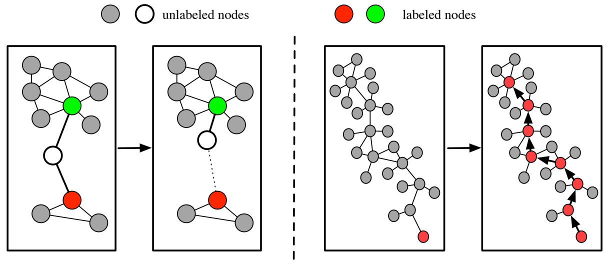
Density-Aware Graph for Deep Semi-Supervised Visual Recognition

Suichan Li[1,2] Bin Liu[1,2] Dongdong Chen[3] Qi Chu[1,2*] Lu Yuan[3] Nenghai Yu[1,2]
[1]School of Information Science and Technology, University of Science and Technology of China
[2]Key Laboratory of Electromagnetic Space Information, the Chinese Academy of Sciences
[3]Microsoft Research

Abstract

Semi-supervised learning (SSL) has been extensively studied to improve the generalization ability of deep neural networks for visual recognition. To involve the unlabelled data, most existing SSL methods are based on common density-based cluster assumption: samples lying in the same high-density region are likely to belong to the same class, including the methods performing consistency regularization or generating pseudo-labels for the unlabelled images. Despite their impressive performance, we argue three limitations exist: 1) Though the density information is demonstrated to be an important clue, they all use it in an implicit way and have not exploited it in depth. 2) For feature learning, they often learn the feature embedding based on the single data sample and ignore the neighborhood information. 3) For label-propagation based pseudo-label generation, it is often done offline and difficult to be end-to-end trained with feature learning. Motivated by these limitations, this paper proposes to solve the SSL problem by building a novel density-aware graph, based on which the neighborhood information can be easily leveraged and the feature learning and label propagation can also be trained in an end-to-end way. Specifically, we first propose a new Density-aware Neighborhood Aggregation(DNA) module to learn more discriminative features by incorporating the neighborhood information in a density-aware manner. Then a novel Density-ascending Path based Label Propagation(DPLP) module is proposed to generate the pseudo-labels for unlabeled samples more efficiently according to the feature distribution characterized by density. Finally, the DNA module and DPLP module evolve and improve each other end-to-end. Extensive experiments demonstrate the effectiveness of the newly proposed density-aware graph based SSL framework and our approach can outperform current state-of-the-art methods by a large margin.



(a) Density-aware Neighborhood Aggregation (b) Density-ascending Label Propagation

Figure 1: (a) Density-aware Neighborhood Aggregation(DNA) scheme prefer neighbors with higher density when target sample has equal similarity with the neighbors which belong to different clusters. (b) Density-ascending Path-based Label Propagation(DPLP) constructs a density-ascending path and propagates the labels from labelled samples to unlabelled samples within the path.

1. Introduction

Recently, semi-supervised learning (SSL) has been extensively studied to improve the generalization ability of deep neural networks for visual recognition by utilizing limited labelled images and massive unlabelled images. For leveraging the unlabelled data, current SSL methods are usually based on a common density-based *cluster assumption*, *i.e*. samples lying in the same high-density region are likely to belong to the same class[4, 26]. Its equivalent one is *low-density separation assumption*, which states that the decision boundary should not cross high density regions, but instead lies in low density regions. Current methods mainly focus on enforcing the low-density separation assumption by encouraging invariant prediction for perturbations around each unlabelled data point or same predictions for nearby samples, including consistency-regularization based methods [15, 24, 26, 2] and pseudo-label based methods [16, 23, 10].

While *density-peak assumption* [20] states that high-density samples are more likely to be the center of a cluster, thus samples with high density can encode more representative information of the cluster, which are valuable

[*]corresponding author.

clues for semi-supervised learning. However, current methods have not consider such density information explicitly or exploited it in depth. Moreover, as we know, the performance of current SSL frameworks is mainly determined by two aspects: feature learning and pseudo label generation for unlabelled samples. Nevertheless, when learning the feature embedding, current methods mainly leverage the single data sample and ignore the abundant neighborhood information which is helpful to learn more discriminative features. And for pseudo label generation, existing methods often directly choose current model predictions[16] or perform label propagation[10] to generate labels for unlabeled samples, which induce either inaccurate pseudo labels, or high matrix computation cost which is difficult to be end-to-end trained with the feature learning part.

Motivated by above observations, this paper proposes a novel density-aware graph based SSL framework. By building a density-aware graph, the neighborhood information can be easily utilized for each data sample. More importantly, the feature learning part and the label propagation part can also be end-to-end trained in the newly proposed framework. Further more, to better leverage the density information, we explicitly incorporate it into these two parts respectively.

Specifically, given the labelled and unlabelled data samples, a density-aware graph will be first built and we define the density for each node in the graph. Then for feature embedding learning, rather than only based on each single data sample, we propose to aggregate the neighborhood features to enhance the target features instead, which is demonstrated to be very useful in modern Graph Neural Networks(GNNs) [22, 13, 17]. However for the current aggregation schemes, the aggregation weights are often only characterized by the feature similarity between target node and its neighbors. In such case, when the target node has equal similarity with two neighbors which belong to two different clusters, it will give the same weight to these two neighbors. Motivated by the aforementioned density-peak assumption, we propose a novel Density aware Neighborhood Aggregation(DNA) scheme. Concretely, besides considering the feature similarity, we take the neighbor density information into account as well when calculating the aggregating weights. Intuitively, we want higher density neighbors to have higher importance. One simple explanation of this strategy is illustrated in Fig. 1(a).

To generate pseudo-labels for unlabelled samples more efficiently, we follow the basic label-propagation scheme which propagates the labels from labelled samples to unlabelled samples. However, we are not going to perform label propagation through a linear system solver used in [10, 31], which induces high matrix computational cost and works offline while training. Inspired by the aforementioned density assumption again, given one unlabelled sample, we argue the pseudo-label generated from neighboring samples with higher density is more possibly precise than that from neighbors with lower density. Based on this insight, we further propose a novel Density-ascending Path-based Label Propagation (DPLP) module. Specifically, for each sample, we will construct a density-ascending path where densities of samples are characterized by ascending-order, and perform label propagation within this path, which is efficient and can be online trained with feature learning in an end-to-end fashion. A graphical illustration can be found in Fig. 1(b).

In summary, the main contributions of this work include: (1) We propose a novel density-aware graph based SSL framework. To the best of our knowledge, this is the first work which exploits the density information explicitly for deep semi-supervised visual recognition; (2) A new Density-aware Neighborhood Aggregation module and Density-ascending Path-based Label Propagation module are designed for better feature learning and pseudo label generation respectively. The two modules are integrated into a unified framework and can be end-to-end trained; (3) Extensive experiments demonstrate the effectiveness of the proposed framework, which significantly outperforms the current state-of-the-art methods.

2. Related Works

Consistency-regularization for SSL. These methods usually apply a consistency loss on the unlabeled data, which enforce invariant predictions for perturbations of unlabelled data. For example, Π-model [15] proposes to use a consistency loss between the outputs of a network on random perturbations of the same image, while Laine *et al*. [15] apply consistency constraints between the output of the current network and the temporal average of outputs during training. The mean teacher (MT) method [24] replaces output averaging by averaging of network parameters. To utilize the structural information among unlabeled data points, [23] applies a Min-Max Feature regularization loss to encourage networks to learn features with better between-class separability and within-class compactness. Similarly, Luo *et al*. [18] utilize the contrastive loss to enforce neighboring points to have consistent predictions while the non-neighbors are pushed apart from each other. Although these methods have exploited neighborhood and density information, they are in the form of regularization terms or loss functions. By contrast, our method proposes to aggregate neighborhood features to enhance the target feature in a more explicit density-aware manner.

Pseudo-labeling for SSL. To leverage unlabelled data, pseudo-label based methods try to assign *pseudo labels* to the unlabeled samples based on labelled samples, then train the network in a fully supervised way. To generate precise pseudo labels, Lee *et al*. [16] use the current network predictions with high confidence as pseudo-labels for unlabeled examples. Shi *et al.* [23] use the network class prediction as hard labels for the unlabeled samples and introduce an uncertainty weight. Recently, Iscen *et al.* [10] employ graph-based label propagation to infer labels for unlabeled samples. However, they perform label propagation through a linear system solver on the training set offline with high computational cost, thus cannot be trained in an end-to-end way. In this work, we propose to construct a density-ascending path and perform label propagation within this path, which is much more efficient and can be end-to-end trained.

Neighborhood Aggregation in GNNs. Modern GNNs broadly follow a neighborhood aggregation scheme, where each node aggregates feature vectors of its neighbors to get more representative feature vector [28]. Different GNNs can vary in how they perform neighborhood aggregation. For example, Kipf *et al.* [13] use mean-pooling based neighborhood aggregation and Hamilton *et al.* [7] propose three aggregator functions: Mean aggregator, Max-Pooling aggregator and LSTM aggregator. Recently, inspired by self-attention mechanism, Petar *et al.* [25] propose an attention-based aggregation architecture by learning adaptive aggregation weights. Li *et al.* [17] extend the attention-based aggregation by supervising the attention weights with node-wise class relationship. After careful study, we find most of these neighborhood aggregation methods only consider the feature similarity between target sample and its neighbors when defining the aggregating weights. However, density information is shown to be a very important clue for SSL. Therefore, besides feature similarity, this paper also takes the neighborhood density into consideration and proposes a novel density-aware neighborhood aggregation scheme.

3. Preliminary

In semi-supervised learning, a small amount of labelled training samples $\mathcal{D}_l = \{(x_i, y_i)\}_{i=1}^{l}$ and a large set of unlabelled training samples $\mathcal{D}_u = \{x_j\}_{j=1}^{u}$ are often given, where l and u are number of labelled and unlabelled samples respectively and usually $l \ll u$. Then the goal of SSL is to leverage both $\mathcal{D}_l$ and $\mathcal{D}_u$ to train a better and generalized recognition model. Formally, let $m = l + u$ be the total number of training samples, f_θ be the feature extractor and h_ϕ be the classifier, current deep SSL methods adopt a similar optimization formulation:

$$\min_{\theta,\phi} \sum_{i=1}^{m} \ell\left(h_\phi(f_\theta(x_i)), \hat{y}_i\right) + \lambda R(\theta, \phi, \mathcal{D}_l, \mathcal{D}_u) \qquad (1)$$

where ℓ is the loss function like cross-entropy loss. For labelled data, $\hat{y}$ is the ground-truth label, while for unlabelled data, $\hat{y}$ can be pseudo-label. R is the regularization term, which encourages the model to generalize better to unseen data. Inspired by [6, 3], we add regularization term as follow:

$$\begin{aligned} R = &-\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{n_c} h_j(x_i;\theta,\phi) log h_j(x_i;\theta,\phi) \\ &- \frac{1}{n_c}\sum_{j=1}^{n_c} log\bar{h}_j(X;\theta,\phi) \end{aligned} \qquad (2)$$

where n_c is the number of classes, $\bar{h}_j(X;\theta,\phi)$ represents the mean softmax predictions of the model for category j across current training batch. The first term is the entropy minimization objective defined in [6], which simply encourages the model output to have low entropy; while the second term encourages the model to predict each class with equal frequency on average [3].

4. Method

Overview. In this work, we introduce a unified framework for joint feature learning and label propagation on a density-aware graph for semi-supervised visual recognition, which can be trained in an end-to-end fashion. A graphical overview of the proposed framework is depicted in Fig.2. First, we construct a k-nearest neighbor graph and define the *density* for each node in the graph. Then based on the density-aware graph, we propose to learn the feature embedding and pseudo label generation simultaneously for each node in the graph. Specifically, for each target node, we will sample its neighborhood sub-graph and learn feature embedding on this sub-graph by incorporating the neighborhood information with Density-aware Neighborhood Aggregation(DNA). For pseudo label generation, we propose Density-ascending Path-based Label Propagation (DPLP), *i.e.*, build a density-ascending path for each node in the graph and propagate the labels from labelled nodes to unlabelled nodes within this path.

4.1. Density-Aware Graph

Given a pre-trained feature extractor and classifier, we first extract the feature vectors and label predictions for all training samples, and organize them as Feature Bank and Label Bank respectively, which can be accessed through index later. Based on the features in the feature bank, we construct the global k-nearest neighbor affinity graph (k = 64 in this work). We then define the *density* ρ for each node u in the graph as:

$$\rho_u = \frac{1}{|\mathcal{N}_k(u)|} \sum_{v \in \mathcal{N}_k(u)} \tilde{\mathbf{f}}_u^T \tilde{\mathbf{f}}_v \qquad (3)$$

where $\mathcal{N}_k(u)$ is the k-nearest neighbors of node u, and $\tilde{\mathbf{f}}_u$, $\tilde{\mathbf{f}}_v$ are the L2-normalized feature embedding of node u and v. Intuitively, this formula expresses the density of each

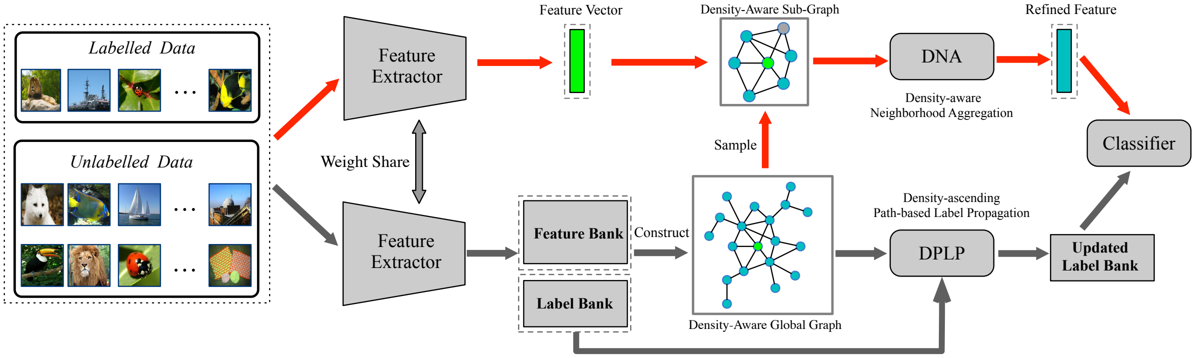


Figure 2: The overview of the proposed density-aware graph based SSL framework. The red line demonstrates the data flow in training, which includes feature extraction, sub-graph sampling and density-ware neighbourhood aggregation. The grey line shows the flow for density-ascending path-based label propagation.

node u as the average of the similarities between u and its neighbors. Need to note that other definitions of the density can also be considered, such as the number of neighbors whose similarity with target node is greater than a predefined threshold [20]. But it is not the focus of this work. We refer to the graph equipped with the density as *Density-Aware Graph*;

4.2. Density-Aware Neighborhood Aggregation

Current methods mainly focus on regularizing the output to be smooth near a data sample locally but learn the feature embedding only based on each single data sample. That is to say, they have not fully explored the important neighborhood information for feature learning, which is demonstrated to be very useful in other tasks [32, 17, 21, 8, 30]. Motivated by this observation, we propose to enhance the feature embedding of each target sample by aggregating the neighborhood features. Specifically, for each sample in the current training batch, we first pass it through the backbone to get the feature vector, and get corresponding node in the global density-aware graph (referred as target node), then we sample the neighborhood nodes of the target node from the global graph to obtain a sub-graph.

Sub-Graph Construction For construction of the sub-graph, we follow [17] and organize the neighbor nodes as a Tree-Graph. In particular, we take the target node as the root node, then build the tree in an iterative fashion. Each time, we extend all the leaf nodes by adding their k nearest neighbors from global-graph as the new leaf nodes. The tree graph grows until it reaches a predefined depth h. Based on Tree-Graph, we then iteratively perform feature aggregation among connected nodes and gradually propagates information within the sub-graph from leaf nodes to the target node. In the experimental part, we will study the effect of the number of sampling neighbors k and graph depth h.

Density-aware Neighborhood Aggregation(DNA) After sampling of sub-graph, we propose to improve target feature embedding by aggregating its neighbors embedding in the sub-graph. General aggregation strategies like mean-pooling and max-pooling cannot determine which neighbors are more important. Recently, to adaptively aggregate the features, [25, 17] proposed an attention-based architecture to perform neighboring aggregation, whose aggregation weights are characterized by the feature similarity between target node and its neighbors. Formally, the adaptive aggregation can be denoted as:

$$\mathbf{f}'_u = \mathbf{W}_A \left(\mathbf{f}_u + \sum_{v \in \mathcal{N}(u)} a_{u,v} \mathbf{f}_v \right) + \mathbf{b}_A \tag{4}$$

where $\mathbf{W}_A, \mathbf{b}_A$ are extra parameters for feature transformation. And $a_{u,v}$ is the aggregation weight denoting how much neighbor node v contributes to the target node u:

$$a_{u,v} = \frac{e^{p_{u,v}}}{\sum_{k \in \mathcal{N}(u)} e^{p_{u,k}}}, \tag{5}$$

$$p_{u,v} = \tilde{\mathbf{f}}_u^T \tilde{\mathbf{f}}_v, \quad \tilde{\mathbf{f}}_u = \frac{\mathbf{f}_u}{\|\mathbf{f}_u\|}, \quad \tilde{\mathbf{f}}_v = \frac{\mathbf{f}_v}{\|\mathbf{f}_v\|} \tag{6}$$

In details, we first perform feature L2-normalization, then define the similarity $p_{u,v}$ as the inner product of L2-normalized feature, and get the final aggregation weights by normalizing the similarity with the softmax operation.

However in SSL, we find only considering the aggregation weight with feature similarity is sub-optimal. In this way, if the target node has equal similarity with two neighbors that belong to two different clusters, the same aggregation weights will be assigned to these two neighbors. In fact, based on the *density-peak assumption*, the nodes with higher density are more closer to the cluster center and more discriminative. Therefore, besides the feature similarity information, we propose to incorporate the density information of each neighbor as well when calculating the aggregation weights. By default, we simply combine feature similarity $\mathbf{p}$ and density ρ with element-wise summation and

Algorithm 1 Density-Ascending Path-based Label Propagation

```
Input: Density-Aware Graph $\mathcal{G}$, Label Bank $\mathcal{B}$, labeled node indices $\mathcal{I}_l$, unlabeled node indices $\mathcal{I}_u$.
Output: Updated label Bank $\mathcal{B}^{'}$
$\mathcal{I}_u^{'} = \mathcal{I}_u, \mathcal{B}^{'} = \mathcal{B}$
$\mathcal{I}_l^{'} = $ SortInDensityDecending$(\mathcal{I}_l, \mathcal{G})$
for $i \in \mathcal{I}_l^{'}$ do
    $\mathcal{P} = $ ConstructDensityAscendingPath$(i, \mathcal{G})$
    for $j \in \mathcal{P}$ do
        if $j \in \mathcal{I}_u^{'}$ then
            $\mathcal{B}^{'} = $ UpdateLabelBank$(\mathcal{B}^{'}, i, j)$
            $\mathcal{I}_u^{'} = \mathcal{I}_u^{'} \setminus \{j\}$
        end if
    end for
end for
for $i \in \mathcal{I}_u^{'}$ do
    $\mathcal{P} = $ ConstructDensityAscendingPath$(i, \mathcal{G})$
    for $j \in \mathcal{P}$ do
        if $j \in \mathcal{I}_l^{'}$ then
            $\mathcal{B}^{'} = $ Updatelabelbank$(\mathcal{B}^{'}, j, i)$
        end if
    end for
end for
return $\mathcal{B}^{'}$
```

rewrite Eq. 5 as follows:

$$a_{u,v} = \frac{e^{(p_{u,v}+\rho_v)}}{\sum_{k\in\mathcal{N}(u)} e^{(p_{u,k}+\rho_k)}}. \quad (7)$$

4.3. Density-ascending Path for Label Propagation

Density-Ascending Path Construction. We construct the density-ascending path for each node in the global density-aware graph. More specifically, for node u, we initialize the density-path as one-element set $\{u\}$. Then we add one new nearest neighbor node v, whose density is greater than the previous added node. We iteratively perform this process until the distance between the candidate node and the last added node is greater than a predefined threshold.

For notation clarity, we define the Density-Ascending Path as $\mathcal{P}(u) = \{v_1, v_2, ..., v_k, ...\}$, where v_k is the node added to the path at k-th step. Supposing the added node at k-th step is v_k, then node to be added is the neighbor node v_{k+1} with higher density:

$$v_{k+1} = \arg\min_v \Psi(\mathbf{f}_{v_k}, \mathbf{f}_v), v \in \{w|\rho_w > \rho_{v_k}\} \quad (8)$$

where $\Psi(\cdot)$ is a distance metric function, and we choose the L2-Euclidean distance metric in this work by default, i.e., $\Psi(\mathbf{f}_{v_k}, \mathbf{f}_v) = \|\mathbf{f}_v - \mathbf{f}_u\|_2$ To alleviate the influence of irrelevant neighbors, we define a threshold σ, to terminate the growth of the density-ascending path, i.e., for each node pair (v_k,v_{k+1}) in $\mathcal{P}(u)$, it satisfies: $\left\|\mathbf{f}_{v_k} - \mathbf{f}_{u_{k+1}}\right\|_2 \leq \sigma$.

Algorithm 2 Density-Aware Graph for Deep SSL

```
Require: Training set $\{\mathcal{X}, \mathcal{Y}_l\}$, training epochs $T$, training iterations $J$, initial feature bank $\mathcal{F}^0$, initial label bank $\mathcal{L}^0$, labeled indices $\mathcal{I}_l$, unlabeled indices $\mathcal{I}_u$
Require: Feature Extractor $f_\theta$, Classifier $g_\phi$
Require: DensityAwareNeighborhoodAggregation Dna$(\cdot|\omega)$
Require: DensityAscendingPathLabelPropagation Dplp$(\cdot)$
for $t \in \{1, \ldots, T\}$ do
    $\mathcal{F}^t, \mathcal{L}^t = $ Initalize$(f_\theta, g_\phi, \mathcal{X}, \mathcal{Y}_l, \mathcal{F}^{t-1}, \mathcal{L}^{t-1})$
    $\mathcal{G}^t = $ ConstructDensityAwareGraph$(\mathcal{F}^t, \mathcal{L}^t)$
    $\mathcal{Y} = $ Dplp$(\mathcal{G}^t, \mathcal{L}^t, \mathcal{I}_l, \mathcal{I}_u)$          ▷ label propagation
    for $i \in \{1, \ldots, J\}$ do
        $X^i, Y^i = $ GetTrainingBatch$(\mathcal{X}, \mathcal{Y}, i)$
        $F^i = f(X^i; \theta)$
        $\mathcal{G}_s = $ SampleSubGraph$(F^i, \mathcal{G}^t)$
        $F^i = $ Dna$(F^i, \mathcal{G}_s; \omega)$          ▷ feature aggregation
        $P^i = g(F^i; \phi)$          ▷ category predictions
        $\ell = $ CaculateLoss$(P^i, Y^i)$
        UpdateParameter$(\ell, \theta, \phi, \omega)$  ▷ back-propagation
    end for
end for
```

Before entering the Density-ascending Path-based Label Propagation, we first introduce the following assumption.

Assumption: *The labelled nodes with higher density in the density-ascending path are more possible to provide correct pseudo labels than the ones with lower density.*

Explanation: As stated in the *cluster assumption*, samples with the same label are more likely to lie in the high-density region. Meanwhile, the *density-peak assumption* [20] shows that high-density nodes are more likely to be the center of a cluster. Thus for one unlabelled node, the labelled nodes with higher density are more representative and more likely to provide correct pseudo labels than the ones having lower density in the same density-ascending path.

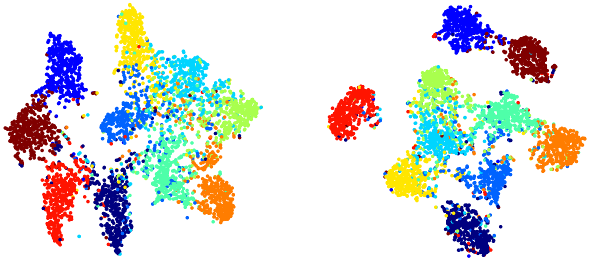
Based on *density-ascending path*, now we introduce our Density-ascending Path-based Label Propagation(DPLP) algorithm. Specifically, we first sort all the labelled nodes based on the density in descending order, then for each labelled node, we construct a density-ascending path and use the label of the max-density labelled node to update the entries of Label Bank corresponding to all the unlabelled nodes in this path. For remaining unlabelled nodes, we also construct a density-ascending path for each of them and update the corresponding entry of Label Bank using the label of labeled node with highest density in the path. The detailed procedures are summarized in Alg. 1.

4.4. Density Aware Graph-based SSL Framework

The above Density-aware Neighborhood Aggregation and Density-ascending Path-based Label Propagation are integrated into a unified Density Aware Graph-based SSL

Method No. of labelled images	CIFAR10 1000	CIFAR100 4000
Baseline	10.96	43.34
NA w/o density	9.46	40.76
NA with density	**9.18**	**40.33**

Table 1: Effectiveness of Neighborhood Aggregation(NA) on CIFAR10 and CIFAR100.



(a) Features without Aggregation

(b) Features with Aggregation

Figure 3: Visualization of feature embeddings on CIFAR10. Each dot in the figure corresponds to one image, and different colors represent different classes. (Best viewed in color.)

framework (dubbed as “DAG”) and can be trained in an end-to-end fashion. We summarize the whole training process in Alg. 2. Specifically, at the beginning of each epoch, we update the Feature Bank and Label Bank with the latest feature extractor and classifier. Then construct a new global density-aware graph based on current Feature Bank and perform density-ascending path label propagation based on DAG and current Label Bank. Need to note that we always use ground-truth labels for the Label Bank entries of labeled samples. After the above training preparation, we then start to train the framework by sampling batch images and labels. In details, for each batch of images, we feed them into the feature extractor and enhance their output features by density-aware neighborhood aggregation on sub-graphs before feeding them into the classifier.

5. Experiments

5.1. Experimental Setup

Dataset Setup. To verify the effectiveness, we conduct experiments on three popular datasets, namely CIFAR10 [14], CIFAR100 [14] and Mini-ImageNet [27]. In details, CIFAR10 and CIFAR100 both contain 50k images for training and 10k images for testing with resolution 32 × 32, but coming from 10 and 100 classes respectively. Following the standard SSL setting, for CIFAR10, we perform experiments with 50, 100, 200, and 400 labelled images per classes. And for CIFAR100, we experiment with 40 and 100 labelled images per class. For Mini-ImageNet, it is a subset of ImageNet[5] and consists of 100 classes with 600 images per class of resolution 84 × 84. With the same setting as [10], we randomly assign 500 images from each class to the training set, and 100 images to the test set. The train and test sets therefore contain 50k and 10k images. We then experiment with 40 and 100 labelled images per class for SSL. We perform ablation study on CIFAR10 and CIFAR100 with an independent validation set of 5K instances sampled from the training set as [1, 19] and compare with state-of-the-art methods on the standard test set.

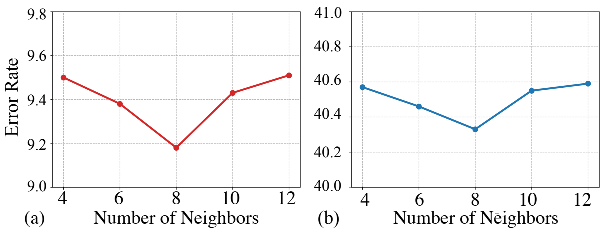


Figure 4: Influence of number of neighbors on CIFAR10(a) and CIFAR100(b).

Method No. of labelled images	CIFAR10 1000	CIFAR100 4000
h = 0(baseline)	10.96	43.34
h = 1	**9.18**	40.33
h = 2	9.20	**39.60**
h = 3	9.26	40.28

Table 2: Influence of neighbor-hops for neighborhood aggregation on CIFAR10 and CIFAR100.

Implementations. For model architecture, we use the same 13-layer CNN network as in [15, 24, 10] for CIFAR10/100 and ResNet-18 [9] for Mini-ImageNet as [10]. The SGD optimizer is used with the initial learning rate 0.1 and 0.2 for CIFAR10/100 and Mini-ImageNet respectively. We decay the learning rates by 0.1 at 250 and 350 epochs and obtain the final model after 400 epochs. We augment training images with random cropping and horizontal flipping as [15, 24]. Inspired by [29, 26, 2], we also employ Mixup strategy[29] to augment the training samples, which give us a stronger baseline. To build the Feature Bank and Label Bank at the first epoch, we use the model pre-trained only on labelled training samples. And during the testing stage, we directly construct neighborhood sub-graph by retrieving neighbors from the Feature Bank built in the training stage for each test sample.

5.2. Ablation Studies

5.2.1 Density-Aware Neighboring Aggregation

Effectiveness of Neighborhood Aggregation. We propose to aggregate the neighboring features to enhance the feature of the target instance, thus improving the performance of the semi-supervised image classification. To show the effectiveness of neighborhood aggregation, we conduct a baseline experiment without neighborhood aggregation and provide the comparison results in Tab.1. It shows that neighborhood aggregation (“NA w/o density” and “NA with density”) can significantly improve the baseline without neighborhood aggregation (“Baseline”). To have a deeper analysis, we further visualize the learned feature embedding of neighborhood aggregation and that of the baseline on the CIFAR10 validation set in Fig. 3, which clearly shows that incorporating neighbourhood features can help learn more discriminative feature embeddings.

Is density improving Neighborhood Aggregation? When learning the aggregation weights, besides considering the feature similarity between target node and its neighbors, we believe incorporating the density information of each neighbor for aggregation weight learning is also very important based on the *density-peak assumption*. To verify it, we compare the proposed density-aware neighbor aggregation (“NA with density”) with the version without considering the density (“NA w/o density”). The results in Tab.1 show that incorporating the density information of neighbors into the learning of aggregation weight can generally achieve superior performance.

Study about Sub-Graph size. By experiments, we find selecting a good neighbor number k and sub-graph depth h (“hop”) is crucial to get the best performance. First to study the influence of k, we only consider different number of neighbors in the first hop ($h = 1$). The results in Fig 4 show that a too large or too small number of neighbors will both result in inferior results. This is because a too small number of neighbors will not get sufficient neighbouring information while a too large number of neighbors will introduce unrelated neighbours which may weaken the effectiveness of neighborhood aggregation, which is consistent with the results in [17]. We then study if incorporate multi-hop neighbors ($h > 1$) can bring performance gain and show the results in Tab. 2. It can seen that incorporating two hops of neighbors can bring addition gain on CIFAR100, while yield no addition gain on CIFAR10. On the other hand, sampling the third hop of neighbors will degrade the performance on both datasets. We think it is because more unrelated samples may also be introduced as the neighbors hop increases, thus impairing the target feature.

5.2.2 Density-Ascending Path Label Propagation

Effectiveness of Density-Ascending Path. In this part, we study the effectiveness of our proposed Density-ascending Path-based Label Propagation(DPLP) which consists of two main sequential steps: first construct a density-ascending path from each labelled sample(denoted as LP-L), then construct a density-ascending path from each remaining unlabelled sample(denoted as LP-LU). Here we study these two steps respectively in Tab. 3. It shows that: (i) Density-Ascending Path-based Label Propagation can significantly improve the classification accuracy; (ii) Constructing the density-ascending path only from labelled samples(LP-L) has already significantly improved the baseline; (iii) Constructing the density-ascending path from remaining unlabelled samples(LP-LU) can further bring additional gain.

Method	CIFAR10		CIFAR100	
No. of labelled images	1000	4000	4000	10000
Baseline	10.96	7.85	43.34	37.80
LP-L	9.57	7.37	40.94	35.94
LP-LU	**9.14**	**7.03**	**40.08**	**34.72**

Table 3: Effectiveness of Density-ascending Path-based Label Propagation on CIFAR10 and CIFAR100.

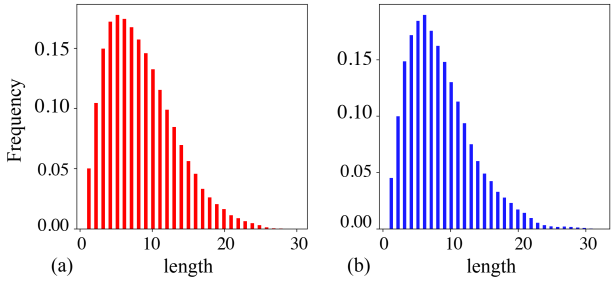


Figure 5: The distribution of length of density-ascending path on CIFAR10 (a) and CIFAR100 (b).

Method	CIFAR10	CIFAR100
No. of labelled images	1000	4000
Baseline	10.96	43.34
DNA	9.18	39.60
DPLP	9.14	40.08
DAG(Overall framework)	**8.35**	**38.04**

Table 4: Effectiveness of overall framework on CIFAR10 and CIFAR100.

Distribution of the length of Density-Ascending Path. As elaborated before, our density-ascending path-based label propagation constructs the path based on the ascending density constraint and terminates according to the feature similarity constraint. Though we have already demonstrated its effectiveness, we are still curious about the distribution of the length of the density-ascending path. In Fig. 5, we show the distribution of the density-ascending path length on CIFAR10(1k labelled samples) and CIFAR100(4k labelled samples) respectively. We can observe that the length of density-ascending path on CIFAR10(1k labelled samples) and CIFAR100(4k labelled samples) have a similar distribution, and is mostly between in 5 and 25. Therefore, it is efficient to online perform label propagation on this density-ascending path during the training stage.

Method	CIFAR10				CIFAR100	
No. of labelled images	500	1000	2000	4000	4000	10000
Π model [15]	-	-	-	12.36 ± 0.31	-	39.19 ± 0.36
TemporalEnsembling [15]	-	-	-	12.16 ± 0.24	-	38.65 ± 0.51
MeanTeacher [24]	-	21.55 ± 1.48	15.73 ± 0.31	12.31 ± 0.28	-	-
SNTG[18]	-	18.41 ± 0.52	13.64 ± 0.32	9.89 ± 0.34	-	37.97 ± 0.29
SWA [1]	-	15.58 ± 0.12	11.02 ± 0.23	9.05 ± 0.21	-	33.62 ± 0.54
ICT [26]	-	15.48 ± 0.78	9.26 ± 0.09	7.29 ± 0.02	-	-
DualStudent [12]	-	14.17 ± 0.38	10.72 ± 0.19	8.89 ± 0.09	-	32.77 ± 0.24
MixMatch[2]	9.65 ± 0.94	7.75 ± 0.32	**7.03 ± 0.15**	6.24 ± 0.06	-	-
TSSDL[23]	-	18.41 ± 0.92	13.54 ± 0.32	9.30 ± 0.55	-	-
LP[10]	24.02 ± 2.44	16.93 ± 0.70	13.22 ± 0.29	10.61 ± 0.28	43.73 ± 0.20	35.92 ± 0.47
DAG(Ours)	**9.30 ± 0.73**	**7.42 ± 0.41**	7.16 ± 0.38	**6.13 ± 0.15**	**37.38 ± 0.64**	**32.50 ± 0.21**

Table 5: Comparison with state-of-the-art methods on CIFAR10 and CIFAR100. Average error rate and standard deviation of 5 runs with different labelled/unlabelled splits are reported.

Method	4000	10000
MeanTeacher [24]	72.51 ± 0.22	57.55 ± 1.11
LP [10]	70.29 ± 0.81	57.58 ± 1.47
LP + MeanTeacher [10]	72.78 ± 0.15	57.35 ± 1.66
DAG(Ours)	**55.97 ± 0.62**	**47.28 ± 0.20**

Table 6: Comparison with state-of-the-art methods on Mini-ImageNet. Average error rate and standard deviation of 3 runs with different labelled/unlabelled splits are reported.

5.2.3 Density Aware Graph-based SSL Framework

In the previous subsections, we have demonstrated the effectiveness of each individual component, now we will study the effectiveness of the overall framework(DAG), *i.e.*, the combination of Density-aware Neighborhood Aggregation(DNA) and Density-ascending Path-based Label Propagation(DPLP). The results on CIFAR10 and CIFAR100 are displayed in Tab. 4. It can be seen that DNA and DPLP are two complementary modules, and combining them can outperform the baseline by a large margin. For example, DAN and DPLP can reduce the error rate to 39.6% and 40.08% on CIFAR100(4k labelled samples) respectively, and their combination further reduces the error rate to 38.04%.

Discussions about computational complexity. The main computation of our framework comes from global graph construction which involving kNNs retrieval, yet there already exists many highly efficient nearest neighbour searching algorithms and tools. The default tool we used is Faiss[11], which can perform efficient billion-scale similarity search with GPU. And for testing, we found the overhead of kNNs search is negligible compared to the feature extraction part, our test time is almost identical to the baselines.

5.3. Comparison with state-of-the-arts

We report the results with the state-of-the-art approaches in Tab. 5 and Tab. 6. To show our superiority, we consider both state-of-the-art consistency-regularization based methods [15, 24, 18, 1, 26, 12, 2] and pseudo-label based methods [23, 10] in Tab. 5 for CIFAR10 and CIFAR100. Among them, ICT [26] and MixMatch [2] both leveraged the Mixup data augmentation strategy [29], which is also used in this work. It can be seen that our method outperforms most state-of-the-art methods on CIFAR10 and CIFAR100 in terms of different numbers of labelled samples. On the more challenging Mini-ImageNet benchmark, our method achieves the best performance and records a new state-of-the-art 55.97% for 4k labelled samples and 47.28% for 10k labelled samples respectively, which beats latest best results [10] by 14.32% and 10.07%.

6. Conclusion

Although existing SSL methods are based on the common density-based *cluster assumption* and achieve impressive results, we find three limitations exist: 1) They have not exploited density information explicitly; 2) Neighborhood information is not considered when learning the feature; 3) Existing label propagation scheme can only be done offline and difficult to be end-to-end trained; In this paper, we propose a novel and unified density-aware graph based framework for semi-supervised visual recognition. Specifically, we propose two novel density-aware modules targeting at the two key SSL components respectively, i.e., Density-aware Neighborhood Aggregation and Density-ascending Path-based Label Propagation. These two modules can be jointly trained and work in a complementary way. Experiments demonstrate our superior performance, which beats current state-of-the-art methods by a large margin.

Acknowledgement

This work is supported by the Fundamental Research Funds for the Central Universities (WK2100330002, WK3480000005), National Key Research and Development Program of China(2018YFB0804100).

References

[1] Ben Athiwaratkun, Marc Finzi, Pavel Izmailov, and Andrew Gordon Wilson. There are many consistent explanations of unlabeled data: Why you should average. In *International Conference on Learning Representations*, 2019. 6, 8

[2] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, 2019. 1, 6, 8

[3] John S. Bridle, Anthony J. R. Heading, and David J. C. MacKay. Unsupervised classifiers, mutual information and ' phantom targets'. In *Advances in Neural Information Processing Systems*, pages 1096–1101, 1992. 3

[4] Olivier Chapelle and Alexander Zien. Semi-supervised classification by low density separation. In *AISTATS*, volume 2005, pages 57–64. Citeseer, 2005. 1

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 6

[6] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in neural information processing systems*, pages 529–536, 2005. 3

[7] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, 2017. 3

[8] Jiangfan Han, Xiaoyi Dong, Ruimao Zhang, Dongdong Chen, Weiming Zhang, Nenghai Yu, Ping Luo, and Xiaogang Wang. Once a man: Towards multi-target attack via learning multi-target adversarial network once. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5158–5167, 2019. 4

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[10] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5070–5079, 2019. 1, 2, 3, 6, 8

[11] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*, 2017. 8

[12] Zhanghan Ke, Daoye Wang, Qiong Yan, Jimmy Ren, and Rynson WH Lau. Dual student: Breaking the limits of the teacher in semi-supervised learning. In *IEEE International Conference on Computer Vision (ICCV)*, 2019. 8

[13] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. 2, 3

[14] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009. 6

[15] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *International Conference On Learning Representations*, 2017. 1, 2, 6, 8

[16] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, ICML*, volume 3, page 2, 2013. 1, 2

[17] Suichan Li, Dapeng Chen, Bin Liu, Nenghai Yu, and Rui Zhao. Memory-based neighbourhood embedding for visual recognition. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 2, 3, 4, 7

[18] Yucen Luo, Jun Zhu, Mengxi Li, Yong Ren, and Bo Zhang. Smooth neighbors on teacher graphs for semi-supervised learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8896–8905, 2018. 2, 8

[19] Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in Neural Information Processing Systems*, pages 3235–3246, 2018. 6

[20] Alex Rodriguez and Alessandro Laio. Clustering by fast search and find of density peaks. *Science*, 344(6191):1492–1496, 2014. 1, 4, 5

[21] Mohammad Sabokrou, Mohammad Khalooei, and Ehsan Adeli. Self-supervised representation learning via neighborhood-relational encoding. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 4

[22] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2008. 2

[23] Weiwei Shi, Yihong Gong, Chris Ding, Zhiheng MaXiaoyu Tao, and Nanning Zheng. Transductive semi-supervised deep learning using min-max features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 299–315, 2018. 1, 2, 3, 8

[24] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*, pages 1195–1204, 2017. 1, 2, 6, 8

[25] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representation*, 2018. 3, 4

[26] Vikas Verma, Alex Lamb, Juho Kannala, Yoshua Bengio, and David Lopez- Paz. Interpolation consistency training for semi-supervised learning. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, 2019. 1, 6, 8

[27] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016. 6

[28] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019. 3

[29] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. 6, 8

[30] Hang Zhou, Kejiang Chen, Weiming Zhang, Han Fang, Wenbo Zhou, and Nenghai Yu. Dup-net: Denoiser and upsampler network for 3d adversarial point clouds defense. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1961–1970, 2019. 4

[31] Xiaojin Zhu and Zoubin Ghahramani. Learning from labeled and unlabeled data with label propagation. 2002. 2

[32] Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins. Local aggregation for unsupervised learning of visual embeddings. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 4