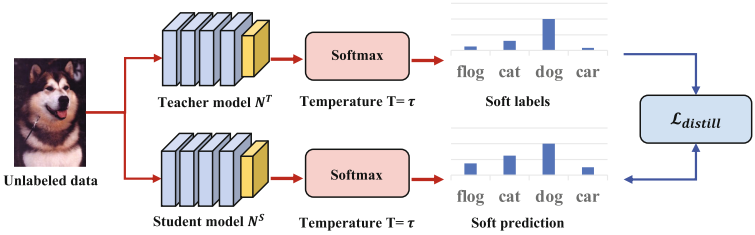

Fig. 5. Pipeline of personalized distillation.

where $|\mathbf{x}_U|$ is unlabeled data size, ω is model parameters, $S_t = n * p$ is the collection of clients participating in this round of training, and t is the current communication round.

3.4 Fine-Tuning and Distillation

To further improve the sharing model performance with central labeled data and local unlabeled data, we adopt supervised fine-tuning and unsupervised distillation to optimize sharing model and preserve local models personalized respectively.

Supervised Fine-Tuning. Fine-tuning is widely used to adapt task-agnostically model for a downstream task. When the self-supervised sharing model is converged, we fine-tuned the sharing model on the server to improve the accuracy of the model. In FedCL, we incorporate the second layer of MLP projection heads into ResNet during fine-tuning, instead of throwing it all away. Then we fine-tuned the network with a few labeled data examples for a specific task. The fine-tuning loss $\mathcal{L}_{fine}$ are not the key points of our work, thus we roughly use the definition in [16,32].

Personalized Distillation. To further improve the network for local specific-task, here we use personalized distillation to train the local model for the target task. As shown in Fig. 5, we fixed the teacher network and only train the student network with local unlabeled data in this procedure. Inspired by [33–36], we use the sharing network as teacher model to classify impute labels for training a student network. Therefore, the distillation loss of client n can be represent as

$$\mathcal{L}^n_{distill} = -\sum_{x_i \in \mathbf{x_U}} [\sum_y p_T(y|x_i;\tau) \log p_S(y|x_i;\tau)], \tag{5}$$

where $p(y|x_i;\tau) = \exp(f(x_i,y)/\tau)/\sum_y \exp(f(x_i,y)/\tau)$, p_T and p_S is the data distribution of teacher network and student network respectively. According to the practical IoT applications, the architecture of the student network and teacher network can be the same or smaller.

Considering that the server may contain a large amount of labeled and unlabeled data in the realistic IoT scenarios, we further extend the self-distillation

procedure to the semi-supervised scenario [16]. While Eq. 5 only focuses on the distillation using unlabeled examples, when there are a significant amount of labeled examples, we can also use a weighted combination to compute the distillation loss with the ground truth labeled examples, the distillation loss can be represent as

$$\mathcal{L}_{distill}^{n} = -(1-\alpha) \sum_{x_i \in \mathbf{x_U}, \mathbf{x_L}} [\log p_S(y|x_i;\tau)] - \sum_{x_i \in \mathbf{x_U}} [\sum_{y} p_T(y|x_i;\tau) \log p_S(y|x_i;\tau)], \tag{6}$$

This procedure can be performed to improve the task-specific performance of each client.

4 Experiment

In experiments, we detail the implementation settings of FedCL, and then we mainly evaluate the performance of our method in two aspects, accuracy and scalability.

4.1 Implementation Details

Datasets. Following the semi-supervised setting in [15,37], we evaluate the efficacy of FedGAN on several commonly used SSL image benchmarks. Specifically, we perform experiments with varying amounts of labeled data on three real-world datasets, including CIFAR-10 [38], SVHN [39], and Mini-ImageNet [40], with a randomly sub-sampled 1% or 10% of labeled images on the server and the rest are distributed on clients, which are widely used for evaluating image-processing deep learning algorithms. We use these datasets in the form of unstructured and low-pixel images, which is similar to the unprocessed fragmented data collected in the IoT scenario, such as image recognition and classification in the smart devices usage process.

CIFAR-10 constitute the proof of experiment since it is a well-established benchmark, CIFAR-10 is an image recognition dataset for machine learning, that contains 60000 color images covering 10 categories.

SVHN is a real-world image dataset for developing machine learning and object recognition algorithms. It can be seen as similar in flavor to MNIST, but incorporates an order of magnitude more labeled data (over 600,000 digit images) and comes from a significantly harder, unsolved real-world problem, which is commonly used in semi-supervised learning evaluation.

Mini-ImageNet is selected from the Imagenet dataset, which is a very famous large-scale visual data set established to promote the research of visual recognition. The mini-ImageNet dataset contains 60000 RGB images with 100 categories, including 600 samples in each category, and the specification of each picture is 84×84. Compared with the CIFAR-10 dataset, the mini-ImageNet dataset is more complex, but it is more suitable for prototype design and experimental research.

We follow the most used linear evaluation protocol [28] to fine-tune the local model. Beyond linear evaluation and fine-tuned, we also compare against SOTA on federated semi-supervised and unsupervised learning. The previous works were almost performed with a uniform distribution of data in which every client was assigned the same data size. In realistic IoT scenarios, however, the data on different clients will typically vary heavily in category and size. To simulate different degrees of unbalancedness, we split the data according to [27] as non-IID settings, and the data size of each client is assigned a fraction:

$$\varphi_c(\delta, \gamma) = \frac{\delta}{n} + (1 - \delta)\frac{\gamma^c}{\sum_{j=1}^{n} \gamma^j}, \tag{7}$$

where δ controls the minimum data size on each client, and γ controls the data concentration.

Experiment Setting. Our program is implemented by PyTorch and all experiments are performed on a server with four NVIDIA Geforce RTX 3090 GPUs. For all experiments, the sharing model has the same network architecture as the local model, for a fair comparison, we take the same neural network architecture as SimCLRv2 [16]. By default, we set client number $n = 20$ and $R = 3$, $M = 19$ in Sect. 3.3, $\tau = 0.9$ in Eq. (2), $K = 256$ in Eq. (3), $p = 0.4$ in Eq. (4). We use ResNet-50 as the base encoder, a 3-layer MLP as the projection head and select the second layer as optimal classifier layer, the optimizer is Adam, the learning rate is 0.001 and momentum parameters $\beta_1 = 0.5$, $\beta_2 = 0.999$. To evaluate the impact of data distribution on model performance, we set $\delta = 0.1$ and $\gamma = 0.9$ as our non-IID data partition settings. We train the self-supervised local model, fine-tuning and unsupervised distillation for 100, 400 and 200 epochs respectively, and the communication round is 200 in federated self-supervised learning.

4.2 Ablation Study

In the following, we analyze the effects of several components in our model.

Influence of RandAugment. In Sect. 3.3, the data augmentation is used to learn the visual representation of the network. We evaluate the CIFAR-10 dataset ten times and reach between 66.48% and 70.34% average Top-1 test accuracy with a median of 69.32%. Then we replace the RandAugment with random crop and color distortion and compare it with default components, experimental results in Table 1 show that the model performance has slightly dropped around 2.64%, which means the RandAugment component can effectively improve self-supervised learning performance from unlabeled data.

Influences of Central Fine-Tuning and Personalized Distillation. To further analyze the improvements of FedCL, we removed the central fine-tuning and personalized distillation respectively. Experimental as shown in Table 1, we can observe that average model performance has dropped (2.56%–11.84%). This gap

Table 1. Personalized accuracy of FedCL on CIFAR-10 dataset with different components

Methods	Label fraction	
	1%	10%
W/O RandAugment	66.68	70.59
W/O central fine-tuning	57.48	65.82
W/O personalized distillation	66.76	76.44
FedCL (with default settings)	**69.32**	**74.35**

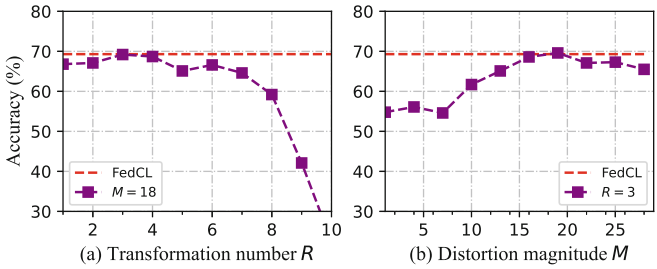

Fig. 6. Optimal data augmentation parameters. All results report CIFAR-10 test validation for ResNet-50 model architecture averaged 10 random initializations. (a) Varying the transformation number. (b) Measuring the effect of augmentation while varying the distortion magnitude. The accuracy of FedCL with default hyperparameters is in the red dotted line. (Color figure online)

tells us that our FedCL improves model optimization while keeping reliable classification knowledge. This effective separation of pretext task and downstream task enhances the overall performance of our methods even without available labeled data on clients.

4.3 Performance Analysis

Below we conduct a series of experiments to evaluate the role of hyperparameters (including transformation number R, distortion magnitude M, training batch size K, and projection head) in FedCL to tease apart the experimental factors that are important to FedCL's improvement.

Influence of Transformation Number and Distortion Magnitude. To explore the influence of parameters in data augmentation, we experiment with different RandAugment hyperparameters on the CIFAR-10 dataset since it is a baseline and well-studied dataset. Our goal is to demonstrate the relative benefits of employing this method over previous random augmentation methods.

We systematically measure the effects of transformation number R and distortion magnitude M with Top-1 model accuracy. We train local models with unlabeled data and measure the average accuracy compared to a baseline model

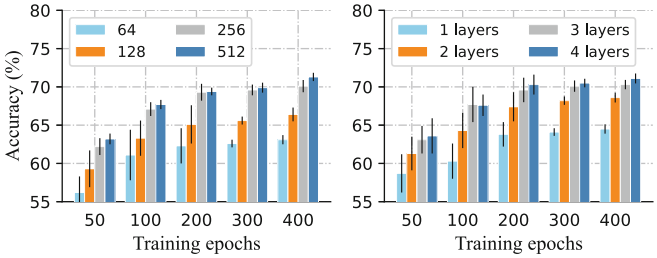

Fig. 7. Evaluation of the different batch sizes and layers of projection head. (a) Results with different batch sizes. (b) Results with different projection heads.

trained with random augmentations (i.e. rotate and color). We fix $M = 18$, the experimental results in Fig. 6(a) show that too much augmentation algorithm fusion will lead to the system underfitting, which will reduce the performance of the shared model. Therefore, we set $R = 3$ as our default value. As shown in Fig. 6(b), we vary M between 1 and 29 in our experiments. We observe that the model accuracy increases monotonically with distortion magnitude. We conjecture that aggressive data augmentation leads to a low single-to-noise ratio in clients' data. Regardless, This experiment trend highlights the benefits of the RandAugment method, here, we set $M = 19$ as our default value.

Influences of Batch Size and Projection Head. We increase the batch size in a range of 64, 128, 256, and 512 on the CIFAR-10 task, Our method shows consistent performance improvement as the batch size increases. Note that the model accuracy shown in Fig. 7(a) can be improved if we allow a larger batch size, however, the improvement is small when batch sizes are larger than 256, considering the limited computing resources of the client, we set batch size is 256 as our default value.

To further evaluate the effect of projection head, we pre-train ResNet using federated self-supervised learning with different numbers of projection heads (from 1 to 4 fully connected layers) and examine average model Top-1 accuracy after fine-tuning local models. The experimental results are shown in Fig. 7(b), we find that using a deeper projection head during local self-supervised training is better when fine-tuning from the optimal layer of the projection head, here, we set the number of layers in the projection head as 3 as our default value.

4.4 Comparison with Related Methods

We compare our method with other related federated semi-supervised learning methods and naive combination of federated learning and semi-supervised learning, that have the potential to optimize sharing model in unsupervised settings. We evaluate the classification ability learned on CIFAR-10 datasets, following the most used linear evaluation protocol [28] and testing the sharing model accuracy. We first train a visual representation with unlabeled data using FedCL and

Table 2. Averaged local performance on semi-supervised and unsupervised task

Methods	Model	Label fraction	
		1%	10%
Supervised baseline [37]	ResNet-50	36.63	51.94
Semi-supervised learning methods			
FL Pseudo Label [41]	ResNet-50	–	56.73
FL UDA [21]	ResNet-50	–	60.15
DS-FL [5]	VGG-16	55.38	62.71
FL FM-GAN [42]	GAN	58.34	64.28
FedMatch (labels-at-client) [7]	ResNet-9	62.47	70.88
Unsupervised learning methods			
FedMatch (labels-at-server) [7]	ResNet-9	56.35	62.35
FedMatch (labels-at-server) [7]	ResNet-50	62.84	71.21
FCL [22]	U-Net	61.26	66.47
FL SimCLR [15]	ResNet-50	64.17	71.75
Our methods with different network architecture			
FedCL (self-distilled)	ResNet-50	69.32	74.35
FedCL (distilled)	ResNet-50(2×)	70.83	75.19
FedCL (distilled)	ResNet-152(3×)	**72.41**	**78.73**

other baseline methods for 200 epochs; Next, we fix the representation parameters and train a new classifier at the output layer. The following are baselines and training details. (1) *Federated semi-supervised learning methods:* we compare our method with semi-supervised learning methods, including FL Pseudo Label [41], FL UDA [21], DS-FL [5], FL FM-GAN [42], and FedMatch [7] that only have labels at clients. (2) *Federated unsupervised learning methods:* we compare our method with unsupervised learning methods, including FedMatch [7] that have labels at server, FCL [22], and FL SimCLR [15]. Besides, we also compare FedCL with different network architectures. For the compared supervised and semi-supervised baselines, we evenly split labeled data for each client. We find this setting to be realistic as IoT scenarios, since the user may not have willing and skills that would label the examples collected by smart devices.

Table 2 shows the performance comparison of our FedCL and related methods on 1% and 10% label fraction tasks with default network architecture, our method consistently outperforms these baseline methods with large margins (about 6%–13%) in both label fractions. In particular, compared with supervised baselines in fine-tuning settings, we observe that our model achieves significant improvements in model performance, which means our method can effectively address the issues of lack labeled caused by requirements of various application scenarios.

To further study the effect of network architecture on our method, we train ResNet by varying width and depth. We can see that increasing network architecture can improve model performance (about 4%). While even the smallest model can offer decent or even competitive performances compared to the related works. We believe that these comparison experimental results further strengthen our paper.

4.5 Analysis of Scalability

Performance Under Different Label Fraction. To study the effects of label fraction and number of clients. We conduct experiments with our methods and two baselines (containing supervised methods FedAvg and semi-supervised FedMatch). As shown in Fig. 8(a), our FedCL has good scalability when the label fraction is changing and shows much performance improvement when the label fraction increases. Interestingly, we observe that our method improves most when there are fewer label data, which implies that our FedCL has the effectiveness of contrastive learning and preserves reliable knowledge in the novel federated unsupervised scenarios.

Performance Under Different Number of Clients. With the increase of clients, the data will become more scattered, leading to model performance degradation. To evaluate the scalability of our method, we conduct a comparative experiment on related works and our method. We train on CIFAR-10 datasets with 10% labeled examples for each work (including FedAvg, FedMatch, and our results under different clients number). Then we train each client for 50 epochs in parallel and take the average accuracy in 100 communication rounds as model performance.

The comparison results in Fig. 8(b) show that with the increase of clients, our method achieves better performance compared with the baselines (15%–25.41%). We conjecture that this is because, for the unsupervised pretext task, the model may not sufficiently be learned by local data, while the FedCL method overcomes it by utilizing the data augmentation and contrastive learning. Due to the huge amount of IoT clients in the actual application scenario, our method obviously has strong advantages.

Performance Under Different Distillation Methods. To analyze the influence of different distillation methods on model performance, we trained distributed model with local unlabeled data to preserve personalized in two ways: (a) student model has the same structure as the teacher model (excluding projection heads); (b) student model has relative small structure than the teacher model. Here, we set ResNet-152 as our teacher model.

The evaluation results as shown in Fig. 9, for both personalized methods, distillation improves the average model performance by transferring task-specific knowledge to a client model. For FedCL, even though it reduced model parameters amount, our method still significantly improves the semi-supervised learning performance, which indicates our FedCL is meaningful for lightweight applications in the IoT scenarios.

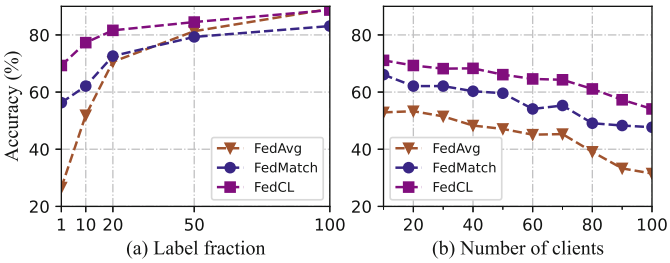

Fig. 8. Evaluation on label fraction and number of clients. (a) Model performance with different label fractions. (b) Model performance with a different number of clients.

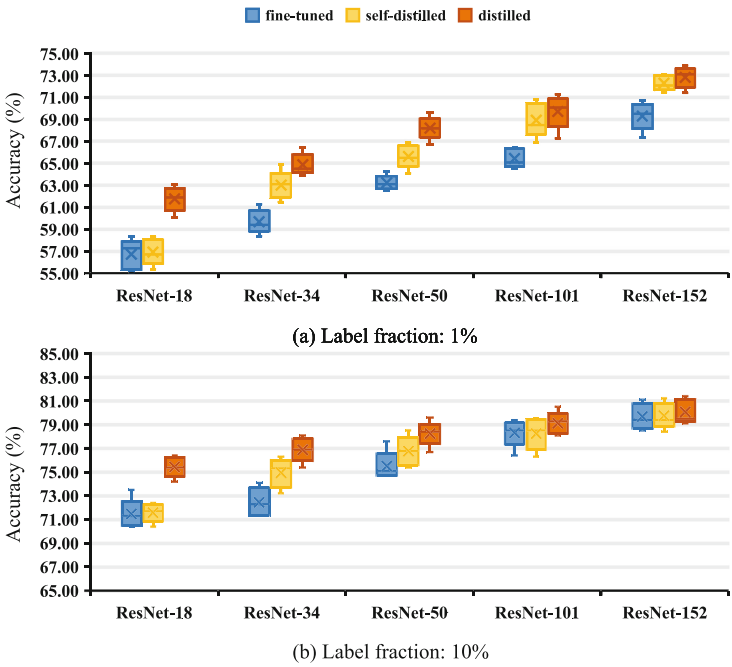

Fig. 9. Top-1 accuracy of FedCL method compared to the fine-tuned model. (a) The self-distilled model has the same structure as the teacher model. (b) The distilled model is trained by the ResNet-152 model.

Performance on Fine-Tuning and Linear Evaluation. For fine-tuning, after federated self-supervised training on clients, we add a full-connected layer after encoder as a linear classifier and use 1% labeled data to train the whole network, we do not use any regularization algorithm. For linear evaluation, we take the same training steps as fine-tuning, except we train the linear classifier on full labeled data.

Table 3. Compared with supervised method on different datasets

Methods	CIFAR-10	SVHN	Mini-ImageNet
Linear evaluation			
Supervised Learning [1]	88.53	89.73	79.42
FCL [22]	62.49	75.72	51.24
FL SimCLRv2 [16]	63.27	80.05	69.71
FedCL (ours)	67.41	81.88	69.94
Fine-tuned (default setting)			
Supervised Learning [1]	92.52	94.18	87.36
FCL [22]	63.44	77.68	65.44
FL SimCLRv2 [16]	66.59	83.68	72.79
FedCL (ours)	**69.32**	**85.75**	**74.64**

The experimental results are shown in Table 3, compared with the linear evaluation that linear classifier layers are frozen, FedCL only uses 1% labeled data and achieves better model performance (3.87%–4.70%), and linear evaluation uses the full labeled data. We note that our method can be improved by incorporating extra unlabeled data and more complex encoder network.

Performance Under Different Datasets. As our goal is not to optimize model performance on CIFAR-10, but rather to provide further confirmation of our improvements on model sharing in IoT scenarios, we use the ResNet-50 as the base architecture for SVHN and Mini-ImageNet experiments. Since the Mini-ImageNet examples are much bigger than CIFAR-10 and SVHN, we replace the first 3×3 Conv of stride 1 with 7×7 Conv of stride 2 and increase max-pooling operations after the first convolutional layer. The rest of the settings (training rounds, batch size, optimizer, etc.) are the same as CIFAR-10.

The experimental results are shown in Table 3, we observe that our model outperforms all naive combinations of federated learning and self-supervised learning for both linear evaluation and fine-tuned tasks. In particular, under the unsupervised IoT scenario, we observe that the naive combination methods significantly suffer from the so-called catastrophic forgetting [43] and their performances keeps deteriorating after sharing model converged. This phenomenon is mainly caused by fine-tuned model failing to properly preserve local personalization, in which case the learned sharing model from the central labeled data causes inter-task interference. Contrarily, our methods adapt to the optimized personalized target to integrate new knowledge (plasticity) without significant interference of new unsupervised examples on existing knowledge (stability). In addition, we observe that our best model trained with batch size 256 can achieve 69.32%, 85.75%, and 74.64% top-1 on three datasets. Although model performance is worse than the supervised method (8.43%–23.2%), consider that we only use 1% labeled data, and the supervised baseline achieves 87.36% need full-

labeled datasets, which implies our method has the scalability and strength on commonly used semi-supervised datasets.

5 Advantage and Limitation

According to the advantages of representation learning and self-supervised learning afore-mentioned in Sect. 1, our federated unsupervised learning method can leverage unlabeled data from multiple clients to learn image classification, and can dynamically balance the sharing model consensus and personalization.

However, our FedCL is a three-step method that combines improved techniques and requires a manual search for the optimal parameter combination, which is time-consuming and error-prone. This is much different from other methods that mainly used representation learning and supervised fine-tuning (e.g., [12,16,24]). Therefore, in future work, we plan to examine the potential of reinforcement learning to discover the structure of the unsupervised model for the best performance given a set of clients and datasets.

6 Conclusion

In this paper, we propose FedCL, an efficient federated learning method for unsupervised image classification. To guarantee the sharing method are efficient and scalable, we designed a local self-supervised pre-train mechanism, a central supervised fine-tuning, and a personalized distillation mechanism.

Our experimental results demonstrate that FedCL can effectively optimize sharing model on commonly used semi-supervised and real-world datasets while preserving locally personalized. At the same time, we note that FedCL also provides good scalability, our analysis in Sect. 3.4 suggests that because it preserves the local knowledge for each client, and is optimized for specific-task.

In our experimental process, we performed a manual search for the encoder network architecture, which is time-consuming and error-prone. In future work, we plan to examine the potential of reinforcement learning to discover the structure of the sharing model and optimal hyperparameters for the best performance given a set of clients and datasets.

Acknowledgement. This work is supported by the National Natural Science Foundation of China (62072049).

References

1. McMahan, B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Proceedings of Machine Learning Research, Fort Lauderdale, FL, USA, vol. 54, pp. 1273–1282 (2017)
2. Hard, A., et al.: Federated learning for mobile keyboard prediction. arXiv preprint arXiv:1811.03604 (2018)

3. Zhu, Y., Liu, Y., Yu, J.J.Q., Yuan, X.: Semi-supervised federated learning for travel mode identification from GPS trajectories. IEEE Trans. Intell. Transp. Syst. 1–12 (2021). https://doi.org/10.1109/TITS.2021.3092015
4. Tran, N.H., Bao, W., Zomaya, A., Nguyen, M.N.H., Hong, C.S.: Federated learning over wireless networks: optimization model design and analysis. In: IEEE INFOCOM 2019 - IEEE Conference on Computer Communications, pp. 1387–1395 (2019). https://doi.org/10.1109/INFOCOM.2019.8737464
5. Itahara, S., Nishio, T., Koda, Y., Morikura, M., Yamamoto, K.: Distillation-based semi-supervised federated learning for communication-efficient collaborative training with non-iid private data. arXiv preprint arXiv:2008.06180 (2020)
6. Nandury, K., Mohan, A., Weber, F.: Cross-silo federated training in the cloud with diversity scaling and semi-supervised learning. In: ICASSP 2021–2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3085–3089 (2021). https://doi.org/10.1109/ICASSP39728.2021.9413428
7. Jeong, W., Yoon, J., Yang, E., Hwang, S.J.: Federated semi-supervised learning with inter-client consistency & disjoint learning. arXiv preprint arXiv:2006.12097 (2020)
8. Sahu, A.K., Li, T., Sanjabi, M., Zaheer, M., Talwalkar, A., Smith, V.: Federated optimization in heterogeneous networks. CoRR abs/1812.06127 (2018). http://arxiv.org/abs/1812.06127
9. Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., Chandra, V.: Federated learning with non-IID data. CoRR abs/1806.00582 (2018). http://arxiv.org/abs/1806.00582
10. Li, Q., He, B., Song, D.: Model-contrastive federated learning. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10708–10717 (2021). https://doi.org/10.1109/CVPR46437.2021.01057
11. Zhang, F., et al.: Federated unsupervised representation learning. arXiv preprint arXiv:2010.08982 (2020)
12. Zhuang, W., Gan, X., Wen, Y., Zhang, S., Yi, S.: Collaborative unsupervised visual representation learning from decentralized data. CoRR abs/2108.06492 (2021). https://arxiv.org/abs/2108.06492
13. Saeed, A., Salim, F.D., Ozcelebi, T., Lukkien, J.: Federated self-supervised learning of multisensor representations for embedded intelligence. IEEE Internet Things J. **8**(2), 1030–1040 (2021). https://doi.org/10.1109/JIOT.2020.3009358
14. Shi, H., Zhang, Y., Shen, Z., Tang, S., Li, Y., Guo, Y., Zhuang, Y.: Federated self-supervised contrastive learning via ensemble similarity distillation. CoRR abs/2109.14611 (2021). https://arxiv.org/abs/2109.14611
15. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International Conference on Machine Learning, pp. 1597–1607. PMLR (2020)
16. Chen, T., Kornblith, S., Swersky, K., Norouzi, M., Hinton, G.: Big self-supervised models are strong semi-supervised learners. arXiv preprint arXiv:2006.10029 (2020)
17. Zhang, C., Zhu, Y., Markos, C., Yu, S., Yu, J.J.: Towards crowdsourced transportation mode identification: a semi-supervised federated learning approach. IEEE Internet Things J. (2021). https://doi.org/10.1109/JIOT.2021.3132056
18. Thakur, A., Sharma, P., Clifton, D.A.: Dynamic neural graphs based federated reptile for semi-supervised multi-tasking in healthcare applications. IEEE J. Biomed. Health Inform. (2021). https://doi.org/10.1109/JBHI.2021.3134835
19. Verma, V., Kawaguchi, K., Lamb, A., Kannala, J., Bengio, Y., Lopez-Paz, D.: Interpolation consistency training for semi-supervised learning. arXiv preprint arXiv:1903.03825 (2019)

20. Sohn, K., et al.: Fixmatch: simplifying semi-supervised learning with consistency and confidence. arXiv preprint arXiv:2001.07685 (2020)
21. Xie, Q., Dai, Z., Hovy, E.H., Luong, M.T., Le, Q.V.: Unsupervised data augmentation. CoRR abs/1904.12848 (2019). http://arxiv.org/abs/1904.12848
22. Wu, Y., Zeng, D., Wang, Z., Shi, Y., Hu, J.: Federated contrastive learning for volumetric medical image segmentation. In: de Bruijne, M., et al. (eds.) MICCAI 2021. LNCS, vol. 12903, pp. 367–377. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-87199-4_35
23. Jin, Y., Wei, X., Liu, Y., Yang, Q.: Towards utilizing unlabeled data in federated learning: a survey and prospective. arXiv preprint arXiv:2002.11545 (2020)
24. Berlo, B., Saeed, A., Ozcelebi, T.: Towards federated unsupervised representation learning. In: Proceedings of the Third ACM International Workshop on Edge Systems, Analytics and Networking, pp. 31–36 (2020)
25. Zhuang, W., Gan, X., Wen, Y., Zhang, X., Zhang, S., Yi, S.: Towards unsupervised domain adaptation for deep face recognition under privacy constraints via federated learning. arXiv preprint arXiv:2105.07606 (2021)
26. Zhuang, W., Wen, Y., Zhang, S.: Joint optimization in edge-cloud continuum for federated unsupervised person re-identification. In: Proceedings of the 29th ACM International Conference on Multimedia, pp. 433–441 (2021)
27. Sattler, F., Wiedemann, S., Müller, K.R., Samek, W.: Robust and communication-efficient federated learning from non-IID data. IEEE Trans. Neural Netw. Learn. Syst. **31**(9), 3400–3413 (2019)
28. Bachman, P., Hjelm, R.D., Buchwalter, W.: Learning representations by maximizing mutual information across views. arXiv preprint arXiv:1906.00910 (2019)
29. Tschannen, M., Djolonga, J., Rubenstein, P.K., Gelly, S., Lucic, M.: On mutual information maximization for representation learning. arXiv preprint arXiv:1907.13625 (2019)
30. Cubuk, E.D., Zoph, B., Shlens, J., Le, Q.V.: Randaugment: practical data augmentation with no separate search. CoRR abs/1909.13719 (2019). http://arxiv.org/abs/1909.13719
31. Van Gansbeke, W., Vandenhende, S., Georgoulis, S., Proesmans, M., Van Gool, L.: SCAN: learning to classify images without labels. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12355, pp. 268–285. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58607-2_16
32. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? arXiv preprint arXiv:1411.1792 (2014)
33. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
34. Bucilua, C., Caruana, R., Niculescu-Mizil, A.: Model compression. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA (2006)
35. Yalniz, I.Z., Jégou, H., Chen, K., Paluri, M., Mahajan, D.: Billion-scale semi-supervised learning for image classification. arXiv preprint arXiv:1905.00546 (2019)
36. Xie, Q., Luong, M.T., Hovy, E., Le, Q.V.: Self-training with noisy student improves imagenet classification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10687–10698 (2020)
37. Zhai, X., Oliver, A., Kolesnikov, A., Beyer, L.: S4L: self-supervised semi-supervised learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1476–1485 (2019)
38. Krizhevsky, A.: Learning multiple layers of features from tiny images (2009)

39. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.: Reading digits in natural images with unsupervised feature learning (2011)
40. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: CVPR 2009 (2009)
41. Lee, D.H., et al.: Pseudo-label: the simple and efficient semi-supervised learning method for deep neural networks. In: Workshop on Challenges in Representation Learning, ICML, vol. 3, p. 896 (2013)
42. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training GANs. Adv. Neural. Inf. Process. Syst. **29**, 2234–2242 (2016)
43. French, R.M.: Catastrophic forgetting in connectionist networks. Trends Cogn. Sci. **3**(4), 128–135 (1999). https://doi.org/10.1016/S1364-6613(99)01294-2