Deep Co-Space: Sample Mining Across Feature Transformation for Semi-Supervised Learning

Ziliang Chen, Keze Wang, Xiao Wang, Pai Peng, Ebroul Izquierdo, and Liang Lin

***Abstract*—Aiming at improving performance of visual classification in a cost-effective manner, this paper proposes an incremental semi-supervised learning paradigm called Deep Co-Space (DCS). Unlike many conventional semi-supervised learning methods usually performing within a fixed feature space, our DCS gradually propagates information from labeled samples to unlabeled ones along with deep feature learning. We regard deep feature learning as a series of steps pursuing feature transformation, i.e., projecting the samples from a previous space into a new one, which tends to select the reliable unlabeled samples with respect to this setting. Specifically, for each unlabeled image instance, we measure its reliability by calculating the category variations of feature transformation from two different neighborhood variation perspectives, and merged them into an unified sample mining criterion deriving from Hellinger distance. Then, those samples keeping stable correlation to their neighboring samples (i.e., having small category variation in distribution) across the successive feature space transformation, are automatically received labels and incorporated into the model for incrementally training in terms of classification. Our extensive experiments on standard image classification benchmarks (e.g., Caltech-256 [1] and SUN-397 [2]) demonstrate that the proposed framework is capable of effectively mining from large-scale unlabeled images, which boosts image classification performance and achieves promising results compared to other semi-supervised learning methods.**

***Index Terms*—Cost-effective model, Visual Classification, Deep Semi-supervised Learning, Incremental Processing, Visual Feature Learning.**

I. Introduction

RECENTLY, tremendous advancements have been made in the field of vision by convolutional neural networks (CNNs), including classification [3], object detection [4], scene and human parsing [5], [6] and image caption generation [7]. The successes on these vision applications have exhibited impressive performances with ample well-annotated images for training. Though label information plays such a crucial role in those applications, the establishment of large scale dataset is too expensive to affordable under a practical scenario. Besides, annotating by human labor also tends to bring in certain noisy labels caused by the limitation of knowledge background from the ordinary subjects.

As the growing demand of improving the usage of existing label information to reduce the annotation cost, semi-supervised learning (SSL) obtains increasing attention. By ingeniously bridging the connection among unlabeled data and labeled information, SSL can performs well with a limited number of labeled samples. Its semi-supervised manner of learning and cost-effective property make it always in the forefront of computer vision and machine learning research. Currently, the progress of deep learning focuses on two branches for SSL algorithms, i.e., feature-fixed and feature-learnable SSL. The former usually refers to a variety of conventional SSLs (e.g., Graph-based SSL [8]–[10]), which consider samples in a handcrafted feature space during the whole training process. Differently, the latter additionally focuses on learning representation according to SSL configuration. Through learning both feature representation and training model parameter simultaneously, this branch usually pays close attention to the exploration about nonlinear functional approximation via semi-supervised metric learning [11] and newly rising deep learning [12], [13].

In spite of achieving remarkable successes in visual recognition, these two branches still face several limitations. In specific, conventional feature-fixed SSLs heavily rely on the feature engineering, which tends to strengthen some information illustrated in statistics and discard other information in visual aspect as a return. This leads to its failure under task-orientated scenario, which requires accurate feature representation to be adaptive to different visual understanding tasks. In respect to feature-learnable SSLs, though seeking overall distribution in feature space [14] under an end-to-end network training regime, it cannot be progressively optimized in such an incremental way due to the ignorance of modeling the local relationship among samples [15]. As discussed in [16], [17], these aforementioned limitations are still arousing wide concern in research.

Attempting to overcome these limitations from another point of view, we introduce an innovative sample mining strategy, which incrementally explores the related local structure for each unlabeled sample within two different feature spaces. More specifically, assuming that each couple of deep learning models being fine-tuned before/after can be viewed as two successive yet different feature spaces, we define the one-one nonlinear correspondence for each sample from the previous feature space to a new one as "feature transformation". As

Z. Chen, K. Wang and L. Lin are with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China and also with Engineering Research Center for Advanced Computing Engineering Software of Ministry of Education, China. Email: zlchilam@163.com; kezewang@gmail.com; linliang@ieee.org.

X. Wang is with the School of Computer Science, Anhui University, Hefei, P. R. China. Email: wangxiaocvpr@foxmail.com.

P. Peng is with Youtu Lab of Tencent, Shanghai, P. R. China. Email: popeyepeng@tencent.com.

E. Izquierdo is with the School of Electronic Engineering and Computer Science, Queen Mary University of London, London, U.K. (e-mail: ebroul.izquierdo@qmul.ac.uk).

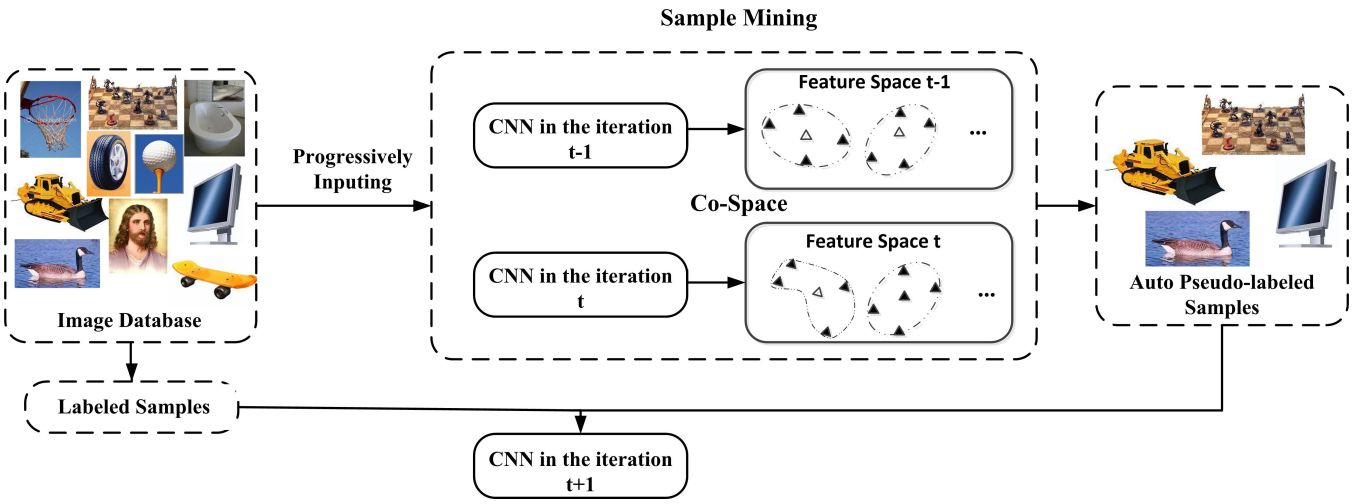

Fig. 1. The pipeline of the proposed Deep Co-Space framework. At the beginning, we have limited labeled data and infinite unlabeled data for training. The labeled data would be used to fine-tune a pre-trained CNN-based deep model and results in a new one. After that, all labeled and unlabeled data will be extracted by the old and new models to construct two successive feature space (Co-Space) respectively. We measure the distribution variation of labeled neighbors for each unlabeled sample in Co-Space, and assign those samples having stable structures with pseudo-labels. Then these selected samples are employed to update the model for the next iteration.

feature space transforming, according to low density separability assumption [18], samples in the same category tend to cluster together, keeping locally compact and semantically coherent, then those in different classes are inclined to diverge to corresponding categories, which leading to its labeled neighbors apparently to change. More precisely, with regard to each unlabeled sample in the transformed feature space, its labeled neighbors tend to form a locally stable distribution for a certain category. This inspires us that some unlabeled samples, remaining stable labeled neighbor distribution in the two successive feature spaces, can be employed by assigning them pseudo-labels to augment labeled dataset and improve the performance.

As illustrated in Fig. 1, an innovative incremental sample mining framework based on the intuition above, is proposed for deep semi-supervised learning. Since the progressive sample mining seems like a sequence of steps pursuing feature space transformation via gradually polishing a deep model, we name our framework as **Deep Co-Space (DCS)**, namely, there are two CNN models with the same architecture at each step in our framework. Note that, the second CNN has been fine-tuned based on the first one via the updated labeled data pool, then the training phase is performed as follows. Firstly, we extract feature representation for all labeled and unlabeled samples based on these two CNN models. Thus, we have obtained two successive feature spaces (Co-Space); Secondly, we launch our sample mining strategy to select those unlabeled samples which has locally stable neighbor distributions in the Co-Space and automatically annotate them with pseudo-labels. More specifically, for each unlabeled sample, we measure its reliability by calculating the category variations of its K nearest neighbors in Co-Space. In order to resist semantic drift [19], the variation needed to be considered from two points of views, i.e., neighborhood intrinsic variation and neighborhood category variation. Neighborhood intrinsic variation represents the intrinsic structure non-consistency of unlabeled samples via feature transformation in the Co-Space, while neighborhood category variation denotes the transforming variation of covariances among local labeled samples related to different categories. Then, we merge them into an unified sample mining criterion, which is based on Hellinger distance. Finally, given samples selected by this criterion, we augment labeled data pool in the image database and further fine-tune the CNNs. In this way, the updated CNN leads to a new Co-Space for sample mining at the next iteration.

The main contributions of this paper are in three-fold: i) To the best of our knowledge, DCS is the first incremental semi-supervised learning framework attempting to progressively propagate information from labeled samples to unlabeled ones along with a sequence of steps, which aims at leveraging feature transformation in a two successive feature space; ii) We present sufficient discussions and clarifications about how to incorporate the neighborhood intrinsic and category variation into an unified sample mining criterion deriving from Hellinger distance; iii) Extensive experiments on two public visual classification benchmarks, i.e., Caltech-256 [1] and SUN-397 [2], demonstrate the effectiveness of our DCS in SSL not only on the vanilla Alexnet [3] and VGG [20], but also on the recent DSSL network architecture [14].

The rest of the paper is organized as follows. Sect. II presents a review of approaches related to DCS. Sect. III overviews the a complete model about DCS, including definition, pipeline, and some theoretical discussion. The experimental results, comparisons and component analysis are presented in Sect. IV. Finally, Sect. V concludes the paper.

II. Related Work

In this section, we will give a brief review of some feature-fixed SSL approaches related to our framework, and the SSL methods related to neural network are exhibited in Sect. II-A.

Since our DCS shares some properties of multi-view learning, then the comparability and difference between them are discussed in Sect. II-B.

A. Semi-supervised Learning

1) Feature-fixed SSL: The most aged SSL method starts from self-training [21], which was invented to train a classifier with small amount of labeled data to annotate unlabeled data, then retrain the classifier with labeled and unlabeled data iteratively. The method is straightforward both in intuition and formulation, but always beset by semantic drift. It has been extended into many variants [22] [23] to prevent this problem, and most of them rely on knowledge from fixed feature space.

Probabilistic graphical model plays an important role in the development of SSL. For instance, Ji et al. [24] merges the supervised and unsupervised hidden Markov models into an associated estimation problem as a set of fixed point equations; Mao et al. [25] explores new latent topic in LDA (Latent Dirichlet Allocation) with labeled hierarchical information. All of them utilize all data to model the joint probability distribution in generative process with discriminative information. They are well-defined in theory, but suffer from high variance in generative process when the assumption of prior distribution is inappropriate. Besides, compared with deep learning, pure graphical models rely on features with high-level semantics in statistics, which makes those methods more preferable in addressing problems about natural language processing.

Graph based semi-supervised learning draws attention of many researchers both in transductive and inductive learning settings, such as label propagation [18], manifold regularization [26], Planetoid [27] e.t.c. The problem is usually formulated as

$$\min_{f} \ \lambda f(X)^T \Delta f(X) + \mathcal{L}(f(X),\ Y)$$

where $\Delta = \begin{bmatrix} L_{uu} & L_{ul} \\ L_{ul}^T & L_{ll} \end{bmatrix}$ is a matrix about unlabeled and labeled dataset, and is related to the finite weighted graph $\mathcal{G} = (V, E, W)$. Specifically, $\mathcal{G}$ consists of a set of vertexes V based on all data, and can be provided from external knowledge or pre-definition. The edge set E and its specified weights W are formulated with non-negative symmetric function. Note that Δ is also determined before optimizing. When $\mathcal{G}$ is required for calculation, we will interpret the $W(i, j)$ as a local similarity measure between the vertexes x_i and x_j. Then based on K nearest neighbor graph (Knn), the element of weighted matrix Δ is denoted as:

$$\Delta(i, j) = \frac{W(i, j)}{\sum_{x_k \in Knn(x_i)} W(i, k)} \tag{1}$$

$$s.t. \quad W(i, j) = \begin{cases} \dfrac{h(\frac{\rho(x_i, x_j)^2}{\mu\sigma^2})}{\sum\limits_{x_k \in Knn(x_i)} h(\frac{\rho(x_i, x_k)^2}{\mu\sigma^2})} & x_j \in Knn(x_i) \\ 0 & otherwise \end{cases}$$

where h is a function with exponential decay at infinity, which is often $exp(-x)$. ρ is a distance measurement between two given samples. μ and δ are both hyper-parameters. Moreover, δ can be calculated by mean distance to Knn of x_i [28]. In the case of transductive learning, f is always denoted as: $\begin{bmatrix} \mathbf{f}_u \\ \mathbf{f}_l \end{bmatrix}$, in which $\mathbf{f}_l$ and $\mathbf{f}_u$ are label probabilities for labeled and unlabeled data respectively.

As the Sect. I exhibits, DCS aims at searching unlabeled samples that have kept stable correlations with its neighbors during feature space transformation by measuring intrinsic variation and category variation. Transductive GSSL is an ideal bridge to estimate the intrinsic structure among unlabeled samples. In the implementation of our DCS, we employ label propagation for transductive label inference (Please see more details in Sect. III.

2) Feature-learnable SSL (DSSL): DSSL for visual classification is usually categorized into two classes: reconstruction model and generation model. The former focuses on training deep model with reconstruction architecture in SSL manner [14]. It has a mirror architecture with encoding and decoding pathways like auto-encoder, and makes discrimination and unsupervised reconstruction for all data during the training phase. On the contrary, generation model achieves semi-supervised learning through creating data to classify. The generation model based methods start from deep generative network [29], and have received a great success with the development of generative adversarial network (GAN) [30], [31] and variational auto-encoder [32] . In fact, the idea of generation model is close to semi-supervised graphical model as we have mentioned above. In other words, both of them make inference and generation with discrimination. However, unlike pure graphical model, generation model shows more promising in generating data in continuous space (image and video).

Some researchers focus on combining neural networks and conventional methods. Liang et al. [33] formulated an incremental semi-supervised learning framework to train a network-based object detector via transferring knowledge from video. The incremental active learning technique by Lin [34], achieving an cost-effective labor in manual labeling, has recently received great attentions in the deep CNNs area for visual recognition. Weston et al. [16] invented deep semi-supervised embedding (DSSE) in the perspective of training neural network with graph-based regularity. Incorporating the graph relationship as a balancing loss into parameter updating, DSSE receives positive results with different configurations of network structure in many semi-supervised classification tasks. The approach can be treated as a GSSL variant in deep learning. Nevertheless, the relational graph describing locality among data, must be pre-defined before training. This premise is common in social network analysis [35]. However, in visual classification problem, delivering total relation information among all samples is actually a strong supervision and not an usual case.

B. Mult-view Learning (MVL)

Multi-view learning is a family of learning algorithms deriving from Co-Training [22], and focuses on exploiting data with multi-representation. For example, a cartoon character can

be represented by different views of the character like color histogram, skeleton and contour [11], and the views helps to select reliable samples and label them in supplementary way. The recent MVLs have extended to many kinds of application, e.g., clustering [36], reconstruction [37] and representation learning [38]. It is interesting that the design about Co-Space is similar to two views learning, which both views come from data feature extracted before and after the network fine-tuning. Both of them make the decision about labeling according to both views together. Nevertheless, the MVL usually results in two classifiers, which are implemented in the SSL case. In contrast, DCS proposes to perform sample mining via feature transforming in the chain of Co-Spaces. Those Co-Spaces come from a single neural network with different parameters, which are fine-tuned from an image database incrementally. Besides, in MVL setting, each view has to be independent to other views [39], but both views in Co-Space apparently correlate with each other in some way instead.

III. Deep Co-Space

In this section, we discuss the formulation of our proposed framework. In Sect. III-A, we introduce the pipeline of DCS, then the concept about Co-Space and feature transformation are defined. We leverage the feature transformation to formulate the sample mining strategy in Sect. III-B, which is most important part in DCS. Finally, further analysis about the strategy is discussed in Sect. III-C.

A. DCS architecture and Co-Space

In the context of visual classification, suppose that we have n samples taken from m classes for training. They are raw image data and we denotes the image database as $D = \{\mathbf{x}_i\}_{i=1}^{n}$. Then one-hot vector $\mathbf{y}_i = \{y_k\}_{k=1}^{m}$ represents the label for $\mathbf{x}_i$ and the $\mathbf{Y}$ is category set. In the setting of semi-supervised learning, only parts of images in D are labeled. For the simplicity in further discussion, we denote D^L as labeled images and D^U as unlabeled images respectively.

A CNN-based model f_θ is introduced to attain visual classifier and deep feature learning jointly, which f is the network architecture and θ means its parameter. The CNN model has been pre-trained by some large scale visual recognition database, which contributes some of visual semantics to the initial f. As the description about DCS in Sect. I, feature for each image is extracted iteratively and utilized to calculate the category consistency in feature transformation. Then the feature for image $\mathbf{x}$, which extracted from the network f in iteration t, is denoted as $f_{\theta_t}(\mathbf{x})$. (The output of $\mathbf{x}$ in f_θ is a result of classification. Since we don't use the classification result to explore sample in DCS, the $f_\theta(\mathbf{x})$ is treated as the output feature map/vector for $\mathbf{x}$ which extracted from f_θ in our formulation.)

We use θ_0 to present the pre-trained f parameter, and after the fine-tuning with D, f_{θ_0} leads to an updated model f_{θ_1} in the first iteration. In analogy, f_{θ_t} is the updated model in the t-th iteration, which have been fine-tuned from the model $f_{\theta_{t-1}}$ with renewed dataset augmented by sample mining result in previous iteration. Here we obtain the definition of Co-Space as follows:

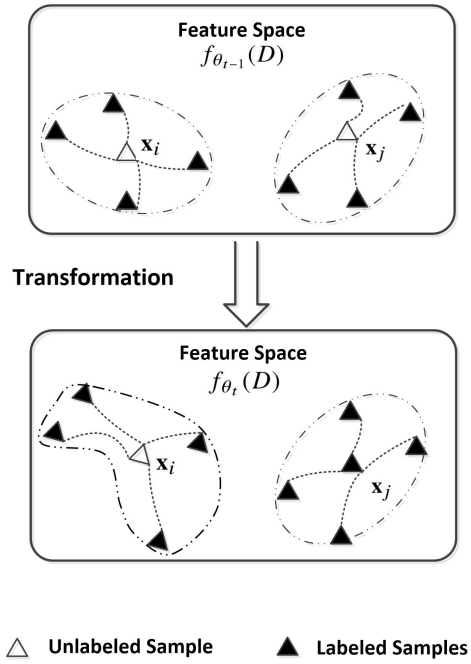

Fig. 2. We consider unlabeled instance $\mathbf{x}_i$ and $\mathbf{x}_j$ according to their neihborhoods. Area in dotted lines denotes the intrinsic structure around them. As we can see, instance $\mathbf{x}_i$ changes a lot in intrinsic structure, and its labeled neighbors also varies in transformation; and vice versa for instance $\mathbf{x}_j$. Eventually, instance $\mathbf{x}_j$ is selected and labeled.

Definition 1. *(**Co-Space**) Suppose $f_\theta(D)$ is a feature set for dataset D, which extracted from model f_θ. The couple of feature sets $<f_{\theta_{t-1}}(D), f_{\theta_t}(D)>$, is defined as the Co-Space of dataset D in the iteration t.*

Notice that in the definition above, Co-Space is generally interpreted as the construction in the t-th iteration. When t equals 1, a Co-Space is obtained as the description above; as DCS works, a set of unlabeled samples will be selected as pseudo-labeled candidates and used to retrain the f from θ_1 to θ_2, which leads to the next Co-Space $<f_{\theta_1}(D), f_{\theta_2}(D)>$, and then following this process of deduction. As we notice, dataset D is non-specific, which means D also representing any subset in the whole possible data space.

Using Co-Space, the feature transformation is defined as below:

Definition 2. *(**Feature Transformation**) Provided dataset D, $<f_{\theta_{t-1}}(D), f_{\theta_t}(D)>$ is a corresponding Co-Space in the iteration t. For all x belong to D, the projection $F_t^D : f_{\theta_{t-1}}(x) \rightarrow f_{\theta_t}(x)$ denotes the feature space transformation in the iteration t; and given specific sample $\boldsymbol{x}$, the feature transformation denotes as $F_t(\mathbf{x}) =< f_{\theta_{t-1}}(\mathbf{x}), f_{\theta_t}(\mathbf{x}) >$.*

Obviously, $F_t(\mathbf{x})$ is an one-to-one matching feature relation in Co-Space, thus for each $\mathbf{x}$, there is only one $F_t(\mathbf{x})$ acting as response. As for each unlabeled sample $\mathbf{x}$, DCS launches sample selection according to $F_t(\mathbf{x})$, which promising given an unlabeled sample $\mathbf{x}$, there is single one decision in sample mining.

Algorithm 1 Label Propagation with Knn [18]

Require:
Labeled dataset D^L and unlabeled dataset D^U; The label set Y^L corresponding to labeled dataset D^L; δ; μ; max iteration T.

Ensure:
Soft labels Y^U for unlabeled dataset.

1: Utilize Eq. (1) to initiate transition matrix P with D_L, D_U, δ and μ;
2: Initiate soft label set $Y_0 = [Y_0^L; Y_0^U]$, which $Y^U = \mathbf{0}$;
3: **for** t = 1 to T **do**
4: $[Y_t^L; Y_t^U] = P * [Y_{t-1}; Y_{t-1}]$;
5: $Y_t^L = Y^L$;
6: **end for**
7: $Y^U = Y_T^U$.

Both of the definitions compose the basis of our sample mining strategy. As an overview in brief, Fig. 2 demonstrates how to select reliable samples via feature transforming in Co-Space. In specific, sample $\mathbf{x}_i$ and $\mathbf{x}_j$ are both unlabeled samples we considering to select. In the definition, they have two feature expressions in Co-Space, corresponding to their feature transforming respectively. After the transformation, as for $\mathbf{x}_i$, the correlation between $\mathbf{x}_i$ and its neighbors largely change (intrinsic structure varies as the change of unlabeled neighbors; labeled local sample covariance varies as the change of labeled neighbors). In contrast, the category of $\mathbf{x}_j$ and its local samples keeps relatively stable. According to this leaked information, $\mathbf{x}_j$ is more preferable as a reliable candidate.

Specifically in each iteration, we use initial training set or training set augmented by pseudo-labeled sample, to update the model and obtain a new Co-Space. The Co-Space brings about feature transforming in dataset, which leading to the sample selection decision for each unlabeled samples. Then those selected samples are plugged into labeled data pool to replay the progressively semi-supervised learning process.

B. Sample Selection via Transforming Features

In the previous discussion, the change about local samples via feature transformation plays an important role to select reliable unlabeled samples. We attribute the change into two different variations. Firstly, as an off-the-shelf tool, label propagation algorithm is provided in Co-Space to assign a couple of soft labels for each sample. And the otherness between them, named as **neighborhood intrinsic variation**, is used to measure the change about intrinsic structure around each unlabeled sample. Secondly, we take a consideration in the labeled neighbors of unlabeled samples. The situation about local samples belong to each class are estimated in statistic, and its discrepancy in transformation is interpreted as **neighborhood category variation**. Finally, we incorporate both variations into an unified criterion to screen data.

1) Neighborhood Intrinsic Variation: In this subsection, neighborhood intrinsic variation will be formulated as the discussion below. We introduce the label propagation algorithm (LP), which is a key part of calculating the variation. As a wrapper algorithm, Eq. (1) in Sect. II is utilized to construct the transition matrix P. Then an original LP algorithm with Knn graph is demonstrated as Algorithm 1.

Algorithm 2 Neighborhood Intrinsic Variation

Require:
Labeled Co-Space $<f_{\theta_b}(D^L), f_{\theta_a}(D^L)>$ and unlabeled Co-Space $<f_{\theta_b}(D^U), f_{\theta_a}(D^U)>$, where θ_b and θ_a corresponding to mode parameter before/after updating; The label set Y^L corresponding to Co-Space $<f_{\theta_b}(D^L), f_{\theta_a}(D^L)>$; δ; μ; max iteration T.

Ensure:
Soft labels $Y_{\theta_b}^U$ and $Y_{\theta_a}^U$ for unlabeled dataset; low dimensional feature sets $f_{\theta_b}(D)$ and $f_{\theta_a}(D)$ for $D = D^U \cup D^L$.

1: Obtain $f_{\theta_b}(D) = f_{\theta_b}(D^L) \cup f_{\theta_b}(D^U)$;
2: Obtain $f_{\theta_a}(D) = f_{\theta_a}(D^L) \cup f_{\theta_a}(D^U)$;
3: Construct transition matrix $P_{\theta_b}(D)$ with $f_{\theta_b}(D)$ by Eq. (1);
4: Construct transition matrix $P_{\theta_a}(D)$ with $f_{\theta_a}(D)$ by Eq. (1);
5: Initiate soft label set $Y_{\theta_a}(0) = Y_{\theta_b}(0) = [Y^L; \mathbf{0}]$;
6: **for** t = 1 to T **do**
7: $Y_{\theta_b}(t) = P_{\theta_b}(D) * Y_{\theta_b}(t-1)$,
$Y_{\theta_a}(t) = P_{\theta_a}(D) * Y_{\theta_a}(t-1)$;
8: $Y_{\theta_b}^L(t) = Y_{\theta_a}^L(t) = Y^L$;
9: **end for**
10: $Y_{\theta_b}^U = Y_{\theta_b}^U(T)$, $Y_{\theta_a}^U = Y_{\theta_a}^U(T)$;

In the configuration we discuss, it needs some revision for Algorithm 1 to adapt to DCS. Firstly, since LP is a kind of fixed-feature transductive learning algorithm, D in Algorithm 1 has been default as an extracted feature set for data. As for our framework, CNN extract features on the fly, and is also updated in the progressively training process. It motivates us to use $f_\theta(D)$ instead of D. Secondly, the algorithm is launched in Co-Space, which need two feature spaces to attain two label propagation outputs for comparison. Using $<\tilde{f}_{\theta_b}(D), \tilde{f}_{\theta_a}(D)>$ as input, the adaptive LP in Co-Space is shown in Algorithm 2.

As the illustration in Algorithm 2, Co-Space leads to a couple of transition matrices $<P_{\theta_b}(D), P_{\theta_a}(D)>$ to predict soft label sets $Y_{\theta_b}^U$ and $Y_{\theta_a}^U$. Suppose normalized vector $\mathbf{y}_{\theta_b}(\mathbf{x}) \in R^m$ belongs to $Y_{\theta_b}^U$ and normalized vector $\mathbf{y}_{\theta_a}(\mathbf{x}) \in R^m$ belongs to $Y_{\theta_a}^U$. The non-consistency between $\mathbf{y}_{\theta_b}(\mathbf{x})$ and $\mathbf{y}_{\theta_a}(\mathbf{x})$, is performed as the neighborhood intrinsic variation in the transformation of sample $\mathbf{x}$, composing the sample mining criterion about to mention. (in the iteration t, the θ_b, θ_a refer to θ_{t-1}, θ_t). Since soft label is assigned through the intrinsic structure, which provided by an approximated manifold embedded in feature space [18] according to Knn graph. It implies that in neighborhood intrinsic variation, the change about local samples closer to $\mathbf{x}$ are more concerned.

2) Neighborhood Category Variation: Different in perspective about intrinsic variation, neighborhood intrinsic variation prefers to the change of category information. In specific, neighborhood category variation aims to find out which class with similarity changing most in statistic, through calculating the change in density of its labeled neighbors via feature

transforming. As for the sake of further discussion, we firstly introduce local labeled sample covariance matrix.

Specifically, we have a sample $\mathbf{x}$ and $f(\mathbf{x})$ is the corresponding feature. $N(f(\mathbf{x}))$ the neighborhood around $\mathbf{x}$, then the local sample covariance matrix for $f(\mathbf{x})$ is interpreted as follows:

$$\Sigma_{f(\mathbf{x})} = \frac{\sum\limits_{x' \in N(f(\mathbf{x}))} (x' - \mu_{f(\mathbf{x})})^T (x' - \mu_{f(\mathbf{x})})}{|N(f(\mathbf{x}))| - 1}.$$

where $|N(f(\mathbf{x}))|$ denotes how many local samples in $N(f(\mathbf{x}))$, and $\mu_{f(\mathbf{x})} = \frac{\sum\limits_{x' \in N(f(\mathbf{x}))} x'}{|N(f(\mathbf{x}))|}$ is the mean for local samples in $N(\mathbf{x})$. Considering different class belong to different distribution, we assume $\mathbf{x}$ is classified as y. Then $f(\mathbf{x})$'s labeled neighbors belong to class y denote as $N_y(f(\mathbf{x}))$ and the mean value in the neighborhood about class y is rewrited to $\mu_f^y(\mathbf{x}) = \frac{kf(\mathbf{x}) + \sum\limits_{\mathbf{x}' \in N_y(f(\mathbf{x}))} x'}{|N_y(f(\mathbf{x}))| + k}$. The local labeled sample covariance matrix of class y neighbors around $f(\mathbf{x})$ is defined as:

$$\Sigma^y_{f(\mathbf{x})} = \frac{k(f(\mathbf{x}) - \mu^y_{\mathbf{x}})^T (f(\mathbf{x}) - \mu^y_{\mathbf{x}}) + \sum\limits_{x' \in N_y(f(\mathbf{x}))} (x' - \mu^y_{\mathbf{x}})^T (x' - \mu^y_{\mathbf{x}})}{|N_y(f(\mathbf{x}))| + k - 1}.$$

where $f(\mathbf{x})$ is arranged as one part of its labeled neighbors, and we treat it as class y when covariance matrix $\Sigma^y_{f(\mathbf{x})}$ is considered. k is a weight to balance the importance between $f(\mathbf{x})$ and its y labeled neighbors.

The covariance matrix $\Sigma^y_{f(\mathbf{x})}$ captures the local geometry density and statistic about labeled samples, which in the area around $\mathbf{x}$ and belong to class y. After that, transformation distance is leveraged to measure the similarity between labeled neighbor in different classes. More specifically, for a sample $\mathbf{x}$ given class y, there is a Gaussian local distribution $p_y(f(\mathbf{x}))$ presenting as $\mathcal{N}(f(\mathbf{x}); 0, \Sigma^y_{f(\mathbf{x})})$; then providing feature transformation $F_t(\mathbf{x})$, transformation distance deriving from Hellinger distance, is calculated as follows:

$$\begin{aligned}\rho(F_t(\mathbf{x}), y; f_\theta) &\equiv H(p_y(f_{\theta_{t-1}}(\mathbf{x})), p_y(f_{\theta_t}(\mathbf{x}))) \\ &= \sqrt{1 - \frac{2^{D/2} |\Sigma^y_{f_{\theta_{t-1}}(\mathbf{x})}|^{1/4} |\Sigma^y_{f_{\theta(t)}(\mathbf{x})}|^{1/4}}{|\Sigma^y_{f_{\theta_{t-1}}(\mathbf{x})} + \Sigma^y_{f_{\theta_t}(\mathbf{x})}|^{1/2}}} \end{aligned} \quad (2)$$

where D is the dimensionality of the feature space, and the $|\Sigma^y_{f(\mathbf{x})}|$ means the determinant of matrix $\Sigma^y_{f(\mathbf{x})}$.

In the formulation above, a major problem comes from the computational complexity, which increasing in pace with the size of class number m. But thanks to the locality, we are just interested in an area around $\mathbf{x}$ and unnecessary to take all classes into account. In specific for image $\mathbf{x}$, we choose the intersection of its labeled neighborhoods before/after transformation. The major top-s categories in the intersection are considered by Eq. (2), and the top-s category set for image $\mathbf{x}$ denotes as:

$$\mathcal{Y}(F_t(\mathbf{x}), s) \subset \mathbf{Y}$$

Heuristically in implementation, we only choose a few classes (s less than 5, case by case) as the consideration in labeled neighborhood, then for other categories, the related Hellinger distances are set as infinity. We use h to denote exponential decay function $exp(-x)$ in DCS, then for a specific class y , Eq. (2) is reformulated to κ as:

Algorithm 3 Neighborhood Category Variation

Require:
Unlabeled data D^U, Co-Space $<f_{\theta_{t-1}}(D),\ f_{\theta_t}(D)>$, s.
Ensure:
Transformation matrix set $\{M_{f_\theta}(F_t(\mathbf{x}))|\mathbf{x} \in D^U\}$.
1: **for** i = 1 until $|D^U|$ **do**
2: Obtain $\mathcal{Y}(F_t(\mathbf{x}_i), s)$, where $\mathbf{x}_i \in D^U$ and $F_t(\mathbf{x}_i) \in <f_{\theta_{t-1}}(D),\ f_{\theta_t}(D)>$
3: **for** k = 1 until s **do**
4: For $y_k \in \mathcal{Y}(F_t(\mathbf{x}_i), s)$, obtain $\Sigma^{y_k}_{f_{\theta_{t-1}}(\mathbf{x}_i)}$ and $\Sigma^{y_k}_{f_{\theta_{t-1}}(\mathbf{x}_i)}$
5: Obtain $\rho(F_t(\mathbf{x}_i), y_k; f_\theta)$ via Eq. (2) from step 4
6: **end for**
7: Obtain $\{\kappa(F_t(\mathbf{x}_i), y; f_\theta)|y \in \mathbf{Y}\}$ via Eq. (3).
8: Obtain $M_{f_\theta}(F_t(\mathbf{x}_i))$
9: **end for**

$$\kappa(F_t(\mathbf{x}), y; f_\theta) = \begin{cases} \dfrac{h(\rho(F_t(\mathbf{x}), y; f_{\theta_t}))}{\sum\limits_{y' \in \mathcal{Y}(F_t(\mathbf{x}), s)} h(\rho(F_t(\mathbf{x}), y'; f_\theta))} & y \in \mathcal{Y}(F_t(\mathbf{x}), s) \\ 0 & otherwise \end{cases} \quad (3)$$

3) Sample Mining Criterion: In transformation about $\mathbf{x}$, the neighbors variation is estimated from two aforementioned points of view. Further, we assemble both variations into one criterion. We have a feature transformation matrix deriving from Eq. (3) as:

$$M_{f_\theta}(F_t(\mathbf{x})) = \begin{bmatrix} \kappa(F_t(\mathbf{x}),1;f_\theta) & 0 & \cdots & 0 \\ 0 & \kappa(F_t(\mathbf{x}),2;f_\theta) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \kappa(F_t(\mathbf{x}),m;f_\theta) \end{bmatrix} \quad (4)$$

where each row (column) in the matrix refers to a specific class in $\mathbf{Y}$, and the workflow shows in Algorithm 3.

Appending the result in Algorithm 2, we obtain the confidence function $R(\mathbf{x}; \theta_t)$ below, scoring the reliability for each unlabeled image $\mathbf{x}$:

$$R(\mathbf{x}; \theta_t) = r_b(\mathbf{x}; \theta_t)^T r_a(\mathbf{x}; \theta_t) \quad (5)$$

$$\begin{aligned} s.t.\ \mathbf{x} &\in D_t^U \\ r_b(\mathbf{x}; \theta_t) &= \frac{\sqrt{M_{f_\theta}(F_t(\mathbf{x}))}\mathbf{y}_{\theta_{t-1}}(\mathbf{x})}{|\sqrt{M_{f_\theta}(F_t(\mathbf{x}))}\mathbf{y}_{\theta_{t-1}}(\mathbf{x})|}; \\ r_a(\mathbf{x}; \theta_t) &= \frac{\sqrt{M_{f_\theta}(F_t(\mathbf{x}))}\mathbf{y}_{\theta_t}(\mathbf{x})}{|\sqrt{M_{f_\theta}(F_t(\mathbf{x}))}\mathbf{y}_{\theta_t}(\mathbf{x})|}; \end{aligned}$$

which implies the cosine similarity between vectors $r_b(\mathbf{x}; \theta_t)$ and $r_b(\mathbf{x}; \theta_t)$. Those samples with high score in Eq. (5) will be chosen, and get annotation by the rule as:

$$L(\mathbf{x}; \theta_t) = \underset{y}{\mathbf{argmax}}\{v(\mathbf{x}; \theta_t)\} \quad (6)$$

$$s.t.\ v(\mathbf{x};\theta_t) = r_b(\mathbf{x};\theta_t)\cdot r_a(\mathbf{x};\theta_t)$$

where $\cdot$ is the dot product for matrices (vectors) with same dimensions, **argmax**$\{v\}$ choose the biggest scalar value from the entries of vector v .

Function R is used to measure the consistency in unlabeled samples via feature transformation, in which each class in label set is considered and contributes to the confidence score in Eq. (5). If the value is more than a pre-defined threshold, the unlabeled sample will be selected and labeled as a class with largest contribution in Eq. (5).

There are M samples chosen for each iteration at most, and threshold th promise confidence score always bigger than a constant. The candidates with pseudo labels are used to enrich the labeled data pool and the cycle repeats as the progressive process in DCS.

C. Further Discussion about DCS

We discuss how the consistency is estimated through the criterion. As a careful observation in Eq. (5), reliability function R calculates the cosine distance between a couple of class re-weighted label propagation results in Co-Space, and the rebalanced weights for each classes, is provided by feature transformation matrix $M_{f_\theta}(F_t(\mathbf{x}))$. Our intuition comes from the noninformative problem about LP [40]. Regardless of feature transformation matrix, Eq. (5) degenerates to a simple cosine similarity between the couple of soft label in Co-Space, showing extremely unstable results in incrementally features transforming setting in our experiment. This is explained by the continuous augmentation of large scale training data, which triggering the noninformative problem. Feature transformation matrix helps the sample mining strategy focus on a few classes mainly acting on a provided sample, and restrain the redundant class information propagated through unstable relationship structure, which constructed from unlabeled samples represented by immature features.

In another point of view, we explain the strategy as measuring "micro-structure and "macro-structure around data. In "micro-structure about data in semi-supervised learning, data present as a manifold where their classes change smoothly [41]. Intrinsic structure about data presenting by $k$nn, captures the local property around the considered data instance $\mathbf{x}$. In case of that, we use a couple of soft labels to contrast the category change across intrinsic structures in different spaces. Differently, "macro-structure assuming data belong to same class should cluster together [42]. We use local labeled sample covariance of each class to perceive the geometric property change around $\mathbf{x}$ (density, shape and so on), and the class with steady geometric property is more preferable.

In the implementation of DCS, $f_\theta(D)$ as CNN-based features, are inappropriate as a direct input to calculate transition matrix $P_\theta(D)$. Due to in visual classification, $f_\theta(D)$ often coming from fully connected layer, the extracted feature is high-dimensional. Referring to the analysis in [40], GSSL algorithm applied in high-dimensional scenario inevitably runs into a common problem, leading to the value of label function for unlabeled sample constant almost everywhere in feature space.

Algorithm 4 Deep Co-Space (DCS) Sample Mining

Require:

Labeled image dataset D^L and unlabeled image dataset D^U; The label set Y^L corresponding to dataset D^L; CNN-based model f with a pre-trained parameter θ^o, pre-setted max iteration M.

Ensure:

well-trained CNN-based model f_{θ^*}; pesudo-labeled image set D^*.

initiate f by θ^o and obtain f_{θ_0}; $D_0^L = D^L$ and $Y_0^L = Y^L$; $D_0^U = D^U$; $D_0 = \{D_0^L, Y^L\} \cup D_0^U$; $D^* = \emptyset$;
Obtain θ_1 via fine-tuning f_{θ_0} with D_0;
for t = 1 until M **do**
  Obtain Co-Space $<f_{\theta_{t-1}}(D),\ f_{\theta_t}(D)>$ through feature extraction by $f_{\theta_{t-1}}$ and f_{θ_t};
  Utilize LargeVis [43] to $<f_{\theta_{t-1}}(D),\ f_{\theta_t}(D)>$, Obtain Co-Space $<\tilde{f}_{\theta_{t-1}}(D),\ \tilde{f}_{\theta_t}(D)>$;
  Runs Algorithm 2 to obtain soft label sets $Y^U_{\theta_{t-1}}$ and $Y^U_{\theta_t}$ for D_t^U;
  Runs Algorithm 3 to obtain feature transformation matrix set for D_t^U;
  Use Eq. (5) to select the top M samples with highest scores and annotate them in the principle of Eq. (6);
  $D_t^L = D_{t-1}^L \cup D_s$, $Y_t^L = Y_{t-1} \cup Y_s$, $D_t^U = D_{t-1}^U / D_s$, $D_t = \{D_t^L, Y_t^L\} \cup D_t^U$;
  Fine-tune CNN $f_{\theta_{t-1}}$ with D_t, obtain θ_t; $D^* = D_s \cup D^*$;
end for

An alternatives to compromise the problem is dimensionality reduction. However, linear reduction methods [44] decompose the local correlation among samples, and when we choose to preserve the locality [45], the computational complexity will be demanding. In the up-to-date related researches, LargeVis [43] is an ideal option in balance. As an innovative approach to make data visualization, LargeVis can also be treated as a locality preserving technique for dimensionality reduction. The computational complexity of LargeVis is $O(lMn)$ (n is the number of images we are about to cope with; M and l are the dimensionality of the original space and target space respectively.), keeping sample mining strategy computationally feasible in the incremental processing setting.

Then we discuss the computational complexity about DCS. Since DCS is an incremental learning framework to gradually process large scale data, we only observe one iteration in the cycle. The LargeVis achieves dimensionality reduction with complexity of $O(lMn)$, which linearly related to sample size n. Using LP algorithm within Co-Space in the generic style, the graph construction and propagation have a total time cost as $O(2n^2)$. Afterwards, in order to attain feature transformation matrix M, we runs Algorithm 3. It seems complicated but the total computation cost is $O(ns(N+k))$, where N is the maximal number of labeled neighbors for each unlabeled sample and k is the balancing weight in the calculation of $\Sigma^y_{f(\mathbf{x})}$. As the discussion regardless of considering feature extraction and the fine-tuning model, the bottleneck in complexity is the graph construction for LP. Actually, there exists a more scalable alternative to build relational graph [46] in linear n

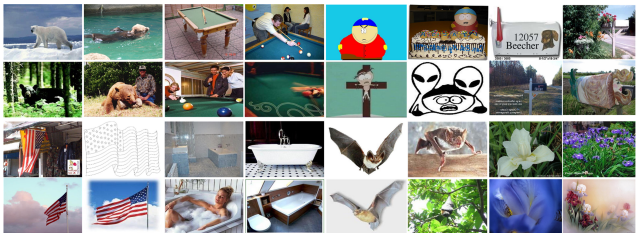

Fig. 3. Samples of images about Caltech-256. We selected 22,746/6,612 of RGBs for training and testing respectively.

Fig. 4. Samples of images about SUN-397. We chose a subset of them in the experiment setting.

time complexity. Moreover, inversely to the progressively data processing setting, we also sift out unlabeled samples selected previously, which improving the speed to account relationship between features in our DSSL experiment.

The work flow of DCS thoroughly shows in Algorithm 4

IV. Experiments

We evaluate our DCS in two different semi-supervised learning settings to validate its effectiveness in Sect. IV-A. Moreover, we further analyze the components of our DCS to clarify their contributions in Sect. IV-B.

A. Empirical Study

Experimental Setting Thanks to be independent of any specific network architecture, our DCS can be easily grafted to many different deep convolution-based models, and improve their performances in visual classification aided by large scale unlabeled images. To justify this, we conduct two experiments to evaluate DCS with two sorts of deep semi-supervised learning strategy. The first one is deep semi-supervised learning model (**DSSL**), in which the convolutional network architecture is invented to receive labeled images and unlabeled images in parallel. while the second one is standard supervised neural network model (**SSNN**), namely, the network must be trained with full supervision. Accordingly, we utilize labeled information to initiate SSNN models, then fine-tune them with an augmented labeled image set in a progressive style. Then in each iteration of DCS, the labeled image pool will be enriched by reliable pseudo-labeled samples, leading to an updated CNN for next iteration. We set the upper limit of labeled data augmentation as 1000 and the initial base learning rate as 0.001, then, all CNNs are updated via stochastic gradient decent algorithm with the momentum 0.9 and weight decay 0.0005. After the dimensionality reduction by LargeVis, the dimension in Co-Space is reduced to 15 before label propagation.

Fig. 5. Samples of web images that collected by the search engine. They are full of noise and we treat them as unlabeled images to expand the Caltech-256.

Since Co-Spaces are sequentially constructed in DCS, it requires the scalability about the LP algorithm in DCS. In our empirical experiment, δ is set 0.9 and the algorithm runs 50 times for each iteration. Besides, in k nearest labeled neighbors setting in Eq. (2), the closest 300 local labeled samples are considered and top 5 classes with most samples number will be selected to compute class-specific Eq. (6).

*1) **The Experiment of DSSL**:* The experiment is conducted on public object recognition benchmark Caltech-256 [1], which includes 30,607 images in total (Please see Fig. 3 for more details). We randomly select 80% images of each class as training data and the rest 20% images are treated as testing data, then there is 29,358 RGB images, which contain 22,746/6,612 for training and testing respectively.

Since the training data is insufficient in DSSL experiment, we construct an unlabeled dataset to address this issue. Specifically, we utilized python-tools web-crawler to collect images based on the keywords as all categories in Caltech-256, and images of which size less than 50*70 or larger than 500*700 are screened out (Please see Fig. 5 for more details). We select 100 candidates as unlabeled images for each class in the rest, and there are 25,600 images in total to expand the original Caltech-256. As for semi-supervised training, 40% samples in original Caltech-256 are selected as an initial labeled images pool, and the rest and web-crawled images are treated as unlabeled data in the progressive learning framework.

Implementation Details: we leverage stacked what-where auto-encoder (SWWAE) [14] as the architecture in implementation. Considering the mirror architecture in SWWAE, we take 16-layer VGG network as the encoding pathway. As for the decoder, we initiate decoding layers with Gaussian random noise, then make deep unsupervised learning to pre-train whole the architecture with ImageNet ILSVRC 2012 dataset (the encoder is fixed). In the pre-training and fine-tuning phase of DCS, spatial batch normalization [49] layers are leveraged to enhance the network performance for faster convergence. The balance weights in all reconstruction losses are coincident to 0.2, and the weight of discriminator loss is set 1.

Comparison and Analysis: We compare our SWWAE-based DCS with the original SWWAE. Besides, VGG sharing the

TABLE I
COMPARISON OF OUR RESULTS WITH SEVERAL COMPARISON METHODS ON CALTECH-256-VGG (DCS AND SWWAE HAVE BEEN AUGMENTED WITH WEB IMAGES)

Methods	VGG (100%labeled)	VGG-SWWAE	VGG-SWWAE-DCS
Percentage of labeled data	100%	18.8%	18.8%
Error rates	23.47%	29.42%	26,73%

TABLE II
COMPARISON OF OUR RESULTS WITH SEVERAL COMPARISON METHODS ON SUN-397-80-VGG

	SUN-397(40% labeled)	SUN-397(45% labeled)	SUN-397(50% labeled)
labeled-data-CNN	60.65%	67.89%	73.07%
ASL [47]	59.95%	66.62%	68.47%
YA [48]	47.77%	57.80%	64.80%
DSSE [16]	51.24%	57.48%	67.97%
DCS	**61.78**%	**69.66**%	**74.56**%

TABLE III
COMPARISON OF OUR RESULTS WITH SEVERAL COMPARISON METHODS ON CALTECH-256-ALEXNET (NO AUGMENTATION WITH WEB IMAGES)

	Caltech-256(40% labeled)	Caltech-256(45% labeled)	Caltech-256(50% labeled)
labeled-data-CNN	60.05%	63.87%	68.98%
ASL [47]	**60.96**%	62.22%	65.59%
YA [48]	57.98%	**64.80**%	69.10%
DSSE [16]	52.22%	63.48%	69.24%
DCS	59.12%	64.18%	**69.86**%

configuration with the encoding pathway in SWWAE, has also been adopted for comparison. This VGG is trained with 100% labeled samples in Caltech-256 without web images augmentation. Such experimental setting raise a question, whether we can improve the performance of deep semi-supervised learning model by adding unlabeled image samples. Table I includes the ratio of labeled/unlabeled images for training the corresponding models, and illustrates the comparison results based on error rates.

As one can see from Table I, DCS outperforms SWWAE, and even is close to the fully-supervised learning performance. This demonstrates that the performance of deep semi-supervised neural network can be enhanced by our DCS with unlabeled data. In further discussion, we note that unlabeled images from the Internet are often full with intra-class variation of the visual appearance, which tends to bring mild negative effects to the original SWWAE, which employs auto-encoder to reconstruct all unlabeled images to learn a latent expression. Besides, benefiting from the proposed sample mining criterion for reconstruction, SWWAE-based DCS shows the resistance of intra-class variation, and also reassuringly brings about more category information to obtain a clear performance gain.

*2) **The Experiment of SSNN**:* Under this experiment setting, we evaluate our DCS on two public visual classification databases: SUN-397 [2] and the original Caltech-256 [1] (no web image augmentation). SUN-397 is a large scale of images for scene categorization, whose image number across categories varies from 100 to over 2000. SUN-397 contains 397 classes and 108,754 images in total (Please see Fig. 4 for more details). Note that we only use its subset, which includes 80 classes for training and testing. All the datasets are split as training set and test set with a ratio 4:1 in all the following experiments. We account for 40%, 45% and 50% in the proportion of database as initial training samples respectively. To avoid only few samples for the certain category, we ensure the number of samples in any class more than 20.

Implementation Details: As for the network architecture, Vanilla Alexnet is implemented in the experiment about Caltech-256 without web images expansion, while VGG is applied in the subset of SUN-397. The corresponding parameters obtained on ImageNet ILSVRC 2012 dataset are used to initiate these two networks.

Comparison and Analysis: We compare our DCS with other incrementally SSL training frameworks: i) Yarowsky algorithm (YA) [48]. On the purpose of a significant comparison with this methods, we utilize deep learning architecture in YA same as DCS; ii) Deep learning via semi-supervised embedding (DSSE) [16]. DSSE tends to build the deep network with the relationship between samples. It is regrettable that no existing relation information is provided except for partially labeled data. Therefore, we make the modification of the DSSE and adapt the algorithm to the incremental learning experiment setting. Specifically, each couple of images with same label is viewed as a close relationship; then some unlabeled samples with high confidence (small entropy loss) at each iteration are assigned a corresponding label. The modification promises all unlabeled samples without relationship given are taken into consideration for training. Similarly, we use the convnet with the same configuration as DCS. 3). Adaptive semi-supervised learning (ASL) [47]. ASL is not a deep learning algorithm, yet still keeps well-performed in some visual recognition benchmarks. We take it as a conventional feature-fixed method for comparison, and let an aforementioned CNN models to

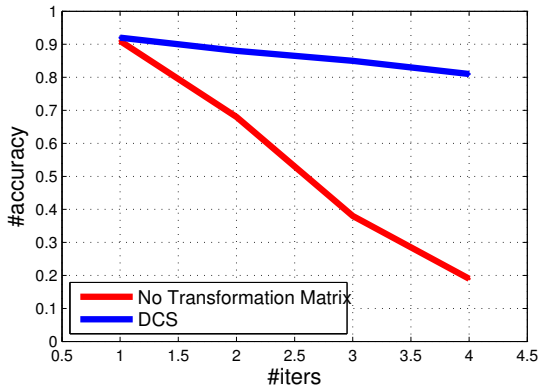

Fig. 6. The diagram above demonstrates the result of component analysis about the results with/without transformation matrix. Axis x and y denote how many times of the iteration and the corresponding label prediction accuracy of the selected unlabeled images. Red/Blue line comes from DCS/Eq. (7) in the DSSL experiment

extract features as input. Finally, labeled samples have been used to train a baseline model, which named "labeled-data-CNN" in the tables.

In the experiment of SUN-397 dataset, the results are shown in Table II. As we can see, when initial labeled images for each class is sufficient, DCS out-performs all the compared methods. This justifies the effectiveness of the our DCS. However, according to the results from Table III for the experiment with Caltech256, DCS obtains an unstable improvement (around 0.5% to 1.4%) in the comparison with the other methods. We explain the result in two reasons. Firstly, Caltech-256 shares some categories with ImageNet ILSVRC 2012 which have been used to pre-trained the deep model in the supervised learning style. It makes the performance of those classes stuck in bottleneck and constrain the whole dataset performance. Secondly, according to Eq. (2), each unlabeled sample has a feature transformation matrix based on its labeled neighbors. Small proportion of each class leads to less labeled neighbors for each unlabeled sample, and increases the variance in the calculation of transformation matrix. The setting with 50% labeled data expand the labeled neighbors, which relieve the problem and helps the model achieve a performance better than the other algorithms.

B. Component Analysis

The feature transformation matrix is a core in our DCS framework. For further demonstration of its contribution, we revise Eq. (5) and Eq. (6), and design a relevant component analysis with a new sample selection criterion as below:

$$R(\mathbf{x};\theta_t) = \frac{\mathbf{y}_{\theta_{t-1}}(\mathbf{x})^T \mathbf{y}_{\theta_t}(\mathbf{x})}{|(\mathbf{y}_{\theta_{t-1}}(\mathbf{x}))^T||\mathbf{y}_{\theta_t}(\mathbf{x})|} \tag{7}$$

$$s.t.\ \mathbf{x} \in D_t^U$$

$$L(\mathbf{x};\theta_t) = \underset{y}{\mathbf{argmax}}\{v(\mathbf{x};\theta_t)\} \tag{8}$$

$$s.t.\ v(\mathbf{x};\theta_t) = \mathbf{y}_{\theta_{t-1}}(\mathbf{x}) \cdot \mathbf{y}_{\theta_t}(\mathbf{x})$$

Specifically, the feature transformation matrix $M_{f_\theta}(F_t(\mathbf{x}))$ is replaced by identity matrix, meaning the new criterion with Eq. (7) and Eq. (8) chooses unlabeled samples only considering the consistency of soft labels in transformation.

The result has been demonstrated in Fig. 6. There seems no distinction between the two criteria at the first iteration. Both strategies achieve high accuracy in label prediction of selected samples, and the transformation matrix merely enhances the accuracy about 1%. But in the second iteration, the prediction accuracy in accordance with Eq. (7) rapidly decreases to 68%; and drastically falls down to 19% at the fourth iteration. In comparison, the DCS regularized by Eq. (5) remains accuracy above 80% till the fifth iteration. The phenomenon illustrates transformation matrix $M_{f_\theta}(F_t(\mathbf{x}))$ helps to maintain the selection quality and defer the semantic drift problem.

V. Conclusion

This paper proposes a novel semi-supervised learning framework named Deep Co-Space (DCS) to improve deep visual classification performance via an incrementally cost-effective manner. Considering deep feature learning as a sequence of steps pursuing feature transformation, DCS proposes to measure the reliability of each unlabeled image instance by calculating the category variations of the instance and its nearest neighbors from two different neighborhood variation perspectives, and merged them into an unified sample mining criterion deriving from Hellinger distance. Extensive experiments on standard image classification benchmarks demonstrate the effectiveness of the proposed DCS. In the future, we will pay more attention to extend our DCS to other vision tasks (e.g.,object detection and segmentation)

References

[1] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. *California Institute of Technology*, 2007.
[2] Jianxiong Xiao, Krista A. Ehinger, James Hays, Antonio Torralba, and Aude Oliva. Sun database: Exploring a large collection of scene categories. *International Journal of Computer Vision*, 119(1):1–20, 2014.
[3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25(2):2012, 2012.
[4] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.
[5] Liang Lin, Guangrun Wang, Rui Zhang, Ruimao Zhang, Xiaodan Liang, and Wangmeng Zuo. Deep structured scene parsing by learning with image descriptions. *arXiv preprint arXiv:1604.02271*, 2016.
[6] Xiaodan Liang, Si Liu, Xiaohui Shen, Jianchao Yang, Luoqi Liu, Jian Dong, Liang Lin, and Shuicheng Yan. Deep human parsing with active template regression. *IEEE transactions on pattern analysis and machine intelligence*, 37(12):2402–2414, 2015.
[7] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164, 2015.
[8] Xiaojin Zhu. Semi-supervised learning literature survey. *Computer Science*, 37(1):63–77, 2008.
[9] Andrew B Goldberg, Xiaojin Zhu, Aarti Singh, Zhiting Xu, and Robert D Nowak. Multi-manifold semi-supervised learning. In *AISTATS*, pages 169–176, 2009.
[10] Shilin Ding, Grace Wahba, and Xiaojin Zhu. Learning higher-order graph structure with features by structure penalty. In *Advances in Neural Information Processing Systems*, pages 253–261, 2011.

[11] Jun Yu, Meng Wang, and Dacheng Tao. Semisupervised multiview distance metric learning for cartoon synthesis. *IEEE Transactions on Image Processing*, 21(11):4636–4648, 2012.

[12] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[13] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. The cifar-10 dataset, 2014.

[14] J Zhao, M Mathieu, R Goroshin, and Y LeCun. Stacked whatwhere auto-encoders. *CoRR, vol. abs/1506.02351*, 2015.

[15] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[16] Jason Weston, Frédéric Ratle, Hossein Mobahi, and Ronan Collobert. Deep learning via semi-supervised embedding. In *Neural Networks: Tricks of the Trade*, pages 639–655. Springer, 2012.

[17] Yu-Feng Li and Zhi-Hua Zhou. Towards making unlabeled data never hurt. *IEEE transactions on pattern analysis and machine intelligence*, 37(1):175–188, 2015.

[18] Xiaojin Zhu, Zoubin Ghahramani, and Tommi Jaakkola Mit. Semi-supervised learning with graphs. In *International Joint Conference on Natural Language Processing*, pages 2465 – 2472, 2005.

[19] James R Curran, Tara Murphy, and Bernhard Scholz. Minimising semantic drift with mutual exclusion bootstrapping. In *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics*, volume 3, 2007.

[20] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[21] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Meeting on Association for Computational Linguistics*, pages 189–196, 1995.

[22] Kamal Nigam and Rayid Ghani. Analyzing the effectiveness and applicability of co-training. *Cikm*, 33(2):86–93, 2015.

[23] Xinlei Chen, Abhinav Shrivastava, and Abhinav Gupta. Neil: Extracting visual knowledge from web data. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1409–1416, 2013.

[24] Dong Zhang, Daniel Gatica-Perez, Samy Bengio, and Iain McCowan. Semi-supervised adapted hmms for unusual event detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 611–618. IEEE, 2005.

[25] Xian-Ling Mao, Zhao-Yan Ming, Tat-Seng Chua, Si Li, Hongfei Yan, and Xiaoming Li. Sshlda: a semi-supervised hierarchical topic model. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 800–809. Association for Computational Linguistics, 2012.

[26] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of machine learning research*, 7(Nov):2399–2434, 2006.

[27] Zhilin Yang, William Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. *arXiv preprint arXiv:1603.08861*, 2016.

[28] Bo Wang, Zhuowen Tu, and John K. Tsotsos. Dynamic label propagation for semi-supervised multi-class multi-label classification. In *IEEE International Conference on Computer Vision*, pages 425–432, 2013.

[29] Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589, 2014.

[30] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[31] Junbo Zhao, Michael Mathieu, and Yann LeCun. Energy-based generative adversarial network. *arXiv preprint arXiv:1609.03126*, 2016.

[32] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[33] Xiaodan Liang, Si Liu, Yunchao Wei, Luoqi Liu, Liang Lin, and Shuicheng Yan. Towards computational baby learning: A weakly-supervised approach for object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 999–1007, 2015.

[34] Liang Lin, Keze Wang, Deyu Meng, Wangmeng Zuo, and Lei Zhang. Active self-paced learning for cost-effective and progressive face identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[35] Hisashi Kashima, Tsuyoshi Kato, Yoshihiro Yamanishi, Masashi Sugiyama, and Koji Tsuda. Link propagation: A fast semi-supervised learning algorithm for link prediction. In *Proceedings of the 2009 SIAM international conference on data mining*, pages 1100–1111. SIAM, 2009.

[36] Abhishek Kumar, Piyush Rai, and Hal Daume. Co-regularized multi-view spectral clustering. In *Advances in neural information processing systems*, pages 1413–1421, 2011.

[37] Chang Xu, Dacheng Tao, and Chao Xu. Multi-view intact space learning. *IEEE transactions on pattern analysis and machine intelligence*, 37(12):2531–2544, 2015.

[38] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 945–953, 2015.

[39] Chang Xu, Dacheng Tao, and Chao Xu. A survey on multi-view learning. *arXiv preprint arXiv:1304.5634*, 2013.

[40] Boaz Nadler, Nathan Srebro, and Xueyuan Zhou. Semi-supervised learning with the graph laplacian: The limit of infinite unlabelled data. *Advances in neural information processing systems*, 21, 2009.

[41] Xiaojin Zhu, Zoubin Ghahramani, John Lafferty, et al. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, volume 3, pages 912–919, 2003.

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

[43] Jian Tang, Jingzhou Liu, Ming Zhang, and Qiaozhu Mei. Visualizing large-scale and high-dimensional data. In *Proceedings of the 25th International Conference on World Wide Web*, pages 287–297. International World Wide Web Conferences Steering Committee, 2016.

[44] Zohar Karnin and Edo Liberty. Online pca with spectral bounds. In *Proceedings of the 28th Annual Conference on Computational Learning Theory (COLT)*, pages 505–509, 2015.

[45] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[46] Wei Liu, Junfeng He, and Shih-Fu Chang. Large graph construction for scalable semi-supervised learning. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 679–686, 2010.

[47] De Wang, Feiping Nie, and Heng Huang. Large-scale adaptive semi-supervised learning via unified inductive and transductive model. In *SIGKDD*, pages 482–491, 2014.

[48] Gholam Reza Haffari and Anoop Sarkar. Analysis of semi-supervised learning with the yarowsky algorithm. 2012.

[49] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.