Semi-Supervised Deep Coupled Ensemble Learning With Classification Landmark Exploration

Jichang Li, Si Wu, Cheng Liu, Zhiwen Yu, and Hau-San Wong

***Abstract*— Using an ensemble of neural networks with consistency regularization is effective for improving performance and stability of deep learning, compared to the case of a single network. In this paper, we present a semi-supervised Deep Coupled Ensemble (DCE) model, which contributes to ensemble learning and classification landmark exploration for better locating the final decision boundaries in the learnt latent space. First, multiple complementary consistency regularizations are integrated into our DCE model to enable the ensemble members to learn from each other and themselves, such that training experience from different sources can be shared and utilized during training. Second, in view of the possibility of producing incorrect predictions on a number of difficult instances, we adopt class-wise mean feature matching to explore important unlabeled instances as classification landmarks, on which the model predictions are more reliable. Minimizing the weighted conditional entropy on unlabeled data is able to force the final decision boundaries to move away from important training data points, which facilitates semi-supervised learning. Ensemble members could eventually have similar performance due to consistency regularization, and thus only one of these members is needed during the test stage, such that the efficiency of our model is the same as the non-ensemble case. Extensive experimental results demonstrate the superiority of our proposed DCE model over existing state-of-the-art semi-supervised learning methods.**

***Index Terms*— Semi-supervised classification, deep ensemble, consistency regularization, landmark learning.**

I. Introduction

THE development of deep neural networks [1] has led to significant improvement in semi-supervised image classification [2]–[4]. Consistency regularization based methods, such as 'Temporal-Ensembling' [5], have been applied to exploit unlabeled data for alleviating over-fitting. A widely used assumption is that similar instances should have consistent label predictions by the classification model. Minimizing the divergence caused by the perturbation in the input enables the model to learn more abstract invariance, and push the decision boundaries away from high-density regions in the latent space [6]–[8]. Furthermore, adversarial samples [9], [10] generated by including small perturbation to the input in the direction sensitive to the model prediction have led to significant performance gains. In addition, the principle of ensemble learning has also been exploited for different semi-supervised learning tasks, since ensemble models usually achieve better and more stable predictions than the constituent models [11]–[13].

This work aims to improve semi-supervised classification from the aspect of network ensemble. Existing works have verified that network ensemble is useful for boosting semi-supervised learning, but its potential has not been fully explored. We incorporate a complementary regularization approach into standard supervised learning to force the constituent networks in our model to learn from itself and others. In this case, the training experience from different sources can be shared and utilized during training. Consequently, the consensus results can reasonably be expected to yield more accurate estimation of the true labels for unlabeled data in most cases. To further alleviate the influence of incorrect predictions for unlabeled data, instead of treating all unlabeled data equally, the important ones are distinguished through class-wise mean feature matching. These data points are viewed as classification landmarks, on which the model predictions are more reliable. Re-weighting them is useful to make the model generalize well to other unseen instances. Integrating classification landmark exploration into the proposed network ensemble scheme to improve semi-supervised learning is non-trivial and to our knowledge has not been attempted before.

Specifically, we propose a Deep Coupled Ensemble (DCE) model for improving semi-supervised classification. We develop a temporal ensemble network as the constituent network, and adopt two such networks to build our coupled ensemble model. In addition to temporal ensemble predictions on unlabeled data, the constituent networks are able to provide their predictions as training targets for each other. To stabilize model prediction, each constituent network is also required to be robust to adversarial perturbation. Therefore, three corresponding consistency regularization terms are incorporated into standard supervised learning for jointly training network ensemble. Consequently, the constituent networks are

Manuscript received October 5, 2018; revised March 18, 2019 and June 16, 2019; accepted July 26, 2019. Date of publication August 13, 2019; date of current version September 25, 2019. This work was supported in part by the Research Grants Council of the Hong Kong Special Administration Region under Project CityU 11300715, in part by the National Natural Science Foundation of China under Project U1611461, Project 61722205, Project 61751205, and Project 61572199, in part by the City University of Hong Kong under Project 7005055, in part by the Natural Science Foundation of Guangdong Province under Project 2016A030310422 and Project 2017A030312008, in part by Fundamental Research Funds for the Central Universities under Project 2018ZD33, and in part by the National Key R&D Program of China under Grant 2018YFB1700300. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Julian Fierrez. *(Corresponding author: Si Wu.)*

J. Li and Z. Yu are with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China (e-mail: cslijichang@mail.scut.edu.cn; zhwyu@scut.edu.cn).

S. Wu is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China, and also with the Department of Computer Science, City University of Hong Kong, Hong Kong (e-mail: cswusi@scut.edu.cn).

C. Liu is with the Department of Computer Science, Shantou University, Shantou 515063, China (e-mail: chengliu10@gmail.com).

H.-S. Wong is with the Department of Computer Science, City University of Hong Kong, Hong Kong (e-mail: cshswong@cityu.edu.hk).

Digital Object Identifier 10.1109/TIP.2019.2933724

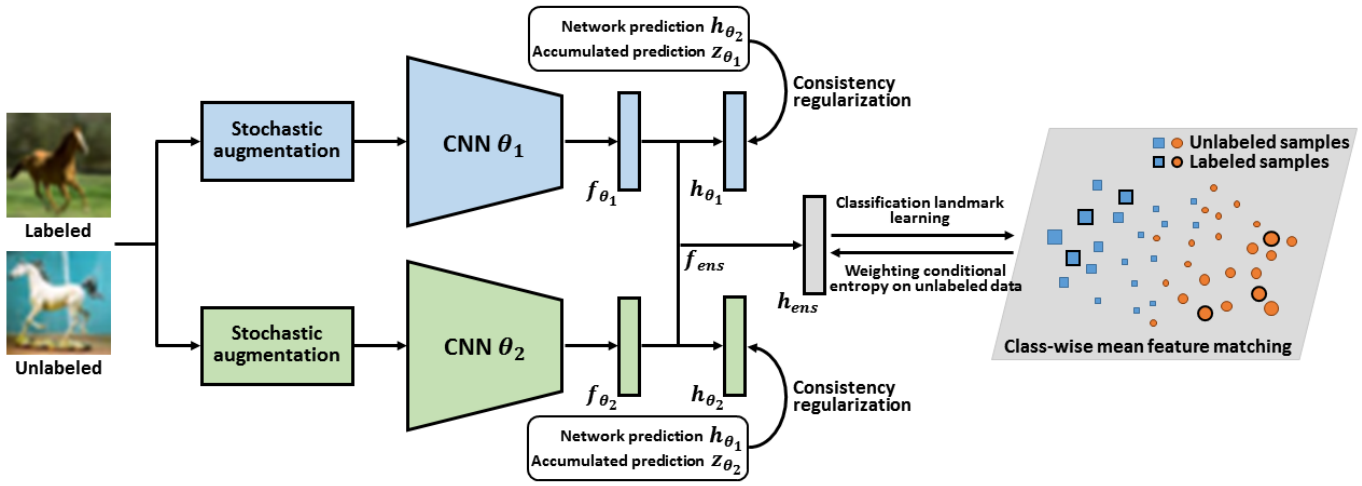


Fig. 1. An overview of the proposed DCE model for improving semi-supervised classification. Our ensemble model consists of two constituent networks having the same architecture but different initializations and dropout. Self-learning and collaborative learning between the networks can be enhanced by integrating multiple complementary consistency regularizations. Furthermore, classification landmark learning is embedded into each mini-batch based update step for re-weighting unlabeled training samples, which alleviates the influence of incorrect predictions on the final decision boundaries. Note that the data points shown in different sizes indicate the unlabeled samples with different weights determined through class-wise mean feature matching with labeled samples.

able to collaboratively learn with each other. In contrast to traditional ensemble models requiring all members during the test stage, our DCE model encourages the constituent networks to produce consistent predictions, and thus only one of them is needed for testing, which ensures the efficiency is as high as those of the single network models having the same network architecture. On the other hand, minimizing the conditional entropy for unlabeled data is able to enforce the network to produce confident predictions on them. This is useful for pushing the decision boundaries to low-density data regions in the latent space. We take into account the reliability of the predictions for unlabeled data, and distinguish the important ones through class-wise mean feature matching with labeled samples. An overview of the proposed DCE model is shown in Fig.1. The effectiveness of the proposed improvement strategies is experimentally verified, and the proposed approach shows state-of-the-art performance on multiple semi-supervised classification benchmarks when compared with recently proposed techniques. The main contributions of this work are as follows:

- We formulate a new coupled ensemble model that integrates complementary consistency regularization to jointly train two constituent networks for improving semi-supervised classification, while the test efficiency is the same as the non-ensemble case.
- To alleviate the influence from incorrect predictions, we embed classification landmark learning in each mini-batch based update step to re-weight the samples, and include a weighted conditional entropy term in the overall loss function to force the final decision boundaries to move away from important unlabeled data points.
- We demonstrate that our proposed improvement strategies facilitate learning on partially labeled data, and lead to better classification performance than existing state-of-the-art semi-supervised learning methods on multiple standard benchmarks.

In Section II, we review recent deep models for semi-supervised learning. In Sections III and IV, we introduce a temporal ensemble network as a constituent network in our DCE model, and present details of the proposed approach, respectively. In Section V, we verify the effectiveness and advantages of the proposed DCE model on multiple standard semi-supervised classification benchmarks. Finally, we conclude this paper in Section VI.

II. Related Works

Different from transfer learning which focuses on applying the knowledge gained from learning on one task to a different but related task, semi-supervised learning aims to make use of a small amount of labeled data with a large amount of unlabeled data on the same task to achieve improvement in learning accuracy over unsupervised learning, while without the costs needed for supervised learning. Semi-supervised learning has attracted extensive studies, in view of its capability to improve generalization performance of the classification models by utilizing a limited amount of labeled instances, and a large amount of unlabeled instances [14]–[20]. We here review the related deep models developed for this purpose. Generative adversarial networks (GANs) [21] have recently achieved promising results when applied to semi-supervised classification tasks [22]–[25]. Springenberg [26] adopted a generative model to learn class conditional distributions. In [27], a semi-supervised GAN was proposed to learn a generative model and a classification model simultaneously. Various techniques for training generative models were investigated for boosting the classification performance of GANs in [24]. To jointly learn the generation and inference networks, Dumoulin *et al.* [28] developed an adversarially learned inference model, which

enables better representation learning. In [29], Li *et al.* proposed a triple generative adversarial network, in which a classifier is added to form a three-players' game, such that the conditional distribution between instances and labels can be characterized. In addition, Gan *et al.* [30] proposed another new GAN framework, triangle generative adversarial network. This network is composed of two generators and two discriminators to learn the bi-directional conditional distributions between the image-label domains.

In addition to the GAN-based methods, a number of recent semi-supervised methods, such as [31], [32], were proposed based on a critical clustering assumption that the decision boundaries should be located in low-density regions [7]. Besides minimizing the estimated conditional entropy on unlabeled data, various consistency regularization techniques have been used in semi-supervised deep models. In [33], the network was updated by minimizing the divergence in different passes of each instance through the network after stochastic transformations and perturbations are applied. Laine and Aila [5] proposed the 'Π-model', which is similar to the 'Γ-model' of the 'Ladder-Network' framework in [34], trained by minimizing a loss function of the divergence between the predictions of the same input under different stochastic augmentation and dropout conditions. Instead of including independent noise, Goodfellow *et al.* [35] proposed adversarial training by slightly changing the input to induce predictions that are as different as possible from the original predictions, which is effective in improving model prediction. Miyato *et al.* [9], [10] proposed virtual adversarial training to carefully determine the required perturbation, and achieve impressive results for semi-supervised learning. From another perspective, Park *et al.* [36] proposed adversarial dropout to maximize the divergence between the ground-truth class label and the network prediction, which was observed to lead to performance improvement.

There are other semi-supervised deep models which are designed by using an ensemble learning mechanism. Implicit ensemble can be implemented by using a single network through 'Dropout' [37], 'DropConnect' [38], stochastic depth [39], 'Swapout' [40], and so on. In these models, a particular section of the network is involved in each iteration of the training process. As a result, the complete network can be considered as an ensemble of trained sub-networks. To encourage ensemble members to learn from each other, a few recent works focus on explicit ensembling of a small number of networks. In [41], He *et al.* proposed a dual learning mechanism, in which two cross-lingual models interactively learn to solve the translation problem from each other, and an additional language model can be used to provide the supervision. Different from the 'Π-model' [5], Tarvainen and Valpola [11] proposed a 'Mean-Teacher' model in which a teacher model is updated by averaging the network parameters of a student model, while the student model is trained by minimizing the divergence between the teacher and itself. Instead of pre-specifying a teacher and a student, Zhang *et al.* [12] adopted a mutual learning mechanism to enable member models to collaboratively teach and learn from each other. Our proposed ensemble model is different from the existing works in the following two aspects. On one hand, we enhance both self-learning and collaborative learning between constituent networks by incorporating complementary consistency regularization, such that training experience from different sources can be shared and exploited during training. On the other hand, we embed classification landmark learning in the network training process to force the final decision boundaries to move away from the important unlabeled data points, which improves the generalization capability of the proposed model.

Also related to this work, reweighting training data has been widely used in learning with label noise [42], [43]. A number of methods, e.g., [44]–[46], were proposed to identify and filter out mislabeled training instances, such that the models can be robust to class-conditional random label noise. Further, Cheng *et al.* [47] proposed a learning model for a more generalized case of instance- and label-dependent noise, along with the theoretical guarantee that a classifier learnt on the collected reliable data will converge to the Bayes optimal classifier under certain conditions. In their model, a kernel mean matching method was adopted to reweight the reliable instances to match the statistics of clean data. The corresponding optimization can be formulated as a quadratic programming problem, which is similar to our work in determining the weights of unlabeled instances. In the proposed model, we minimize the weighted conditional entropy to enforce the model to produce confident predictions on reliable unlabeled instances, without using the predictions to generate pseudo-labels as the noisy version of true labels.

III. Temporal Ensemble Networks

It is critical for semi-supervised learning to exploit a large amount of unlabeled data to facilitate learning on a limited amount of labeled data. Although the true class labels of unlabeled samples are unknown, aggregating the previous prediction over training epochs for each unlabeled sample can yield more accurate estimation of the true label in most cases. Compared to the current predictions, the temporal ensemble predictions are better training targets for unlabeled samples. Reducing the divergence between the current and temporal ensemble predictions for unlabeled samples ensures that the network predictions become more accurate and stable in the training process. Before introducing the proposed DCE model, we present a temporal ensemble network in this section. We will adopt this network as the constituent network in our coupled ensemble framework.

We slightly modify the 'Temporal Ensembling' model [5] by including a conditional entropy term in the overall loss function. A widely used assumption in clustering is that decision boundaries should not cross high-density data regions. Minimizing the conditional entropy for unlabeled data is able to enforce the network to produce confident predictions on them. This is useful for pushing the decision boundaries to low-density data regions in the latent space, thereby improving semi-supervised image classification.

Let $\mathbb{D}_l$ denotes the joint distribution over a set of labeled instances x_l and their corresponding one-hot labels y_l. $\mathbb{D}_u$ is

defined as the distribution over a set of unlabeled instances x_u. The amount of labeled data is usually much smaller than that of unlabeled data. For learning from the partially labeled data, the overall loss function of the temporal ensemble network parameterized by θ consists of the following three terms:

$$\begin{aligned}&\mathcal{L}_{temEns}(\theta; x_l, y_l, x_u)\\&\quad= E_{(x_l,y_l)\sim\mathbb{D}_l}\Big[\ell\big(y_l, h_\theta(x_l)\big)\Big]\\&\qquad+E_{x_u\sim\mathbb{D}_u}\Big[H\big(h_\theta(x_u)\big)\Big]+L_{\mathcal{T}}\big(\theta; x_u, z_\theta(x_u)\big),\end{aligned}\tag{1}$$

where $h_\theta(\cdot)$ denotes the predicted class probability distribution, and $z_\theta(\cdot)$ denotes the temporal ensemble of the predictions over previous training epochs. In Eq.(1), the first term is used to measure the prediction quality on the labeled training samples by adopting the cross entropy function

$$\ell(y_l, h_\theta(x_l)) = -y_l^T \ln h_\theta(x_l),\tag{2}$$

and the conditional entropy function $H(\cdot)$ is used to measure the amount of information needed to describe the class of unlabeled instance x_u according to the prediction as follows:

$$H\big(h_\theta(x_u)\big) = -h_\theta(x_u)^T \ln h_\theta(x_u).\tag{3}$$

The third term in Eq.(1) aims to force the current network prediction to be consistent with the corresponding accumulated result as follows:

$$L_{\mathcal{T}}(\theta; x_u, z_\theta(x_u)) = w(t)E_{x_u\sim\mathbb{D}_u}\Big[d\big(z_\theta(x_u), h_\theta(x_u)\big)\Big],\tag{4}$$

where the weighting factor $w(t)$ is defined as a function of the number of training iterations t, and the setting is described in Section V-A. The function $d(\cdot,\cdot)$ is used to measure the difference between $z_\theta(x_u)$ and $h_\theta(x_u)$ as follows:

$$d(z_\theta(x_u), h_\theta(x_u)) = \|z_\theta(x_u) - h_\theta(x_u)\|^2.\tag{5}$$

In each training epoch, the current prediction of the class probability distribution for unlabeled training sample x_u is incorporated into the corresponding temporal ensemble output $z_\theta(x_u)$. Specifically, $z_\theta(x_u)$ is updated as follows:

$$z_\theta^{(t+1)}(x_u) \leftarrow \rho z_\theta^{(t)}(x_u) + (1-\rho)h_\theta(x_u),\tag{6}$$

where the momentum coefficient ρ is a non-negative constant for controlling the effect of ensembling during training. In the training process, aggregating the previous predictions can lead to more stable and accurate classification on the unlabeled training samples in most cases, and the results can be used as soft targets to regularize label prediction, such that the network can be trained under self-supervision.

IV. Proposed Approach

To exploit the difference between separate networks for facilitating the estimation of the final decision boundaries, we propose a coupled ensemble model in which the constituent networks collaborate to teach and learn from each other, and important unlabeled samples are gradually identified during training, such that the final decision boundaries can be pushed away from reliable unlabeled data points in the latent feature space. In this section, we present the proposed ensemble model followed by a classification landmark learning module, and then provide the implementation details.

A. Coupled Ensemble Framework

To better balance classification performance and computational efficiency, our DCE model is composed of two temporal ensemble networks parameterized by θ_1 and θ_2, respectively. A max-pooling layer following the two constituent networks produces consensus predictions by max-pooling the network outputs. The overall loss function of our ensemble model consists of multiple corresponding terms as follows:

$$\begin{aligned}&\mathcal{L}_{couEns}(\theta_1, \theta_2; x_l, y_l, x_u)\\&\quad= L_{\mathcal{E}}(\theta_1, \theta_2; x_l, y_l, x_u)\\&\qquad+L_{\mathcal{T}}\big(\theta_1; x_u, z_{\theta_1}(x_u)\big) + L_{\mathcal{T}}(\theta_2; x_u, z_{\theta_2}(x_u))\\&\qquad+L_{\mathcal{N}}(\theta_1, \theta_2; x_u) + L_{\mathcal{A}}(\theta_1; x_l, x_u) + L_{\mathcal{A}}(\theta_2; x_l, x_u),\end{aligned}\tag{7}$$

where the ensemble term $L_{\mathcal{E}}$ assesses the consensus predictions, the temporal consistency term $L_{\mathcal{T}}$ enforces the current predictions to be consistent with the accumulated predictions, the network consistency term $L_{\mathcal{N}}$ enables learning and teaching between constituent networks, and the adversarial training term $L_{\mathcal{A}}$ smooths the model with respect to input perturbations. Specifically, we define the ensemble term as follows:

$$\begin{aligned}L_{\mathcal{E}}(\theta_1, \theta_2; x_l, y_l, x_u) = E_{(x_l,y_l)\sim\mathbb{D}_l}&\Big[\ell\big(y_l, h_{ens}(x_l)\big)\Big]\\&+E_{x_u\sim\mathbb{D}_u}\Big[\varphi(x_u)H\big(h_{ens}(x_u)\big)\Big],\end{aligned}\tag{8}$$

where $\varphi(x_u)$ denotes the coefficient associated with sample x_u, and is defined in the next subsection. The consensus result $h_{ens}(\cdot)$ can be computed by normalizing the max-pooling results of the constituent network outputs as follows:

$$h_{ens}(x) = N\big(\texttt{max-pool}(f_{\theta_1}(x), f_{\theta_2}(x))\big),\tag{9}$$

where $N(\cdot)$ denotes the softmax function for mapping an input to the valid label space, and $f_{\theta_1}(\cdot)$ ($f_{\theta_2}(\cdot)$) denotes the input of the softmax layer of the network θ_1 (θ_2). Through minimizing the weighted conditional entropy with respect to the posterior distribution in Eq.(8), the model tends to be confident on the unlabeled samples having large weights, such that the decision boundaries will be located far away from them.

For sharing training experience between the networks, the network consistency term is defined as follows:

$$L_{\mathcal{N}}(\theta_1, \theta_2; x_u) = \nu(t)E_{x_u\sim\mathbb{D}_u}\Big[d\big(h_{\theta_1}(x_u), h_{\theta_2}(x_u)\big)\Big].\tag{10}$$

Inspired by [48], this term aims to align each network's prediction with the output of the other network, such that more information can be obtained. Specifically, there are a number of random factors in augmentation, sampling, initialization and dropout influencing network prediction, especially in the early training stage. The class probabilities predicted by the constituent networks are in general different, in particular for the probabilities of the next most likely classes. Prediction consistency regularization between the constituent networks is helpful to facilitate collaborative learning. Consequently, the proposed DCE model performs more accurate classification as the constituent networks mutually reinforce each other.

As reported in [8], the classification model may abruptly change its output in the neighborhood of unlabeled data

points if there is no locally-Lipschitz constraint. To stabilize the estimation of conditional entropy on the unlabeled data, the adversarial training term defined below is included in the overall loss function:

$$\begin{aligned} &L_{\mathcal{A}}(\theta_1; x_l, x_u) \\ &\quad = E_{x_l \sim \mathbb{D}_l}\Big[\max_{\|r\| \le \epsilon} \mathcal{D}\big(h_{\theta_1}(x_l) \| h_{\theta_1}(x_l + r)\big)\Big] \\ &\qquad + E_{x_u \sim \mathbb{D}_u}\Big[\max_{\|r\| \le \epsilon} \mathcal{D}\big(h_{\theta_1}(x_u) \| h_{\theta_1}(x_u + r)\big)\Big], \end{aligned} \tag{11}$$

where ϵ denotes a hyper-parameter controlling the intensity of the adversarial perturbation r, and the Kullback-Leibler (KL) divergence $\mathcal{D}(\cdot\|\cdot)$ is used to measure the prediction divergence with respect to the network θ_1 for cases with and without perturbation as follows:

$$\mathcal{D}\big(h_{\theta_1}(x_u) \| h_{\theta_1}(x_u + r)\big) = -h_{\theta_1}(x_u)^T \ln \frac{h_{\theta_1}(x_u + r)}{h_{\theta_1}(x_u)}. \tag{12}$$

In addition, the adversarial training term $L_{\mathcal{A}}(\theta_2; x_l, x_u)$ can be similarly defined.

B. Classification Landmark Learning

Minimization of conditional entropy enforces the network to be confident on unlabeled training samples, which is useful for moving the decision boundaries in low-density regions. However, it is common for semi-supervised classification models to produce incorrect predictions on a number of difficult training samples. To alleviate the influence of incorrect predictions on the final decision boundaries, we adopt a weighted conditional entropy with respect to the posterior distribution. Inspired by [49], important unlabeled instances can be exploited to facilitate our task. We determine the weights of unlabeled training samples based on the assumption that the class-wise distributions of the labeled data and unlabeled data should be the same, such that the unlabeled data points close to the specific class centers should have larger weights. The unlabeled instances with large weights are considered as classification landmarks.

Since both labeled and unlabeled data are produced from the same domain, the class-wise distributions over labeled and unlabeled training samples should also be the same. As a result, the mean position of the labeled data points should coincide with that of the unlabeled data points for each class in the learnt feature space, which can be expressed as follows:

$$E_{x_l \sim \mathbb{D}_l^c}\big[f_{ens}(x_l)\big] = E_{x_u \sim \mathbb{D}_u^c}\big[f_{ens}(x_u)\big], \tag{13}$$

where $c \in \{1, 2, \ldots, C\}$ denotes the class index, $\mathbb{D}_l^c$ ($\mathbb{D}_u^c$) denotes the distribution over the labeled (unlabeled) training samples x_l (x_u) belonging to the c-th class, and $f_{ens}(\cdot)$ denotes the consensus representation for sample x as follows:

$$f_{ens}(x) = \texttt{max-pool}\big(f_{\theta_1}(x), f_{\theta_2}(x)\big). \tag{14}$$

Eq.(13) indicates that class-wise mean feature matching is useful for aligning the posterior distributions. For the unlabeled data points having the same prediction and close to the class center (the mean of the labeled data points belonging to the predicted class), the network outputs should be more reliable, and those data points should play a more important role in the training process. In contrast, the unlabeled data points distant to the corresponding class centers have higher risks of misclassification, and should be less involved during training, because labeled data points are scarce in their neighborhoods. For this purpose, we determine the weights of unlabeled instances in each mini-batch by solving the following quadratic programming problem:

$$\begin{aligned} &\min_{\alpha_{x_u}} \left\| \frac{1}{|B_l^c|} \sum_{x_l \in B_l^c} f_{ens}(x_l) - \sum_{x_u \in B_u^c} \alpha_{x_u} f_{ens}(x_u) \right\|^2, \\ &s.t. \sum_{x_u \in B_u^c} \alpha_{x_u} = 1, \\ &\quad \alpha_{x_u} \in [0, 1],\ x_u \in B_u^c, \end{aligned} \tag{15}$$

where $|\cdot|$ denotes the cardinality of a sample set, B_l^c (B_u^c) denotes the set of labeled (unlabeled) training samples belonging (classified) to class c in a mini-batch, and α_{x_u} denotes the weight of unlabeled instance x_u. In the training stage, we compute the class centers using the samples in a mini-batch. We can also use moving historical averages to make them more stable.

With the above formulation, suitable quadratic programming algorithms, such as the interior point method [50], can be applied to efficiently solve this problem. It is noted that the computational requirement of the optimization process is limited by the size of a mini-batch. According to the resulting solution, the conditional entropy coefficient in Eq.(8) can be computed as follows:

$$\varphi(x_u) = 1 + \alpha_{x_u}. \tag{16}$$

Re-weighting unlabeled data as in Eq.(16) aims to make the classification landmarks play a more important role than others in the process of determining the final decision boundaries. In this case, minimization of the weighted conditional entropy forces the final decision boundaries to move away from reliable unlabeled data points in the latent feature space, which facilitates semi-supervised learning.

C. Implementation

1) Adversarial Perturbation: We include the adversarial training term $L_{\mathcal{A}}$ in Eq.(11) to enhance locally-Lipschitz continuity of the network. In practice, we first determine the direction of adversarial perturbation r_u^* which can maximally change the output distribution as follows:

$$r_u^* = \arg\max_{\|r\| \le \epsilon} \mathcal{D}\big(h_{\theta_1}(x_u) \| h_{\theta_1}(x_u + r)\big), \tag{18}$$

where r denotes a perturbation, and it has the same number of dimensions as the input sample x_u. According to [10], r_u^* can be determined by computing the gradient of $\mathcal{D}$ with respect to r on $r = \xi d_u$ for each input data point x_u as follows:

$$\begin{aligned} &v_u \leftarrow \nabla_r D\big(h_{\theta_1}(x_u) \| h_{\theta_1}(x_u + r)\big)|_{r = \xi d_u}, \\ &r_u^* \leftarrow \epsilon \frac{v_u}{\|v_u\|}, \end{aligned} \tag{19}$$

where d_u is a random unit vector initialized by independent and identically Gaussian-distributed values, and ξ denotes a

small magnitude. $\nabla_r \mathcal{D}$ can be directly computed through back-propagation for the network. Once the adversarial perturbation is determined, the adversarial training term can be obtained by computing the divergence between the outputs of the original sample and the perturbed one as follows:

$$L_{\mathcal{A}}(\theta_1; x_l, x_u) = E_{x_l \sim \mathbb{D}_l}\Big[\mathcal{D}\big(h_{\theta_1}(x_l) \| h_{\theta_1}(x_l + r_l^*)\big)\Big] + E_{x_u \sim \mathbb{D}_u}\Big[\mathcal{D}\big(h_{\theta_1}(x_u) \| h_{\theta_1}(x_u + r_u^*)\big)\Big]. \quad (20)$$

Consequently, it is straightforward to minimize $L_{\mathcal{A}}$ for training the network in this case.

2) Model Training: The proposed DCE model consists of two separate networks, which have the same architecture but different initializations and dropout. These two networks are jointly trained. To avoid unlabeled data dominating the overall loss and incurring performance degradation, each mini-batch consists of both labeled and unlabeled training samples randomly drawn according to a fixed ratio. The weights of unlabeled training samples and the network parameters are updated alternately. Specifically, the quadratic programming problem in Eq.(15) is solved in each mini-batch based update step. Based on the resulting solution, the network parameters are updated using stochastic gradient descent. The terms in Eq.(7) are designed for each specific network except the ensemble term and network consistency term. The back-propagated signal from the ensemble term is used to update both networks. The details of the optimization process are summarized in Algorithm 1.

Algorithm 1 Pseudo-Code of the Proposed DCE Model for Semi-Supervised Classification

Input: Labeled sample set $X_l = \{x_l\}$ and unlabeled sample set $X_u = \{x_u\}$.
Initialize: Constituent networks θ_1 and θ_2 with different initial conditions, learning rate $\varsigma(t)$, and temporal ensemble predictions z_{θ_1} and z_{θ_2}.
for $t = 1$ to T **do**
  for each mini-batch B **do**
    Apply networks θ_1 and θ_2 to compute the representations f_{θ_1} and f_{θ_2}, and predictions h_{θ_1} and h_{θ_2} on labeled (unlabeled) sample x_l (x_u), respectively.
    Compute the consensus representation f_{ens} and prediction h_{ens} according to Eq.(14) and Eq.(9), respectively.
    Apply Interior Point Method to solve the optimization problem in Eq.(15).
    Compute the conditional entropy coefficient $\varphi(x_u)$ according to Eq.(16).
    Apply stochastic gradient descent and update θ_1 and θ_2 using ADAM:

$$\theta_1^{(t+1)} \leftarrow Adam\big(\nabla_{\theta_1}\mathcal{L}_{couEns}, \theta_1^{(t)}, \varsigma(t)\big),$$
$$\theta_2^{(t+1)} \leftarrow Adam\big(\nabla_{\theta_2}\mathcal{L}_{couEns}, \theta_2^{(t)}, \varsigma(t)\big).$$

    Accumulate the current predictions:

$$z_{\theta_1}^{(t+1)}(x_u) \leftarrow \rho z_{\theta_1}^{(t)}(x_u) + (1-\rho)h_{\theta_1}(x_u),$$
$$z_{\theta_2}^{(t+1)}(x_u) \leftarrow \rho z_{\theta_2}^{(t)}(x_u) + (1-\rho)h_{\theta_2}(x_u). \quad (17)$$

  end for
end for
Return θ_1 and θ_2.

V. Experiments and Discussion

In this section, we evaluate the proposed coupled ensemble approach on six standard semi-supervised classification benchmarks: MNIST [51], SVHN [52], CIFAR-10 [53], CIFAR-100 [53], Scene-15 [54], and FaceScrub-100 [55]. Specifically, we perform extensive experiments to determine the extent to which performance is improved, and investigate the contribution of each of our improvement strategies to better understand their roles. As will be demonstrated by the experiment in Section V-E, the two constituent networks in our model eventually have very similar performance during the training stage, and thus there is no need to combine their predictions. Either one of the constituent networks is sufficient for the test stage, and the efficiency of the proposed approach is as high as the main competing methods. To ensure fair comparison with existing state-of-the-art methods, we adopt the same evaluation protocol as in [5] for all the experiments, and report the mean and standard deviation of the test error rates of 10 runs using different random sampling of labeled training samples.

A. Experimental Setting

In all of our experiments, we adopt a relatively small CNN architecture composed of 9 convolutional layers, 3 pooling layers and 1 fully connected layer. TABLE I describes the architecture of the convolutional neural network used. This network architecture had also been used in state-of-the-art methods [5], [9], [36]. A single constituent network has about 3.1M parameters. Although the images of Scene-15 and FaceScrub-100 have higher resolutions than those of other datasets, we slightly modify the network architectures without significantly increasing the number of model parameters. When the proposed network ensemble consists of k constituent networks, the overall parameters of our ensemble model is about k times of that of a single constituent network. The proposed approach is implemented using Python with Tensorflow, and the hardware includes an Intel Core-i7 CPU and a NVIDIA 2080 Ti GPU. In the case of $k = 2$, training our ensemble model in 300 epochs on the test datasets except Scene-15 takes about 34 hours on average. For Scene-15, the corresponding training process takes about 16 hours. The training time of our ensemble model ($k = 2$) is about 1.7 times of that of a single constituent network. This is due to the reason that both constituent networks are evaluated per input along with collaborative learning between them in each epoch. When assigning an exclusive GPU to each constituent network, the whole training process is expected to be speeded up.

For all the experiments, the hyperparameter setting is similar to [36]. In order to improve the generalization performance of the constituent networks, we perform stochastic augmentation by applying random perturbations (Gaussian noise with

TABLE I

THE CONVOLUTIONAL NEURAL NETWORK ARCHITECTURE OF EACH CONSTITUENT NETWORK IN OUR DCE MODEL. WE USE THIS ARCHITECTURE IN ALL THE EXPERIMENTS

Name	Description
Input	$32 \times 32 \times 3$ image ($28 \times 28 \times 1$ for MNIST, $64 \times 64 \times 3$ for FaceScrub-100, $128 \times 128 \times 1$ for Scene-15)
Perturbation	Gaussian noise $\sigma = 0.15$
Convolution	128 filters, 3×3, pad='same', LReLU($\alpha = 0.1$)
Convolution	128 filters, 3×3, pad='same', LReLU($\alpha = 0.1$)
Convolution	128 filters, 3×3, pad='same', LReLU($\alpha = 0.1$)
Pooling	Maxpool 2×2 pixels
Dropout	Dropout, $p = 0.5$
Convolution	256 filters, 3×3, pad='same', LReLU($\alpha = 0.1$)
Convolution	256 filters, 3×3, pad='same', LReLU($\alpha = 0.1$)
Convolution	256 filters, 3×3, pad='same', LReLU($\alpha = 0.1$)
Pooling	Maxpool 2×2 pixels
Dropout	Dropout, $p = 0.5$
Convolution	512 filters, 3×3, pad='valid', LReLU($\alpha = 0.1$)
Convolution	256 filters, 1×1, LReLU($\alpha = 0.1$)
Convolution	128 filters, 1×1, LReLU($\alpha = 0.1$)
Pooling	Global average pool
Dense	Fully connected $128 \rightarrow 10$ Softmax ($128 \rightarrow 100$ for CIFAR-100 and FaceScrub-100, $128 \rightarrow 15$ for Scene-15)

zero-mean and standard deviation 0.15) to each input and encourage the networks to produce robust prediction. The total number of epochs is set to 300, and the size of mini-batch is set to 160: 32 labeled samples / 128 unlabeled samples (15:5/10 for Scene-15 and 40:16/24 for FaceScrub-100). The ensemble model was trained using the ADAM optimizer [56] with the maximum learning rate of 0.02 for CIFAR-10 (0.002 for MNIST, 0.003 for SVHN, Scene-15 and FaceScrub-100, and 0.02 for CIFAR-100) and momentum parameters of $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The learning rate is ramped up in the first 80 epochs using a Gaussian function $exp[-5(1-\tau)^2]$, where $\tau \in [0, 1]$. During the last 50 epochs, the learning rate and β_1 are ramped down to 0 and 0.5, respectively. The weighting factors $w(t)$ and $v(t)$ in the definitions of the temporal consistency term $L_{\mathcal{T}}$ in Eq.(4) and the network consistency term $L_{\mathcal{N}}$ in Eq.(10) varies over training iterations, respectively. The settings of $w(t)$ and $v(t)$ are similar to the learning rate. They have the same ramp-up periods with a maximum value of 25, and then remain unchanged. In Eq.(11), ϵ denotes a hyper-parameter controlling the intensity of the adversarial perturbation. The value of ϵ is set to 10^{-6} in all the experiments, which is the same as [10]. In addition, the momentum parameter ρ in Eq.(17) for updating temporal ensemble predictions is set to 0.6.

B. Model Variants

To investigate the effectiveness of the improvement strategies used in the proposed DCE model, we first train the proposed coupled ensemble model only on the labeled data, and use the result of this 'Supervised-only' model as a lower bound. We implement the temporal ensemble network having the same architecture as the constituent networks of our model as 'Baseline'. We build a variant by incorporating the locally-Lipschitz constraint and classification landmark learning module into the training process of a baseline network, and the resulting model is denoted by 'Baseline w/ AT & LL'. In addition, we remove the network consistency and adversarial training terms in the overall loss function in Eq.(7) to obtain the variants 'DCE w/o NC' and 'DCE w/o AT', respectively. We also train the proposed model without classification landmark learning as another variant 'DCE w/o LL', such that the conditional entropy coefficient $\varphi(x_u) = 1$ in Eq.(16) for all unlabeled training samples. Furthermore, we built the variant 'DCE w/o AT & LL' to investigate the performance of the network ensemble when increasing the number of constituent network. In addition, we substitute average-pooling for max-pooling in Eqs.(9) and (14) to compute consensus representations and predictions, and the resulting model is referred to as 'DCE (`avg-pool`)'. These variants are summarized in TABLE II.

TABLE II

THE VARIANTS OF THE PROPOSED COUPLED ENSEMBLE MODEL FOR INVESTIGATING THE CONTRIBUTION OF EACH IMPROVEMENT STRATEGY USED IN OUR DCE MODEL

Name	Description
Baseline	The constituent network in the proposed model (temporal ensemble network).
Baseline w/ AT & LL	Incorporate the adversarial training term $L_{\mathcal{A}}$ and classification landmark learning.
Supervised-only	The proposed DCE model trained only on the labeled data.
DCE w/o NC	Remove the network consistency term $L_{\mathcal{N}}$ in Eq.(7).
DCE w/o AT	Remove the adversarial training term $L_{\mathcal{A}}$ in Eq.(7).
DCE w/o LL	Disable classification landmark learning ($\varphi(x_u) = 1$ for all unlabeled samples).
DCE w/o AT & LL	Remove the adversarial training term $L_{\mathcal{A}}$ in Eq.(7), and disable classification landmark learning.
DCE (`avg-pool`)	Substitute average-pooling for max-pooling in Eqs.(9) and (14).

C. Comparison With State-of-the-Art Methods

We conduct semi-supervised classification experiments on MNIST, SVHN, CIFAR-10, CIFAR-100, Scene-15 and FaceScrub-100, since existing state-of-the-art methods mostly focus on these benchmarks. We use the same backbone architecture as the main competing methods including 'Temporal-Ensembling' [5], 'Mean-Teacher' [11], 'VAT' [10], 'VAdD' [36] and 'SNTG' [58]. The test error rates of the proposed DCE model and competing methods are shown in TABLES III-VI.

The MNIST dataset contains 10 classes of 60,000 28×28 training images of small cropped handwritten digits. We test the proposed model for the cases where there are 50, 100 and 200 labeled training samples given and the remaining training samples are unlabeled. This benchmark is comparatively simple, and the competing methods had achieved very low test error rates, e.g., 'SNTG + Π-model' (0.66% in the case of 100 labels). The proposed DCE model achieves lower test error rates in all the cases.

Compared to MNIST, SVHN is more challenging because this dataset contains 73,257 32×32 RGB images of real-world house numbers. Our model improves 'Baseline', and reduces the test error rates to 3.20%, 2.92% and 2.88%, which are

TABLE III

COMPARISON OF THE PROPOSED DCE MODEL AND COMPETING METHODS ON THE MNIST DATASET FOR THE GIVEN NUMBERS OF LABELED TRAINING SAMPLES

Method	Test error rate (%) with # labels			
	50 labels	100 labels	200 labels	60,000 labels (All)
Ladder-Network [34]	-	1.06±0.37	-	0.57±0.02
ImprovedGAN [24]	2.21±1.36	0.93±0.07	0.90±0.04	-
TripleGAN [29]	1.56±0.72	0.91±0.58	-	-
SPCTN [57]	1.72±0.13	1.00±0.11	0.86±0.06	-
Π-model [5]	1.02±0.37	0.89±0.15	-	-
SNTG+Π-model [58]	0.94±0.42	0.66±0.07	-	-
CT-GAN [25]	-	0.89±0.13	-	-
Baseline	2.19±0.12	1.76±0.09	0.78±0.17	0.39±0.06
DCE	**0.70±0.10**	**0.59±0.08**	**0.41±0.08**	**0.18±0.06**

TABLE IV

COMPARISON OF THE PROPOSED DCE MODEL AND COMPETING METHODS ON THE SVHN DATASET FOR THE GIVEN NUMBERS OF LABELED TRAINING SAMPLES

Method	Test error rate (%) with # labels			
	500 labels	1,000 labels	2,000 labels	73,257 labels (All)
ImprovedGAN [24]	18.44±4.80	8.11±1.30	6.16±0.58	-
ALI [28]	-	7.42±0.65	-	-
TripleGAN [29]	-	5.77±0.17	-	-
SPCTN [57]	9.79±1.24	7.37±0.30	5.88±0.23	-
Π-model [5]	6.83±0.66	4.95±0.26	-	2.50±0.07
Temporal-Ensembling [5]	5.12±0.13	4.42±0.16	-	2.74±0.06
Mean-Teacher [11]	4.18±0.27	3.95±0.19	-	2.50±0.05
VAT [10]	-	3.74±0.09	-	2.69±0.04
VAdD [36]	-	4.16±0.08	-	2.31±0.01
VAdD+VAT [36]	-	3.55±0.05	-	2.23±0.03
SNTG+Π-model [58]	4.52±0.30	3.82±0.25	-	2.42±0.05
SNTG+VAT [58]	-	3.83±0.22	-	-
Baseline	6.33±0.23	5.36±0.17	4.42±0.13	2.74±0.12
DCE	**3.20±0.07**	**2.92±0.08**	**2.88±0.07**	**2.20±0.02**

TABLE V

COMPARISON OF THE PROPOSED DCE MODEL AND COMPETING METHODS ON THE CIFAR-10 DATASET FOR THE GIVEN NUMBERS OF LABELED TRAINING SAMPLES

Method	Test error rate (%) with # labels			
	1,000 labels	2,000 labels	4,000 labels	50,000 labels (All)
Ladder-Network [34]	-	-	20.40±0.47	-
ImprovedGAN [24]	-	19.61±2.09	18.63±2.32	-
ALI [28]	-	-	17.99±1.62	-
TripleGAN [29]	-	-	16.99±0.36	-
SPCTN [57]	-	17.99±0.50	14.17±0.27	-
Π-model [5]	27.36±1.20	18.02±0.60	13.20±0.27	6.06±0.11
Temporal-Ensembling [5]	-	-	12.16±0.31	5.60±0.10
Mean-Teacher [11]	21.55±1.48	15.73±0.31	12.31±0.28	5.94±0.15
VAT [10]	-	-	11.96±0.10	5.65±0.17
VAdD [36]	-	-	11.68±0.19	5.27±0.10
VAdD+VAT [36]	-	-	10.07±0.11	**4.40±0.12**
SNTG+Π-model [58]	21.23±1.27	14.65±0.31	11.00±0.13	5.19±0.14
SNTG+VAT [58]	-	-	9.89±0.34	-
CT-GAN [25]	-	-	9.98±0.21	-
Baseline	23.19±0.38	17.96±0.29	12.13±0.26	6.02±0.15
DCE	**16.53±0.14**	**12.34±0.16**	**9.84±0.07**	4.69±0.12

lower than those of competing methods for the cases where there are 500, 1,000 and 2,000 labels given, respectively.

In addition, our DCE model achieves significant improvement over 'Baseline' on the CIFAR-10 dataset containing 50,000 32×32 RGB images from 10 object classes. For the cases of 1,000, 2,000 and 4,000 labels given, we observe an improvement over 'Baseline' by about 7, 6 and 2 percentage points by our DCE model, from 23.19%, 17.96% and 12.13%

TABLE VI

COMPARISON OF THE PROPOSED DCE MODEL AND COMPETING METHODS ON THE SCENE-15, CIFAR-100 AND FACESCRUB-100 DATASETS FOR THE GIVEN NUMBERS OF LABELED TRAINING SAMPLES. * INDICATES OUR IMPLEMENTATION

	Scene-15		CIFAR-100		FaceScrub-100	
Method	150 labels	1,500 labels (All)	10,000 labels	50,000 labels (All)	2,000 labels	12,915 labels (All)
Π-model [5]	55.63±0.57*	28.42±0.49*	39.19±0.36	26.32±0.04	28.79±1.08*	**5.36±0.46***
Temporal-Ensembling [5]	62.32±0.71*	30.63±0.51*	38.65±0.51	26.30±0.15	23.50±0.93*	5.93±0.48*
SNTG+Π-model [58]	-	-	37.97±0.29	-	-	-
Baseline	52.00±0.74	28.00±0.55	43.40±0.18	27.67±0.16	21.56±0.73	5.58±0.43
DCE	**46.63±0.56**	**24.80±0.35**	**36.75±0.15**	**25.62±0.13**	**16.06±0.57**	5.63±0.41

TABLE VII

ABLATION STUDIES FOR THE IMPROVEMENT STRATEGIES USED IN THE PROPOSED DCE MODEL ON THE MNIST, SVHN, CIFAR-10 AND CIFAR-100 DATASETS. THE MODEL VARIANTS ARE DESCRIBED IN TABLE II

	MNIST	SVHN	CIFAR-10	CIFAR-100
Method	100 labels	1,000 labels	4,000 labels	10,000 labels
Supervised-only	4.23±0.12	9.33±0.17	17.48±0.17	44.10±0.24
Baseline	1.76±0.09	5.36±0.17	12.13±0.26	43.40±0.18
Baseline w/ AT & LL	1.30±0.11	3.96±0.21	11.33±0.32	39.68±0.35
DCE (2 Nets) w/o AT & LL	1.01±0.14	3.93±0.08	11.95±0.20	38.94±0.41
DCE (3 Nets) w/o AT & LL	0.52±0.08	3.72±0.10	11.71±0.26	37.44±0.32
DCE (4 Nets) w/o AT & LL	0.39±0.05	3.64±0.10	11.65±0.32	36.99±0.27
DCE (2 Nets) w/o NC	1.08±0.07	3.22±0.15	10.50±0.23	38.10±0.45
DCE (2 Nets) w/o AT	0.88±0.05	3.66±0.14	11.52±0.19	37.57±0.41
DCE (2 Nets) w/o LL	0.77±0.15	3.84±0.15	11.01±0.17	38.70±0.39
DCE (2 Nets) (`avg-pool`)	1.84±0.13	3.92±0.18	11.13±0.15	36.91±0.38
DCE (2 Nets)	**0.70±0.10**	**2.92±0.08**	**9.84±0.07**	**36.75±0.15**

to 16.53%, 12.34% and 9.84%, respectively. The proposed DCE model surpasses the competing methods for all the different labeled sample sizes. When all labels are used for fully supervised training, our DCE model approximately matches the state-of-the-art error rate.

Similar to CIFAR-10, CIFAR-100 contains 50,000 color natural images of size 32×32. However, CIFAR-100 is more challenging than CIFAR-10 due to the reason that there are 100 categories. We slightly modify the network architecture by replacing the final layer with a 100-class classification layer. Following [5], we randomly sample 10,000 training images, and provide the corresponding class labels in the experiment. The same data augmentation is performed as the experiment on CIFAR-10. TABLE VI compares the DCE model with the baseline model and the existing state-of-the-art methods, which reported the classification performance on this benchmark. Our DCE model significantly outperforms 'Baseline', and reduces the test error rate to 36.75%, which is lower than those of other methods. This results demonstrate the effectiveness of the proposed model in coping with the increase in the number of categories for semi-supervised classification.

We also conduct experiments on Scene-15 and FaceScrub-100. Scene-15 is a comparatively small dataset, which consists of 15 scene categories with a few hundred gray images per class. FaceScrub is a human face dataset. We construct FaceScrub-100 by selecting the 100 largest classes from the original dataset. In the experiment, all the images of Scene-15 and FaceScrub-100 are resized to 128×128 and 64×64, respectively. We slightly modify the network architectures in TABLE I without significantly increasing the number of model parameters. The results of the proposed approach on these two new datasets are shown in TABLE VI. Similar to the results achieved on other datasets, the proposed approach improves the baseline by about 5 and 6 percentage points on Scene-15 with 150 labels and FaceScrub-100 with 2000 labels, respectively.

D. Analysis of Improvement Strategies

The foregoing comparisons with 'Baseline' demonstrate the effectiveness of the complete coupled ensemble model. To assess the relative contributions of the improvement strategies used in our DCE model, we conduct a number of ablation studies on MNIST, SVHN, CIFAR-10 and CIFAR-100 in this subsection. We intend to focus on comparing the performance gains using the proposed ensemble learning mechanism and classification landmark learning. Specifically, we compare the DCE model with its variants including 'Supervised-only', 'DCE w/o NC', 'DCE w/o AT', 'DCE w/o LL', 'DCE w/o AT & LL' and 'DCE (`avg-pool`)' described in TABLE II, and the results are shown in TABLE VII.

1) Exploiting Unlabeled Data: Compared to the 'Supervised-only' model in the four datasets, the test error rates are reduced by about 4, 6, 8 and 7 percentage points when utilizing the unlabeled training samples in our

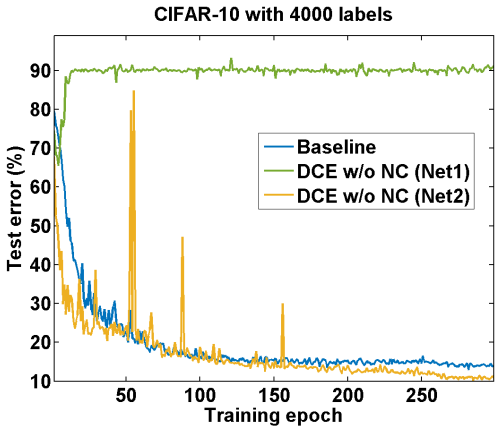


Fig. 2. Comparing ensemble member behaviors without network consistency regularization on CIFAR-10 with 4,000 labels. Only one constituent network in the 'DCE w/o NC' model can be trained normally, which indicates that the network consistency term in Eq.(7) plays an important role in stabilizing and regularizing the label prediction of each constituent network in our ensemble model.

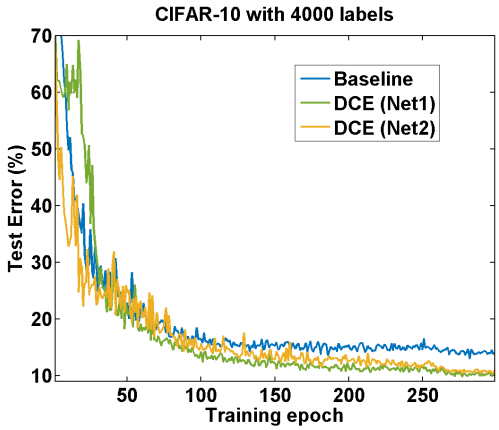


Fig. 3. Comparison of the baseline model and the proposed DCE model on CIFAR-10 with 4000 labels. The proposed ensemble learning mechanism and consistency regularization ensure that the two constituent networks eventually have very similar performance.

model. We consider that the proposed model is able to exploit unlabeled data for significantly improving the generalization capability of the constituent networks.

2) Number of Constituent Networks: We can extend the proposed ensemble scheme to the cases of multiple constituent networks by re-defining the network consistency term $L_{\mathcal{N}}$ in Eq.(10) as the divergence between the individual prediction and the consensus result h_{ens}. In TABLE VII, we provide the performances of the model variants 'DEC (k Nets) w/o AT & LL' ($k = 2$, 3 and 4) having different number of constituent networks. We can clearly observe that 'DCE (2 Nets) w/o AT & LL' significantly outperforms 'Baseline' in all cases, due to the reason that the proposed network ensemble scheme is effective in producing more accurate training targets for unlabeled data. We can also observe that the performance of our ensemble can be improved when increasing the number of constituent networks, but the relative improvement becomes smaller. This observation suggests that using two constituent networks is able to attain a balance between classification performance and computation efficiency.

3) Network Consistency Regularization: 'Baseline w/ AT & LL' surpasses 'Baseline', but underperforms the variant 'DCE (2 Nets)' significantly. The network consistency term in Eq.(7) enables the two constituent networks of our ensemble to learn from each other. When we remove this term, the classification accuracies of the resulting model 'DCE w/o NC' are also much lower than those of the DCE model. Fig.2 shows the training curves of the constituent networks, 'DCE w/o NC (Net1)' and 'DCE w/o NC (Net2)'. We observe that only one constituent network can be trained normally. Since there is no network consistency regularization and max-pooling is used to obtain consensus predictions, the corresponding gradient signal may be suitable for one member only, while misleading the other.

4) Adversarial Training: Removing the adversarial training terms in Eq.(7) also leads to a drop in performance. However, 'DCE w/o AT' still surpasses 'Baseline'. This term imposes a prediction consistency constraint for neighboring data points, and stabilizes the estimation of conditional entropy on unlabeled training samples. We consider that perturbation-based adversarial training is complementary to temporal consistency and network consistency regularization in achieving performance gains.

5) Weighted Conditional Entropy: The weighted conditional entropy term in Eq.(7) is used to force the model to produce confident predictions on the important training samples, such that the decision boundaries are pushed away from these samples in the latent feature space. We disable classification landmark learning, and set the conditional entropy coefficient $\varphi(x_u) = 1$ for all training samples. The resulting model 'DCE w/o LL' performs worse than the DCE model in all the cases, which confirms that the classification landmarks are more important than other unlabeled training samples for determining the final decision boundaries.

6) Combining Network Outputs: We also explore another strategy to combine the outputs of constituent networks by substituting average-pooling for max-pooling in Eq.(9) and Eq.(14), and the corresponding model 'DCE (`avg-pool`)' slightly underperforms the DCE model. In contrast to average-pooling which treats the two constituent networks equally, max-pooling allows one network to be better trained, which in turn improves the performance of the other via network consistency regularization.

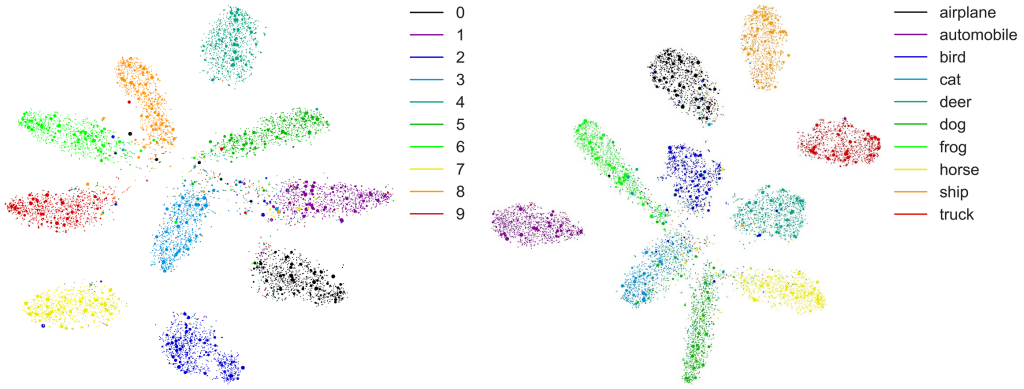
According to the comparison results between the proposed DCE model and its variants, we consider that all these aspects are useful in facilitating semi-supervised classification. In general, adversarial training and classification landmark learning are comparatively more important than other improvement strategies in complex datasets: CIFAR-10 and CIFAR-100.

E. Visualization

To illustrate the training process of the DCE model, Fig.3 shows the accuracy curves of the constituent networks, 'DCE (Net1)' and 'DCE (Net2)', versus the number of training epochs on CIFAR-10 with 4000 labels. In our model, the two networks are separate, and have different initializations and

(a) (b) (c)

Fig. 4. The representative unlabeled training samples with large weights (upper) and small weights (bottom). The unlabeled samples close to the specific class centers have larger weights, and they play a more important role in determining the final decision boundaries. (a) MNIST. (b) SVHN. (c) CIFAR-10.



(a) (b)

Fig. 5. The t-SNE plots of the consensus representations of the unlabeled training samples. The different sizes of the data points indicate the weights of the corresponding samples. Most of the data points with large weights are located in the correct clusters. (a) SVHN. (b) CIFAR-10.

dropout, which facilitates collaborative learning. In particular, network consistency regularization is able to utilize the output of one network to regularize the label prediction of the other, which alleviates the risk of severe performance degradation due to one constituent network dominating the training process. As a result, the members eventually achieve very similar classification performance which is better than that of the baseline model.

To further illustrate the effectiveness of classification landmark learning, Fig.4 shows some unlabeled training samples with different weights for the following cases: MNIST with 100 labels, SVHN with 1,000 labels and CIFAR-10 with 4,000 labels. The unlabeled samples having small weights are usually more difficult for the model to make correct predictions than those having large weights. Therefore, re-weighting unlabeled samples alleviates the influence of incorrect predictions during training.

We also visualize the distribution of unlabeled training samples using t-distributed stochastic neighbor embedding (t-SNE) [59] in Fig.5. In the t-SNE plots of the consensus representations, the data points belonging to the same class should be close together, and their class labels are indicated by different colors. In addition, the unlabeled training samples with different weights are denoted by different point sizes. It is observed that most of the data points with large weights are located in the correct clusters, which verifies the effectiveness of classification landmark learning during training, and these samples play a more important role in the training process through minimizing the corresponding weighted conditional entropy on them.

VI. Conclusion

We have proposed a new coupled ensemble model for improving the performance of deep convolutional networks in semi-supervised classification. With the proposed ensemble mechanism, self-learning and collaborative learning between constituent networks can be enhanced by incorporating complementary consistency regularization. Sharing knowledge from different sources is useful for improving the generalization capability of the ensemble members. To alleviate the influence of incorrect predictions on difficult instances, we adopt class-wise mean feature matching between labeled and unlabeled data to explore classification landmarks, and ensure that the final decision boundaries are distant to them by including a weighted conditional entropy term in the overall loss function. We show that the proposed DCE model

is effective in improving semi-supervised classification, and achieves state-of-the-art performance on the main benchmarks.

We adopt a comparatively small network architecture for each constituent network, and thus our network ensemble can be trained on a single GPU in the experiments. When handling complex datasets, e.g., ImageNet, larger constituent networks are needed. The proposed ensemble scheme can also be applied in this case by using two GPUs, such that an exclusive GPU can be assigned for the training of each constituent network. In addition, the difference between the two constituent networks are mainly due to stochastic augmentation, parameter initializations and dropout. In our future work, we would consider how to enhance the complementarity between the constituent networks to improve the joint training of the network ensemble.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Neural Inf. Process. Syst.*, 2014, pp. 1097–1105.

[2] C. Li, J. Zhu, and B. Zhang, "Max-margin deep generative models for (semi-)supervised learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 11, pp. 2762–2775, Nov. 2018. doi: 10.1109/TPAMI.2017.2766142.

[3] Z. Ding, N. M. Nasrabadi, and Y. Fu, "Semi-supervised deep domain adaptation via coupled neural networks," *IEEE Trans. Image Process.*, vol. 27, no. 11, pp. 5214–5224, Nov. 2018.

[4] H. Wu and S. Prasad, "Semi-supervised deep learning using pseudo labels for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1259–1270, Mar. 2018.

[5] S. Laine and T. Aila, "Temporal ensembling for semi-supervised learning," in *Proc. Int. Conf. Learn. Represent.*, 2017.

[6] P. Bachman, O. Alsharif, and D. Precup, "Learning with pseudo-ensembles," in *Proc. Neural Inf. Process. Syst.*, 2014, pp. 3365–3373.

[7] O. Chapelle and A. Zien, "Semi-supervised classification by low density separation," in *Proc. Int. Workshop Artif. Intell. Statist.*, 2005, pp. 57–64.

[8] R. Shu, H. Bui, H. Narui, and S. Ermon, "A DIRT-T approach to unsupervised domain adaptation," in *Proc. Int. Conf. Learn. Represent.*, 2018.

[9] T. Miyato, S. Maeda, M. Koyama, K. Nakae, and S. Ishii, "Distributional smoothing with virtual adversarial training," in *Proc. Int. Conf. Learn. Represent.*, 2016.

[10] T. Miyato, S. Maeda, S. Ishii, 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. doi: 10.1109/TPAMI.2018.2858821.

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

[12] Y. Zhang, T. Xiang, T. Hospedales, and H. Lu, "Deep mutual learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4320–4328.

[13] G. French and M. Mackiewicz, "Self-ensembling for domain adaptation," in *Proc. Int. Conf. Learn. Represent.*, 2018.

[14] O. Chapelle, B. Scholkopf, and A. Zien, "Semi-supervised learning," *IEEE Trans. Neural Netw.*, vol. 20, no. 3, p. 542, Mar. 2009.

[15] X. Zhu, Z. Ghahramani, and J. D. Lafferty, "Semi-supervised learning using Gaussian fields and harmonic functions," in *Proc. 20th Int. Conf. Mach. Learn.*, 2003, pp. 912–919.

[16] D. Zhou, O. Bousquet, T. Lai, J. Weston, and B. Schöikopf, "Learning with local and global consistency," in *Proc. Neural Inf. Process. Syst.*, 2004, pp. 321–328.

[17] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *J. Mach. Learn. Res.*, vol. 7, pp. 2399–2434, Nov. 2006.

[18] C. Gong, T. Liu, D. Tao, K. Fu, E. Tu, and J. Yang, "Deformed graph Laplacian for semisupervised learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 10, pp. 2261–2274, Oct. 2015.

[19] C. Gong, D. Tao, S. J. Maybank, W. Liu, G. Kang, and J. Yang, "Multi-modal curriculum learning for semi-supervised image classification," *IEEE Trans. Image Process.*, vol. 25, no. 7, pp. 3249–3260, Jul. 2016.

[20] M. Luo, L. Zhang, F. Nie, X. Chang, B. Qian, and Q. Zheng, "Adaptive semi-supervised learning with discriminative least squares regression," in *Proc. Int. Joint Conf. Artif. Intell.*, 2017, pp. 2421–2427.

[21] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[22] A. Kumar, P. Sattigeri, and T. Fletcher, "Semi-supervised learning with GANs: Manifold invariance with improved inference," in *Proc. Neural Inf. Process. Syst.*, 2017, pp. 5534–5544.

[23] Z. Dai, Z. Yang, F. Yang, W. W. Cohen, and R. Salakhutdinov, "Good semi-supervised learning that requires a bad GAN," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6513–6523.

[24] T. Salimans, I. Goodfellow, W. Zaremba, C. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2234–2242.

[25] X. Wei, B. Gong, Z. Liu, W. Lu, and L. Wang, "Improving the improved training of wasserstein GANs: A consistency term and its dual effect," in *Proc. Int. Conf. Learn. Represent.*, 2018.

[26] J. T. Springenberg, "Unsupervised and semi-supervised learning with categorical generative adversarial networks," in *Proc. Int. Conf. Learn. Represent.*, 2016.

[27] A. Odena, "Semi-supervised learning with generative adversarial networks," in *Proc. Int. Conf. Learn. Represent.*, 2016.

[28] V. Dumoulin *et al.*, "Adversarially learned inference," in *Proc. Int. Conf. Learn. Represent.*, 2017.

[29] C. Li, K. Xu, J. Zhu, and B. Zhang, "Triple generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4088–4098.

[30] Z. Gan *et al.*, "Triangle generative adversarial networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5247–5256.

[31] Y. Grandvalet and Y. Bengio, "Semi-supervised learning by entropy minimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2004, pp. 529–536.

[32] D. H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Proc. ICML Workshop Challenges Represent. Learn.*, 2013, p. 2.

[33] M. Sajjadi, M. Javanmardi, and T. Tasdizen, "Regularization with stochastic transformations and perturbations for deep semi-supervised learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 1163–1171.

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

[35] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. Int. Conf. Learn. Represent.*, 2015.

[36] S. Park, J. Park, S.-J. Shin, and I.-C. Moon, "Adversarial dropout for supervised and semi-supervised learning," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 3917–3924.

[37] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[38] L. Wan, M. Zeiler, S. Zhang, Y. Lecun, and R. Fergus, "Regularization of neural networks using DropConnect," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1058–1066.

[39] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger, "Deep networks with stochastic depth," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 646–661.

[40] S. Singh, D. Hoiem, and D. Forsyth, "Swapout: Learning an ensemble of deep architectures," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 28–36.

[41] D. He *et al.*, "Dual learning for machine translation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 820–828.

[42] T. Liu and D. Tao, "Classification with noisy labels by importance reweighting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 3, pp. 447–461, Mar. 2016.

[43] X. Yu, T. Liu, M. Gong, K. Zhang, K. Batmanghelich, and D. Tao, "Transfer learning with label noise," 2018, *arXiv:1707.09724*. [Online]. Available: https://arxiv.org/abs/1707.09724

[44] A. Angelova, Y. Abu-Mostafam, and P. Perona, "Pruning training sets for learning of object categories," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, pp. 494–501.

[45] B. Han *et al.*, "Co-teaching: Robust training of deep neural networks with extremely noisy labels," in *Proc. Neural Inf. Process. Syst.*, 2018, pp. 8527–8537.

[46] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, and L. Fei-Fei, "MentorNet: Learning data-driven curriculum for very deep neural networks on corrupted labels," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 3601–3610.
[47] J. Cheng, T. Liu, K. Ramamohanarao, and D. Tao, "Learning with bounded instance- and label-dependent label noise," 2019, *arXiv:1709.03768*. [Online]. Available: https://arxiv.org/abs/1709.03768
[48] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *Proc. NIPS Deep Learn. Represent. Learn. Workshop*, 2014.
[49] Y.-H. H. Tsai, Y.-R. Yeh, and Y.-C. F. Wang, "Learning cross-domain landmarks for heterogeneous domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 5081–5090.
[50] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[51] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[52] Y. Netzer, T. Wang, A. Goates, A. Bissacco, B. Wu, and A. Ng, "Reading digits in natural images with unsupervised feature learning," in *Proc. NIPS Workshop Deep Learn. Unsupervised Feature Learn.*, 2011.
[53] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Tech. Rep. TR-2009, 2009.
[54] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2006, pp. 2169–2178.
[55] H.-W. Ng and S. Winkler, "A data-driven approach to cleaning large face datasets," in *Proc. Int. Conf. Image Process.*, Oct. 2014, pp. 343–347.
[56] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015.
[57] S. Wu, Q. Ji, S. Wang, H. Wong, Z. Yu, and Y. Xu, "Semi-supervised image classification with self-paced cross-task networks," *IEEE Trans. Multimedia*, vol. 20, no. 4, pp. 851–865, Aug. 2018.
[58] Y. Luo, J. Zhu, M. Li, Y. Ren, and B. Zhang, "Smooth neighbors on teacher graphs for semi-supervised learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8896–8905.
[59] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

Jichang Li is currently pursuing the M.S. degree with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China.

Si Wu received the Ph.D. degree in computer science from the City University of Hong Kong, Hong Kong, in 2013. From 2013 to 2014, he was a Postdoctoral Fellow with the School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON, Canada. He is currently an Associate Professor with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. His research interests include machine learning and computer vision.

Cheng Liu received the Ph.D. degree in computer science from the City University of Hong Kong, Hong Kong, in 2017. He is currently an Assistant Professor with the Department of Computer Science, Shantou University, Shantou, China. His research interests include machine learning and pattern recognition.

Zhiwen Yu received the Ph.D. degree from the City University of Hong Kong, Hong Kong, in 2008. He is currently a Professor with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. He is a Senior Member of the ACM, the China Computer Federation, and the Chinese Association for Artificial Intelligence.

Hau-San Wong received the B.Sc. and M.Phil. degrees in electronic engineering from The Chinese University of Hong Kong, Hong Kong, and the Ph.D. degree in electrical and information engineering from The University of Sydney, Sydney, Australia. He is currently an Associate Professor with the Department of Computer Science, City University of Hong Kong, Hong Kong. He has coauthored the book named *Adaptive Image Processing: A Computational Intelligence Perspective* (CRC Press) and is now in its second edition. His research interests include machine learning, image/video analysis, and bioinformatics.