Transporting Labels via Hierarchical Optimal Transport for Semi-Supervised Learning

Fariborz Taherkhani, Ali Dabouei, Sobhan Soleymani, Jeremy Dawson, and Nasser M. Nasrabadi

West Virginia University
{ft0009, ad0046, ssoleyma} @ mix.wvu.edu, {jeremy.dawson, nasser.nasrabadi} @ mail.wvu.edu

Abstract. Semi-Supervised Learning (SSL) based on Convolutional Neural Networks (CNNs) have recently been proven as powerful tools for standard tasks such as image classification when there is not a sufficient amount of labeled data available during the training. In this work, we consider the general setting of the SSL problem for image classification, where the labeled and unlabeled data come from the same underlying distribution. We propose a new SSL method that adopts a hierarchical Optimal Transport (OT) technique to find a mapping from empirical unlabeled measures to corresponding labeled measures by leveraging the minimum amount of transportation cost in the label space. Based on this mapping, pseudo-labels for the unlabeled data are inferred, which are then used along with the labeled data for training the CNN. We evaluated and compared our method with state-of-the-art SSL approaches on standard datasets to demonstrate the superiority of our SSL method.

Keywords: Semi-Supervised Learning, Hierarchical Optimal Transport.

1 Introduction

Training a CNN model relies on large annotated datasets, which are usually tedious and labor intensive to collect [30]. Two approaches are usually considered to address this problem: Transfer Learning (TL) and Semi-Supervised Learning (SSL). In TL [51], learning of a new task is improved by transferring knowledge from a related task which has already been learned. However, in SSL [41], learning of a new task is improved by using information from an input distribution that is provided by a large amount of unlabeled data. To make use of the unlabeled data, it is assumed that the underlying distribution of this data follows at least one of the following structural assumptions: continuity, clustering, or manifold [12]. In the continuity assumption [61, 8, 36], data points close to each other are more likely to belong to the same class. In the clustering assumption [13, 61, 25], data tend to form discrete clusters, and data in the same cluster are more likely to share the same label. In the manifold assumption [10, 59], data lie approximately on a manifold of much lower dimension than the input space which can be classified by distances between probability measures on the manifold

[55]. To quantify the difference between two probability measures on a manifold properly, modeling the geometrical structures of the manifold is required [5, 4, 53]. One of the methodologies used to model geometrical structures on the probability simplex (i.e., manifold of discrete probability measures) is grounded on the theory of Optimal Transport (OT) [53, 46]. The Wasserstein distance, which arises from the idea of OT, exploits prior geometric knowledge of the base space in which random variables are valued [53]. Computing the Wasserstein distance between two random variables amounts to achieving a transportation plan which requires the minimal expected cost. The Wasserstein distance considers the metric properties of the base space in which a pattern is defined [5]. This characteristic of the Wasserstein distances has attracted a lot of attention for machine learning and computer vision tasks such as computing the barycenters [1, 2] of multiple distributions [50], generating data [7], designing loss function [21], domain adaptation [15, 27, 57, 18, 48, 32], and clustering [17, 28, 23, 37].

Data are usually organized in a hierarchical structure, or taxonomy. For example, considering a set of data belonging to the same class in a dataset as a measure, we can think of all the data in the dataset as a measure of measures. Inspired by OT, which maps two measures with the minimum amount of transportation cost, we can think of using hierarchical OT to map two measure of measures such that the total transportation cost across the measures becomes minimum. In this paper, we propose an SSL method that leverages from hierarchical OT to map measures from an unlabeled set to measures in a labeled set with a minimum amount of the total transportation cost in the label space.

Our method stems from two basic premises: 1) Data in a given class in the labeled and unlabeled sets come from the same distribution. 2) Assume we are given three measures with roughly the same amount of data, where only two of these measures come from the same distribution. The OT cost between two measures from the same distribution is expected to be less than the OT cost between one of these measures and the measure from a different distribution. Following these premises, we thus expect that the hierarchical OT maps measures from the same distribution in the labeled and unlabeled sets such that the total transportation cost between two measure of measures becomes minimum. Based on this mapping, a pseudo-label for unlabeled data in each measure from the unlabeled set is inferred. These unlabeled data annotated by pseudo-labels are then used along with the labeled data to train a CNN. However, data in the unlabeled set are not labeled to allow us to identify the measures. Thus, following the clustering assumption in SSL and the role of OT in clustering [17, 28, 23, 37], we can consider all the measures in the unlabeled set as a group of clusters which are identified by the Wasserstein barycenters of the unlabeled data.

2 Related Work

Pseudo-Labeling is one of the straightforward SSL techniques in which a model incorporates its own predictions on unlabeled data to achieve additional information during the training [33, 44, 20, 35, 24]. The main downside of these approaches is vulnerability to confirmation bias, i.e., they can not correct their own mistakes, when predictions of the model on unlabeled data are confident but incorrect. In such cases, the erroneous data can not contribute to the training and the error of the models is augmented during the training. This effect is intensified in cases where the distribution of the unlabeled data is different from that of labeled data. It has been shown that pseudo-labeling is practically similar to entropy regularization [42], in the sense that it forces the model to produce higher confident predictions for unlabeled data [33, 49]. However, in contrast to entropy regularization, it only forces these criteria onto data which have a low entropy prediction because of the threshold of confidence.

Consistency Regularization is considered as a way of using unlabeled data to explore a smooth manifold on which all of the data points are embeded [10]. This simple criterion has provided a set of methods , such as SWA [8], stochastic perturbations [45], π-model [31], Mean Teacher (MT) [52], and Virtual Adversarial Training (VAT) [38] that are currently considered as state-of-the-art for SSL. The original idea behind stochastic perturbations and π-model is pseudo-ensemble [9]. The pseudo-ensemble regularization techniques are usually designed such that the prediction of the model, $f_\theta(x)$, does not change significantly for realistic perturbed data ($x \to x'$). This goal is obtained by adding a loss term $d(f_\theta(x), f_\theta(x'))$ to the total loss of the model $f_\theta(x)$, where $d(.,.)$ is mean squared error or Kullback-Leibler divergence. The main downside of pseudo-ensemble methods, including π-model, is that they rely on a potentially unstable target prediction, which can immediately change during the training.

To address this issue, temporal ensembling [31] and MT [52], were proposed to obtain a more stable target output $f'_\theta(x)$. Temporal ensembling uses an exponentially accumulated average of outputs, $f_\theta(x)$, to make the target output smooth and consistent. Inspired by this method, MT uses a prediction function which is parametrized by an exponentially accumulated average of θ during the training. Similar to π-model, MT adds a mean squared error loss $d(f_\theta(x), f'_\theta(x))$ as a regularization term to the total loss function for training the network. It has been shown that MT outperforms temporal ensembling in practice [52]. In contrast to stochastic perturbation methods which rely on constructing $f_\theta(x)$ stochastically, VAT initially approximates a small perturbation r, and then adds it to x, which significantly changes the prediction of the model $f_\theta(x)$. Next, a consistency regularization technique is applied to minimize $d(f_\theta(x), f_\theta(x + r))$ with respect to θ which represents the parameters of the model.

3 Preliminaries

3.1 Discrete OT and Dual Form

For any $r \geq 1$, let the probability simplex be denoted by $\varDelta_r = \{q \in \mathbb{R}^r : q_i \geq 0, \sum_{i=1}^{r} q_i = 1\}$, and also assume that $X = \{x_1, ..., x_n\}$ and $X' = \{x'_1, ..., x'_m\}$ are two sets of data points in $\mathbb{R}^d$ such that $\mathcal{X} = \sum_{i=1}^{n} a_i \delta_{x_i}$ and $\mathcal{X}' = \sum_{i=1}^{m} b_i \delta_{x'_i}$ in which δ_{x_i} is a Dirac unit mass located on point x_i, and a, b are the weighting vectors which belong to the probability simplex $\varDelta_n$ and $\varDelta_m$, respectively.

Then, the Wasserstein-p distance $W_p(\mathcal{X}, \mathcal{X}')$ between two discrete measures $\mathcal{X}$ and $\mathcal{X}'$ is the p-th root of the optimum of a network flow problem known as the transportation problem [11]. The transportation problem depends on two components: 1) matrix $M \in \mathbb{R}^{n\times m}$ which encodes the geometry of the data points by measuring the pairwise distance between elements in X and X' raised to the power p, 2) the transportation polytope $\pi(a,b) \in \mathbb{R}^{n\times m}$ which acts as a feasible set, characterized as a set of $n \times m$ non-negative matrices such that their row and column marginals are a and b, respectively. This means that the transportation plan should satisfy the marginal constraints. In other words, let $\mathbf{1}_m$ be an m-dimensional vector with all elements equal to one, then the transportation polytope is represented as follows: $\pi(a,b) = \{T \in \mathbb{R}^{n\times m} | T^\top \mathbf{1}_n = b, T\mathbf{1}_m = a\}$. Essentially, each element $T(i,j)$ indicates the amount of mass which is transported from i to j. Note that in the transportation problem, the matrix M is also considered as a cost parameter such that $M(i,j) = d(x_i, x'_j)^p$ where $d(.)$ is the Euclidean distance.

Let $\langle T, M \rangle$ denote the Frobenius dot-product between T and M matrices. Then, the discrete Wasserstein distance $W_p(\mathcal{X}, \mathcal{X}')$ is formulated by an optimum of a parametric linear program $\mathbf{p}(.)$ on a cost matrix M, and $n \times m$ number of variables parameterized by the marginals a and b as follows:

$$W_p(\mathcal{X}, \mathcal{X}') = \mathbf{p}(a,b,M) = \min_{T\in\pi(a,b)} \langle T, M \rangle. \tag{1}$$

The Wasserstein distance in (1) is a Linear Program (LP) and a subgradient of its solution can be calculated by Lagrange duality. The dual LP of (1) is:

$$\mathbf{d}(a,b,M) = \max_{(\alpha,\beta)\in C_M} \alpha^\top a + \beta^\top b, \tag{2}$$

where the polyhedron C_M of dual variables is as follows: [11]

$$C_M = \{(\alpha, \beta) \in \mathbb{R}^{m+n} | \alpha_i + \beta_j \leq M(i,j)\}. \tag{3}$$

Considering LP duality, the following equality is established: $\mathbf{d}(a,b,M) = \mathbf{p}(a,b,M)$ [11]. Computing the exact Wasserstein distance is time consuming. To alleviate this, in [16], Cuturi has introduced an interesting method that regularizes (1) using the entropy of the solution matrix, $H(T)$, (i.e., $\min\langle T, M \rangle + \gamma H(T)$, where γ is regularization strength). It has been shown that if T'_γ is the solution of the regularized version of (1) and α'_γ is its dual solution in (2), then $\exists! u \in \mathbb{R}^n$, $v \in \mathbb{R}^m$ such that the solution matrix is $T'_\gamma = \text{diag}(u)K\text{diag}(v)$ and $\alpha'_\gamma = -\log(u)/\gamma + (\log(u)^\top \mathbf{1}_n)/(\gamma n))\mathbf{1}_n$ where, $K = exp(-M/\gamma)$. The vectors u and v are updated iteratively between step 1 and 2 by using the well-known Sinkhorn algorithm as follows: step $1: u = a/Kv$ and step $2: v = b/K^\top u$ [16].

3.2 Hierarchical OT

Let θ be a Polish space and $S(\theta)$ be the space of Borel probability measures on θ. Since θ is a Polish space, $S(\theta)$ is also Polish space and can be metrized by the

Wasserstein distance [40]. Considering the recursion of concepts, $S(S(\theta))$ is also a Polish space and is defined as a space of Borel probability measure on $S(\theta)$, which we can then define a Wasserstein distance on this space by using the Wasserstein metric in $S(\theta)$ (Section 3 in [40]). The concept of Wasserstein distance on the measure of measures, $S(S(\theta))$, which is also referred to as Hierarchical OT, is a practical and efficient solution to include structure in the regular OT distance [47, 3, 60, 34]. Hierarchical OT is used to model the data which are organized in a hierarchical structure, and has been recently studied for tasks such as multimodal distribution alignment [34], document representation [60], multi-level clustering [23] and a similarity measure between two hidden Markov models [14].

Let $\mathcal{D} = \{\mathcal{X}_1, \mathcal{X}_2, ..., \mathcal{X}_n\}$ and $\mathcal{D}' = \{\mathcal{X}'_1, \mathcal{X}'_2, ..., \mathcal{X}'_m\}$ be two sets of measures such that $\mathcal{M} = \sum_{i=1}^{n} r_i \delta_{\mathcal{X}_i}$ and $\mathcal{M}' = \sum_{i=1}^{m} s_i \delta_{\mathcal{X}'_i}$ in which $\delta_{\mathcal{X}_i}$ is a Dirac mass located on the measure $\mathcal{X}_i$, and r and s denote the weighting vectors belonging to the probability simplex Δ_n and Δ_m, respectively. Then, the hierarchical OT distance between $\mathcal{M}$ and $\mathcal{M}'$ can be formulated by a linear program as follows:

$$W'_p(\mathcal{M}', \mathcal{M}) = \min_{T' \in \pi'(r,s)} \sum_{i=1}^{n} \sum_{j=1}^{m} T'(i,j) W_p(\mathcal{X}_i, \mathcal{X'}_j), \tag{4}$$

where $\pi'(r,s) = \{T' \in \mathbb{R}^{m \times n} | T'^{\top} \mathbf{1}_m = r, T'^{\top} \mathbf{1}_n = s\}$, and $W_p(.,.)$ is the Wasserstein-p distance between two discrete measures $\mathcal{X}_i$ and $\mathcal{X}'_j$ which is obtained by Eq. (1). In Eq. (4), we have expanded Eq. (1) such that $T'(i,j)$ represents the amount of mass transported from $\delta_{\mathcal{X}_i}$ to $\delta_{\mathcal{X}'_j}$, and $W_p(.,.)$ is the ground metric which has been substituted by the Euclidean distance in Eq. (1) to represent hierarchical nature of the similarity metric between $\mathcal{M}'$ and $\mathcal{M}$.

3.3 Wasserstein Barycenters

Given $N >= 1$ probability measures with finite second moments $\{\mathcal{X}_1, \mathcal{X}_2, ..., \mathcal{X}_N\} \in S_2(\theta)$, their Wasserstein barycenters is a minimizer of F over $S_2(\theta)$ where [1]:

$$F(\tilde{\mathcal{X}}) = \inf_{\tilde{\mathcal{X}} \in S_2(\theta)} \frac{1}{N} \sum_{i=1}^{N} W_2^2(\tilde{\mathcal{X}}, \mathcal{X}_i). \tag{5}$$

In the case where $\{\mathcal{X}_1, ..., \mathcal{X}_N\}$ are discrete measures with finite number of elements, each with size e_i, the problem of finding Wasserstein barycenters $\tilde{\mathcal{X}}$ on the space of $S_2(\theta)$ in (5) is recast to search only on a simpler space $\mathcal{O}_r(\theta)$, where $\mathcal{O}_r(\theta)$ is the set of probability measures with at most r support points in θ, and $r = \sum_{i=1}^{N} e_i - N + 1$ [6]. There are fast and efficient algorithms that find local solutions of the Wasserstein barycenters problem over $\mathcal{O}_r(\theta)$ for $r \geq 1$, which the use of these algorithms for clustering has also been studied in [17, 58].

4 Method

Here, we describe our SSL model as is shown in Figure 1. Here, data belonging to the same class is defined as a measure. Thus, all the initial labeled data hierarchically are considered as a measure of measures. Similarly, all the unlabeled

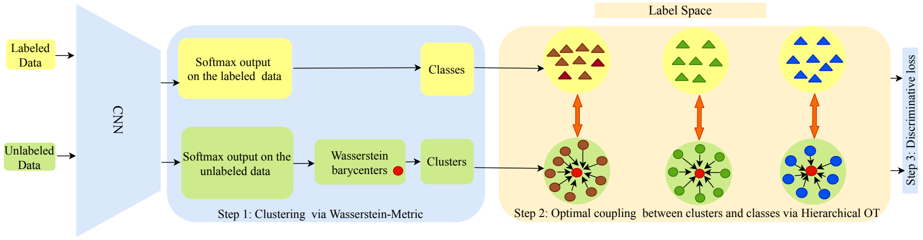

Labeled Data
Unlabeled Data
CNN
Softmax output on the labeled data
Classes
Softmax output on the unlabeled data
Wasserstein barycenters
Clusters
Step 1: Clustering via Wasserstein-Metric
Label Space
Step 2: Optimal coupling between clusters and classes via Hierarchical OT
Step 3: Discriminative loss

Fig. 1. At each epoch, a small amount of unlabeled data is processed through the current CNN and clustered into k groups. Then, the Wasserstein-2 distance is computed between theses groups and the ones formed by the labeled data. Next, a regularized OT is used to form an optimal coupling between the groups from the unlabeled data and the labeled ones, using the Wasserstein-2 distance as cost function (i.e., hierarchical OT). Finally, this coupling provides pseudo-labels for the selected unlabeled data to perform a gradient descent step on the CNN. Here, circles represent unlabeled data and triangles show the labeled data and their color indicate their labels.

data are also a measure of measures, each of which is constructed by data belonging to the same class. Following the basic premises mentioned earlier in the introduction, we use a hierarchical OT to predict pseudo-labels for the unlabeled measures to train a CNN. Our method is a three steps iterative algorithm. In the first step, we make a clustering assumption about the unlabeled data and consider all the unlabeled measures as a group of clusters which are identified by the Wasserstein barycenters of the unlabeled data. In the second step, we use the hierarchical OT to map each of the unlabeled measures to a corresponding labeled measure, based on which, a pseudo-label for the data within each of the clusters is predicted. Finally, unlabeled data annotated with pseudo-labels from the second step are used along with the initial labeled data to train the CNN.

4.1 Finding Unlabeled Measures via Wasserstein Metric

Given an image $z_i \in \mathbb{R}^{m\times n}$ from either the labeled or unlabeled dataset, CNN acts as a function $f(w, z_i) : \mathbb{R}^{m\times n} \rightarrow \mathbb{R}^c$ with the parameters w that maps z_i to a c-dimensional output, where c is number of the classes. Assume that $X = \{x_1, ..., x_m\}$ and $X' = \{x'_1, ..., x'_m\}$ are the sets of c-dimensional outputs represented by the CNN for the labeled and unlabeled images, respectively. Let $\mathcal{P}_i = 1/n_i \sum_{j=1}^{n_i} \delta_{x_j}$ denote a discrete measure constructed by labeled data in the i-th class, where δ_{x_j} is a Dirac unit mass on x_j and n_i is number of the data in the i-th class. Then, all of the labeled data form a measure of measures as follows: $\mathcal{M} = \sum_{i=1}^{c} \mu_i \delta_{\mathcal{P}_i}$, where $\mu_i = n_i/m$ represents amount of the mass in the measure $\mathcal{P}_i$ and $\delta_{\mathcal{P}_i}$ is a Dirac unit mass on the measure $\mathcal{P}_i$. Similarly, unlabeled data construct a measure of measures $\mathcal{M}' = \sum_{j=1}^{k} \nu_j \delta_{\mathcal{Q}_j}$, where each measure $\mathcal{Q}_i$, is created by unlabeled data belonging to the same class, $\nu_j = n'_j/m$ is amount of the mass in the measure $\mathcal{Q}_j$, and $\delta_{\mathcal{Q}_i}$ is a Dirac unit mass on $\mathcal{Q}_j$. However, data in the unlabeled set are not labeled to allow us to identify $\mathcal{Q}_j$.

Algorithm 1 : Finding Unlabeled Measures via Wasserstein Metric

```
input: $\mathcal{Q} \in \mathbb{R}^{c \times m}$.
initialize: $\mathcal{H} \in \mathbb{R}^{c \times k}$, $b = \mathbf{1}_m/m$, $t = 1$, $\eta = 0.5$.
while $\mathcal{H}$ and $a$ have not converged do
    Maximization Step:
    set $\hat{a} = \tilde{a} = \mathbf{1}_m/m$.
    while not converged do
        $\beta = (t+1)/2$, $a \leftarrow (1-\beta^{-1})\hat{a} + \beta^{-1}\tilde{a}$.
        $\alpha \leftarrow \boldsymbol{\alpha}'$: dual optimal form of OT $\mathbf{d}(a, b, M_{\mathcal{H}\mathcal{Q}})$.
        $\tilde{a} \leftarrow \tilde{a} \circ e^{-\beta\alpha}$; $\tilde{a} \leftarrow \tilde{a}/\tilde{a}^{\top}\mathbf{1}_n$.
        $\hat{a} \leftarrow (1-\beta^{-1})\hat{a} + \beta^{-1}\tilde{a}$, $t \leftarrow t+1$.
    end while
    $a \leftarrow \hat{a}$.
    Expectation Step:
    $T' \leftarrow$ optimal coupling for $\mathbf{p}(a, b, M_{\mathcal{H}\mathcal{Q}})$.
    $\mathcal{H} \leftarrow (1-\eta)\mathcal{H} + \eta(\mathcal{Q}T'^{\top})\mathrm{diag}(a^{-1})$, $\eta \in [0,1]$.
end while
```

One simple solution to find $\mathcal{Q}_j$, is to use the labels that are directly predicted by the CNN on the unlabeled data. In this case, there is no need to form unlabeled measures, since unlabeled data annotated by the CNN can be used directly for training the CNN. However, CNN as a classifier trained on a limited amount of the labeled data simply miss-classifies these unlabeled data. Thus, there is little option other than unsupervised methods, such as clustering to explore the unlabeled data belonging to the same class. This criterion stems from the structural assumption based on the clustering in SSL, where it is assumed that the data within the same cluster are more likely to share the same label. Inspired by the role of OT in clustering [17, 28, 23, 37], we leverage the Wasserstein metric to explore these measures underlying the unlabeled data. Specifically, we use the k-means objective incorporated by a Wasserstein metric loss to find $\mathcal{Q}_j$.

Given m unlabeled data $x'_1, ..., x'_m \in \theta$, the k-means clustering as a vector quantization method [43] aims to find a set C containing at most k atoms $c_1, ..., c_k \in \theta$ such that the following objective is minimized:

$$J(C) = \inf_{c_1,\ldots,c_k} \frac{1}{m} \sum_{i=1}^{m} ||x'_i - c_j||^2. \tag{6}$$

Let $\mathcal{Q} = \frac{1}{m}\sum_{i=1}^{m} \delta_{x'_i}$ be the empirical measure of data $x'_1, ..., x'_m$, where $\delta_{x'_i}$ is a Dirac unit mass on x'_i. Then, (6) is equivalent to exploring a discrete probability measure $\mathcal{H}$ including finite number of support points which minimizes:

$$F(\mathcal{H}) = \inf_{\mathcal{H} \in \mathcal{O}_k(\theta)} W_2^2(\mathcal{H}, \mathcal{Q}). \tag{7}$$

When $N = 1$ in (5), then (7) can also be considered as a Wasserstein barycenters problem whose solution is studied in [1, 17, 58]. From this perspective as studied by [17], the algorithm for finding the Wasserstein barycenters introduces an alternative for the popular Loyd's algorithm to find local minimum of the k-means objective, where the maximization step (i.e., the assignment of the weight of each data point to its closest centroid) is equivalent to computing $\boldsymbol{\alpha}'$ in dual form of OT (see Eq. (2)), while the expectation step (i.e., the re-centering step) is equivalent to updating $\mathcal{H}$ using the OT. Algorithm 1 presents clustering algorithm for exploring the unlabeled measures using the Wasserstein metric.

4.2 Mapping Measures via Hierarchical OT for Pseudo-Labeling

We design an OT cost function $f(.)$ to map the measures in $\mathcal{M}' = \sum_{j=1}^{k} \nu_j \delta_{\mathcal{Q}_j}$ to the measures in $\mathcal{M} = \sum_{i=1}^{c} \mu_i \delta_{\mathcal{P}_i}$ as follows:

$$f(\mu, \nu, G) = \min_{R \in \mathcal{T}(\mu,\nu)} \langle R, G \rangle - \omega H(R), \tag{8}$$

where R is the optimal coupling matrix in which $R(i,j)$ is amount of the mass that should be transported from $\mathcal{Q}_i$ to $\mathcal{P}_j$ to provide an OT plan between $\mathcal{M}'$ and $\mathcal{M}$. Thus, if the highest amount of the mass from $\mathcal{Q}_i$ is transported to $\mathcal{P}_r$ (i.e., $\mathcal{Q}_i$ is mapped to $\mathcal{P}_r$); the data belonging to the measure $\mathcal{Q}_i$ are annotated by r which is the label of the measure $\mathcal{P}_r$. Variable G is the pairwise similarity matrix between measures within $\mathcal{M}$ and $\mathcal{M}'$ in which $G(i,j) = W_p(\mathcal{Q}_i, \mathcal{P}_j)$ is the regularized Wasserstein distance between two clouds of data in $\mathcal{Q}_i$ and $\mathcal{P}_j$. Note that the ground metric used for computing $W_p(\mathcal{Q}_i, \mathcal{P}_j)$ is the Euclidean distance. Moreover, $\langle R, G \rangle$ is the Frobenius dot-product between R and G matrices, and $\mathcal{T}$ is transportation polytope defined as follows: $\mathcal{T}(\mu,\nu) = \{R \in \mathbb{R}^{c \times c} | R^{\top} \mathbf{1}_c = \nu, R\mathbf{1}_k = \mu\}$. Finally, $H(R)$ is entropy of the optimal coupling matrix R used for regularizing the OT, and ω is regularization strength in Eq. (8). The optimal solution for the regularized OT in (8) is obtained by Sinkhorn algorithm.

4.3 Training CNN in SSL Fashion

In the third step, we use the generic cross entropy as our discriminative loss function to train our CNN. Let $\{z_i\}_{i=1}^{b}$ be training batch annotated by true labels $\{y_i\}_{i=1}^{b}$, and $\{z_i'\}_{i=1}^{b}$ be training batch annotated by pseudo-labels $\{y_i'\}_{i=1}^{b}$, and c_i denotes barycenter of the cluster that sample z_i' belongs to it. Then, the total loss function $\mathcal{L}(.)$, used to train our CNN in an SSL fashion is as follows:

$$\mathcal{L}(w) = \sum_{i=1}^{b} \mathcal{L}_c(f(w, z_i), y_i) + \xi \Big(\sum_{i=1}^{b} \mathcal{L}_c(f(w, z_i'), y_i') + \frac{1}{b} \sum_{i=1}^{b} ||f(w, z_i') - c_i||^2 \Big), \tag{9}$$

where $f(w, z_i)$ is output of CNN for images z_i, and $\mathcal{L}_c(.)$ denotes cross entropy, and ξ is a balancing hyperparameter. Note that the third term in (9) is the center loss [56] which aims to reduce the distance between the unlabeled data and the barycenters of their corresponding cluster to perform a local consistency regularization [61]. For training, we initially train the CNN using the labeled data as a warm up step, and then use OT to provide pseudo-labels for the

Algorithm 2 : Transporting Labels via Hierarchical OT

```
input: LD: $Z_l = \{(z_l, y_l)\}_{l=1}^m$, UD: $Z_u = \{z'_u\}_{u=1}^n$, balancing coefficients: $\xi$, $\omega$, $\gamma$, learning rate: $r$, batch size: $b$, number of clusters: $k$.
 1: Train CNN parameters initially using the labeled data $Z_l$.
 2: repeat
 3:   Select $\{z'_i\}_{i=1}^m \subset Z_u$, where $m << n$.
 4:   Compute $X = \{x_l\}_{l=1}^m$, $X' = \{x'_u\}_{u=1}^m$: softmax output on $Z_l$ and $\{z'_i\}_{i=1}^m$, resp.
 5:   $\{\mathcal{Q}_1, ..., \mathcal{Q}_k\} \leftarrow$ cluster on $X'$ using Algorithm. 1.
 6:   $\{\mathcal{P}_1, ..., \mathcal{P}_c\} \leftarrow$ group $X$ to $c$ classes.
 7:   Compute $\mu_i$, $\nu_i$ based on the amount of the mass in $\{\mathcal{Q}_i\}_{i=1}^k$ and $\{\mathcal{P}_i\}_{i=1}^c$.
 8:   for each $\mathcal{Q}_i$ and $\mathcal{P}_j$ do
 9:     $G(i,j) \leftarrow W_2(\mathcal{Q}_i, \mathcal{P}_j)$: using regularized OT.
10:   end for
11:   $R \leftarrow$ optimal coupling for $f(\mu, \nu, G)$ in Eq. (8): using regularized OT.
12:   $\{y'_u\}_{u=1}^m \leftarrow$ pseudo-label each cluster $\mathcal{Q}_i$ based on the highest amount of mass transport toward the labeled measure (i.e., argmax $R(i,:)$).
13:   repeat
14:     Select a mini-batch:$\{z_i, z'_i\}_{i=1}^b \subset (\{z'_i\}_{i=1}^m \cup Z_l)$.
15:     $w \leftarrow w - r\nabla_w[\mathcal{L}(w)]$, using Eq. (9).
16:   until for an epoch
17: until a fixed number of epochs
```

unlabeled data to train the CNN along with the labeled data for the next epochs. Specifically, after training the CNN using the labeled data, in each epoch, we randomly select the same amount of labeled data from the pool of unlabeled data to compute their pseudo-labels via hierarchical OT. Then, the CNN is trained in a mini-batch mode. Our overall SSL method is described in Algorithm 2.

Discussion on Time Complexity: Algorithm 2 has two main computational parts: 1) clustering the unlabeled data via Algorithm 1 (i.e. line 5), and 2) mapping the measures via HrOT (i.e. lines 8-11). For part (1): we used [17] and there is an analysis for its Time Complexity (TC) in [58] as follows: Let c, n, k, and i be the number of classes, unlabeled data, barycenters, and iterations in EM for Algorithm 1, resp. Based on the analysis provided in [58], in our case, $N = 1$ (# of distributions), $d = c$ (dimension) and $\tau = 1$ (adjusting the support points for barycenters every τ iterations). Thus, TC of part (1) is $\mathcal{O}(ink) + \mathcal{O}(inck) \approx \mathcal{O}(inck)$. For part (2): Since TC of computing the regularized OT distance between two sets of data each with size m is $\mathcal{O}(m^2)$ [22], then we need $\mathcal{O}((n/c)(n/k)(ck)) = \mathcal{O}(n^2)$ to compute matrix G in lines 8-10, and also we need $\mathcal{O}(ck)$ to find R in line 11. Thus, TC of part (2) is $\mathcal{O}(ck) + \mathcal{O}(n^2) \approx \mathcal{O}(n^2)$. By summing part(1) and (2), the total TC for inferring spudo-labels on n data is $\mathcal{O}(n^2 + inck)$. Note that i is not large due to smoothing[17].

5 Experiments and Setup

Setup: we conduct our experiments on SVHN [39], CIFAR-10/100 [29], and Mini-ImageNet [54] datasets. Mini-ImageNet [54] is subset of ImageNet [19]

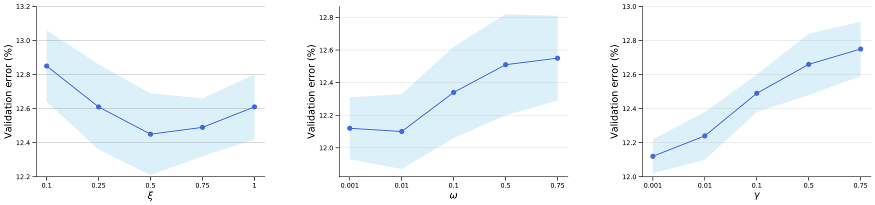


Fig. 2. Validation error for hyperparameter tuning on CIFAR-10.

which consists of 100 classes and 600 images per class. For Mini-ImageNet, we follow the SSL setup in [24]. We use 500 and 100 images per class for the training and testing splits, res. Following the prior works in [31, 52, 24, 8], we use a ResNet-18 network for the Mini-ImageNet, and a “13-layer” network for the SVHN and CIFAR-10/100 to evaluate our model. We use the typical configuration for SVHN and CIFAR-10/100 [52], and the same for the Mini-ImageNet, i.e., normalizing images using dataset mean and Standard Deviation (SD) together and then performing data augmentation by random horizontal flips and random 4 pixel translations [52]. For training, we use Adam optimizer [26] with a learning rate of 3×10^{-3} and the batch size is set to 128. The stopping criteria for the Sinkhorn algorithm is either maxIter = 10,000 or tolerance = 10^{-8}, where maxIter is the maximum number of iterations and tolerance is a threshold for the integrated stopping criterion based on the marginal differences. The barycenters in Algorithm. 1 are initially set to centroids obtained by k-means.

We follow suggested guidelines in [41] to evaluate our model. 1) We report performance of a fully-supervised baseline since the purpose of SSL is to significantly improve the fully-supervised baseline. 2) We vary the amount of labeled data when reporting the accuracy of our SSL method since a perfect SSL algorithm should remain effective even with small amount of labeled data. 3) We compare our method with the case where the unlabeled data are labeled by CNN using its own prediction during the training. 4) We study performance of the soft-pseudo-labels which can be generated by our model. 5) We compare the OT-based clustering method described in Algorithm. 1 vs k-means to show the effectiveness of Wasserstein metric for finding the unlabeled measures in our model. 7) We conduct ablation studies on the clustering resolution, and also we study effect of the center loss as a local consistency regularizer in our SSL model.

Hyperparameter Tuning: Following [41, 8], we use a validation set of 5k images for CIFAR-10/100, and standard validation set of 7k images for SVHN to tune hyperparameters of our model. For CIFAR-10 and SVHN, we use 1k labeled data, and for CIFAR-100, we use 4k labeled data. The results shown in Figure 2 are the mean and SD of error rate on validation set for CIFAR-10. Similarly, for other datasets, we also tuned the hyperparameters. For SVHN, the values chosen for γ, ω and ξ are 0.01, 0.001, and 0.75, resp. For CIFAR-100, the values chosen for γ, ω and ξ are 0.001, 0.001, 0.5, res. For Mini-ImageNet, we used the same hyperparameters tuned for CIFAR-100.

Dataset	**CIFAR-10**			**SVHN**		
Labels	1000	2000	4000	250	500	1000
Supervised	45.89 ± .97	32.14 ± 0.84	21.79 ± 0.23	43.58 ± 1.98	23.78 ± 0.94	14.83 ± 0.79
TDCNN [49]	32.67 ± 1.93	22.99 ± 0.79	16.17 ± 0.37	22.90 ± 1.91	13.79 ± 1.24	8.77 ± 0.82
VAT [38]	-	-	11.36	-	-	5.42
π model [31]	-	-	12.36 ± 0.31	-	6.65 ± 0.53	4.82 ± 0.17
Temporal Ens [31]	-	-	12.16 ± 0.24	-	5.12 ± 0.13	4.42 ± 0.16
MT [52]	19.04 ± 0.51	14.35 ± 0.31	11.41 ± 0.25	4.35 ± 0.50	4.18 ± 0.27	3.95 ± 0.19
LP [24]	22.02 ± 0.88	15.66 ± 0.35	12.69 ± 0.29	-	-	-
LP+MT [24]	16.93 ± 0.70	13.22 ± 0.29	10.61 ± 0.28	-	-	-
SWA [8]	15.58	11.02	9.05	-	-	-
DOT	17.97 ± 0.47	14.46 ± 0.55	11.84 ± 0.20	5.14 ± 0.23	4.74 ± 0.35	4.11 ± 0.26
HrOT (k-means)	15.78 ± 0.65	13.16 ± 0.58	10.94 ± 0.32	4.89 ± 0.27	4.14 ± 0.30	3.86 ± 0.24
HrOT w/o CL	13.65 ± 0.36	10.44 ± 0.39	9.02 ± 0.44	4.82 ± 0.25	4.06 ± 0.24	3.61 ± 0.15
Soft HrOT	12.58 ± 0.34	9.56 ± 0.37	8.14 ± 0.49	4.32 ± 0.28	3.77 ± 0.21	3.55 ± 0.12
HrOT	**11.91 ± 0.25**	**8.87 ± 0.32**	**7.74 ± 0.28**	**4.19 ± 0.16**	**3.52 ± 0.23**	**3.06 ± 0.09**

Table 1. Comparing test error between HrOT and different baselines and SSL methods.

5.1 Fully Supervised and Deep SSL Methods

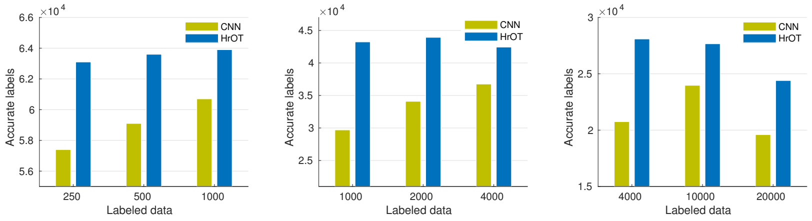
We report the error rate of the "13-layer" CNN on CIFAR-10/100 and SVHN datasets and ResNet-18 on the Mini-ImageNet dataset for both cases where we only use the labeled data (i.e., Supervised in Table. 1 & 2), and the case where we leverage the unlabeled data using the hierarchical OT technique during the training (i.e., HrOT in Table. 1 & 2). All of the compared SSL methods in Table. 1 & 2 use a common CNN architecture. Following the prior works [31, 52, 24, 8], to compare HrOT with other SSL algorithms, we selected the same amount of data in the training set as the labeled data and the remaining as the unlabeled data for SVHN (73k), CIFAR-10/100 (50k) and Mini-ImageNet (50k) datasets. We run our SSL algorithm over 5 times with different random splits of labeled and unlabeled sets for each dataset, and we report the mean and SD of the test error rates. The results in Table. 1 & 2 indicate the potential of our model for leveraging the unlabeled data in comparison to other SSL methods.

5.2 Soft-Pseudo-Labels based on Hierarchical OT

Other than particular manner in HrOT where we choose one particular pseudo-label based on the highest amount of mass transport from an unlabeled measure to a labeled measure to label the unlabeled measure, we also use "soft pseudo-labels" for training the CNN. In other words, instead of having one-hot target in the usual classification loss, we use the row of the transportation mass corresponding to the labeled measures as the target. The compared result in Table. 1 & 2 show that using one-hot targets (HrOT) outperforms using soft pseudo-labels (Soft-HrOT). The reason can be supported by SSL methods based on the entropy minimization [42]. This set of methods forces the model to produce confident predictions (i.e., low entropy). Similarly, once we use one-hot targets, we essentially encourage the network to produce more confident predictions compared to using soft-pseudo labels.

Datasets	**CIFAR-100**		**Mini-ImageNet-top1**		**Mini-ImageNet-top5**	
Labels	4000	10000	4000	10000	4000	10000
Supervised	55.89 ± 0.26	41.07 ± 0.33	75.94 ± 0.41	61.59 ± 0.69	53.85 ± 0.46	38.59 ± 0.53
LP [24]	46.20 ± 0.76	38.43 ± 1.88	70.29 ± 0.81	57.58 ± 1.47	47.58 ± 0.94	36.14 ± 2.19
MT [24]	45.36 ± 0.49	36.08 ± 0.51	72.51 ± 0.22	57.55 ± 1.11	49.35 ± 0.22	32.51 ± 1.31
LP+MT [24]	43.73 ± 0.20	35.92 ± 0.47	72.78 ± 0.15	57.35 ± 1.66	50.52 ± 0.39	31.99 ± 0.55
SWA [8]	-	34.10 ± 0.31	-	-	-	-
DOT	44.28 ± 0.47	36.82 ± 0.33	73.84 ± 0.44	59.26 ± 0.52	48.22 ± 0.75	32.14 ± 0.48
HrOT (k-means)	42.06 ± 0.62	35.57 ± 0.64	72.04 ± 0.35	58.09 ± 0.43	46.47 ± 0.83	31.48 ± 0.33
HrOT w/o CL	40.66 ± 0.71	32.88 ± 0.36	68.94 ± 0.51	55.77 ± 0.83	44.97 ± 0.54	29.18 ± 0.26
Soft HrOT	40.02 ± 0.84	31.76 ± 0.31	68.49 ± 0.63	54.73 ± 0.70	44.16 ± 0.26	28.19 ± 0.24
HrOT	$\mathbf{38.98 \pm 0.91}$	$\mathbf{30.86 \pm 0.56}$	$\mathbf{67.66 \pm 0.75}$	$\mathbf{53.79 \pm 0.46}$	$\mathbf{43.38 \pm 0.39}$	$\mathbf{27.45 \pm 0.59}$

Table 2. Comparing test error between HrOT and different baselines and SSL methods.

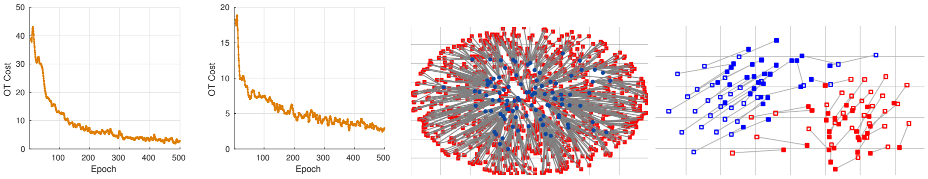


(a) SVHN (b) CIFAR-10 (c) CIFAR-100

Fig. 3. indicate the number of accurate predicted labels by HrOT and CNN.

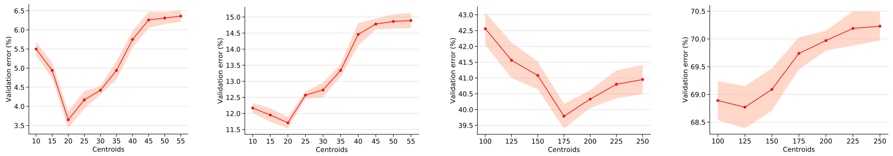
5.3 Contribution of Hierarchical Optimal Transport to SSL

CNN trained on a limited amount of the labeled data simply miss-classifies the unlabeled data. Instead, we use the OT to cluster the unlabeled data and then map them with the labeled measures for pseudo-labeling. To compare these two criteria for pseudo-labeling, we report the number of accurate pseudo-labels obtained for the unlabeled training data using HrOT and the CNN by its own prediction (i.e., ”13 layer” network). We experimentally show how HrOT has a greater positive influence on the training of CNN classifier. Essentially, this comparison allows us to know whether or not the CNN classifier can benefit from our method for producing pseudo-labels during the training, because, otherwise, the CNN can simply use its own predicted labels on the unlabeled training data during the training. To indicate the efficiency of HrOT, we change the number of labeled data in the training set and report the number of accurately predicted pseudo-labels by our CNN, and HrOT on the remaining unlabeled training data. Figure 3(a)-3(c) show that, for SVHN and CIFAR-10/100, the labels predicted by HrOT on the unlabeled training data are more accurate than the CNN, which means that the entire CNN can better benefit from HrOT than the case where it is trained solely by its own predicted labels. Moreover, we monitored the trend of transportation cost between the labeled and unlabeled measures obtained by Eq. 8 during the training. Experiments on SVHN and CIFAR-10 in Figure 4(a) and Figure 4(b) show that this cost reduces as the images fed into



(a) CIFAR-10 (b) SVHN (c) Barycenters (d) Mapping measures

Fig. 4. (a, b) OT cost trend, (c, d) mapping measures to barycenters and each other.



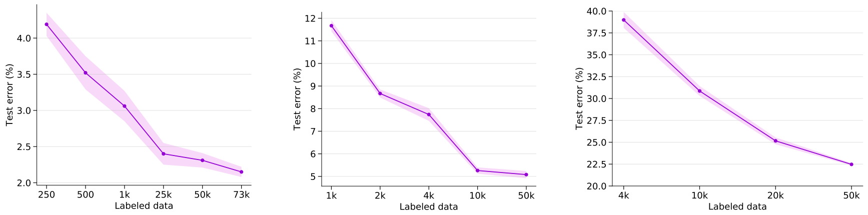
(a) SVHN (b) CIFAR-10 (c) CIFAR-100 (d) Mini-ImageNet

Fig. 5. Validation error for different clustering resolution.

the CNN are represented by a better feature set during the training. In Figure 4(c) and Figure 4(d), we also visualized barycenters of the clusters, and mapping between measures of two classes in CIFAR-10 (i.e., bird and frog) using Sinkhorn algorithm. The Figures show that measures of different classes are separated properly after the training. Here, filled and unfilled squares represent unlabeled, and labeled data, respectively and the color of squares indicates their label.

5.4 Clustering Resolution

We study effect of the clustering resolution on the performance of our model. We use 1k labeled data for SVHN and CIFAR-10, and 4k labeled data for CIFAR-100 and MiniImageNet datasets. We change number of the centriods during the clustering, and report the error on the validation set. The results in Figure 5(a)-5(d) indicates that our model benefits from over-clustering but intense over-clustering decreases performance. The rationale can be supported by SSL models based on consistency regularization [61, 36]. Specifically, if we intensively increase the number of the clusters, we only consider the global structure (or geometry) of the data in the label space, and then we ignore their local structure when transporting labels. This is not appropriate for SSL, since in such a case, we ignore the local consistency of data. However, if we cluster data in the label space via Wasserstein metric and then map them through HrOT, we exploit both local and global structure of the data in the label space during the transporting labels. To further validate this claim, we conducted following ablation study: in each batch, we solve an OT between labeled data $\{x_l\}_{u=1}^{m}$ and unlabeled data $\{x'_u\}_{u=1}^{m}$ directly and use the OT to assign pseudo-labels to the unlabeled data. We refer to this baseline as Direct-OT (DOT), our results in Table 1 & 2 indicate



(a) SVHN (b) CIFAR-10 (c) CIFAR-100

Fig. 6. Performance of HrOT by varying the labeled data.

that mapping data directly using OT significantly reduces the performance of the CNN compared to our original model, HrOT which indicate importance of the clustering and considering hierarchical structure in OT for generating pseudo-labels in our SSL model.

In further study, instead of using OT to cluster, we use the regular k-means in our method. We refer to this baseline as HrOT (k-means). The compared results between HrOT and HrOT(k-means) in Table . 1 & 2 shows the power of Wasserstein-metric in the k-means objective for finding unlabeled measures. Moreover, we ablated the center loss in (Eq. 9) to see the effect of this term as a local consistency regularizer in our SSL model (i.e., HrOT w/o CL in in Tables 1 & 2). The compared results with HroT in Tables 1 & 2 show that this term can have relatively a positive influence on the performance of our model.

5.5 Varying Labeled Data

We evaluate how varying the amount of initial labeled data degrades the performance of HrOT in the very limited label regime. We gradually increase the number of labeled data during the training and report the performance of our SSL method on the testing set. Here, we run our SSL method over 5 times with different random splits of labeled and unlabeled sets for SVHN and CIFAR-10/100, and report the results in Figure 6(a)-6(c). The results show that the performance of HrOT tends to level off as the number of labels increases.

6 Conclusion

We proposed a method which leverages optimal transport to train a CNN classifier in an SSL manner. We used the Wasserstein barycenters of the unlabeled data to identify the measures in the unlabeled set. Then, we used hierarchical optimal transport to map measures from the unlabeled set to measures in the labeled set with a minimum amount of the total transportation cost in the label space. Based on this mapping, pseudo-labels for the unlabeled data were inferred, which were then used along with the labeled data for training the CNN. Finally, we experimentally evaluated our SSL method to indicate its potential for leveraging the unlabeled data when labels are limited during the training.

References

1. Agueh, M., Carlier, G.: Barycenters in the wasserstein space. SIAM Journal on Mathematical Analysis **43**(2), 904–924 (2011)
2. Álvarez-Esteban, P.C., del Barrio, E., Cuesta-Albertos, J., Matrán, C.: A fixed-point approach to barycenters in wasserstein space. Journal of Mathematical Analysis and Applications **441**(2), 744–762 (2016)
3. Alvarez-Melis, D., Jaakkola, T., Jegelka, S.: Structured optimal transport. In: International Conference on Artificial Intelligence and Statistics. pp. 1771–1780 (2018)
4. Amari, S.i.: Information geometry and its applications, vol. 194. Springer (2016)
5. Amari, S.i., Karakida, R., Oizumi, M.: Information geometry connecting wasserstein distance and kullback–leibler divergence via the entropy-relaxed transportation problem. Information Geometry **1**(1), 13–37 (2018)
6. Anderes, E., Borgwardt, S., Miller, J.: Discrete wasserstein barycenters: optimal transport for discrete data. Mathematical Methods of Operations Research **84**(2), 389–409 (2016)
7. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein gan. arXiv preprint arXiv:1701.07875 (2017)
8. Athiwaratkun, B., Finzi, M., Izmailov, P., Wilson, A.G.: There are many consistent explanations of unlabeled data: Why you should average. In: International Conference on Learning Representations (2019)
9. Bachman, P., Alsharif, O., Precup, D.: Learning with pseudo-ensembles. In: Advances in Neural Information Processing Systems. pp. 3365–3373 (2014)
10. Belkin, M., Niyogi, P., Sindhwani, V.: Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. Journal of machine learning research **7**(Nov), 2399–2434 (2006)
11. Bertsimas, D., Tsitsiklis, J.N.: Introduction to linear optimization, vol. 6. Athena Scientific Belmont, MA (1997)
12. Chapelle, O., Scholkopf, B., Zien, A.: Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. IEEE Transactions on Neural Networks **20**(3), 542–542 (2009)
13. Chapelle, O., Weston, J., Schölkopf, B.: Cluster kernels for semi-supervised learning. In: Advances in neural information processing systems. pp. 601–608 (2003)
14. Chen, Y., Ye, J., Li, J.: Aggregated wasserstein distance and state registration for hidden markov models. IEEE Transactions on Pattern Analysis and Machine Intelligence (2019)
15. Courty, N., Flamary, R., Tuia, D., Rakotomamonjy, A.: Optimal transport for domain adaptation. IEEE transactions on pattern analysis and machine intelligence **39**(9), 1853–1865 (2017)
16. Cuturi, M.: Sinkhorn distances: Lightspeed computation of optimal transport. In: Advances in neural information processing systems. pp. 2292–2300 (2013)
17. Cuturi, M., Doucet, A.: Fast computation of wasserstein barycenters. In: International Conference on Machine Learning. pp. 685–693 (2014)
18. Damodaran, B.B., Kellenberger, B., Flamary, R., Tuia, D., Courty, N.: Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation. In: European Conference on Computer Vision. pp. 467–483. Springer (2018)
19. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)

20. Dong-DongChen, W., WeiGao, Z.H.: Tri-net for semi-supervised deep learning. IJCAI (2018)
21. Frogner, C., Zhang, C., Mobahi, H., Araya, M., Poggio, T.A.: Learning with a wasserstein loss. In: Advances in Neural Information Processing Systems. pp. 2053–2061 (2015)
22. Genevay, A., Chizat, L., Bach, F., Cuturi, M., Peyré, G.: Sample complexity of sinkhorn divergences. In: The 22nd International Conference on Artificial Intelligence and Statistics. pp. 1574–1583 (2019)
23. Ho, N., Nguyen, X.L., Yurochkin, M., Bui, H.H., Huynh, V., Phung, D.: Multilevel clustering via wasserstein means. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. pp. 1501–1509. JMLR. org (2017)
24. Iscen, A., Tolias, G., Avrithis, Y., Chum, O.: Label propagation for deep semi-supervised learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5070–5079 (2019)
25. Jia, Y., Kwong, S., Hou, J.: Semi-supervised spectral clustering with structured sparsity regularization. IEEE Signal Processing Letters **25**(3), 403–407 (2018)
26. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
27. Kolouri, S., Park, S.R., Thorpe, M., Slepcev, D., Rohde, G.K.: Optimal mass transport: Signal processing and machine-learning applications. IEEE signal processing magazine **34**(4), 43–59 (2017)
28. Kolouri, S., Zou, Y., Rohde, G.K.: Sliced wasserstein kernels for probability distributions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5258–5267 (2016)
29. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images (2009)
30. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)
31. Laine, S., Aila, T.: Temporal ensembling for semi-supervised learning. arXiv preprint arXiv:1610.02242 (2016)
32. Lee, C.Y., Batra, T., Baig, M.H., Ulbricht, D.: Sliced wasserstein discrepancy for unsupervised domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 10285–10295 (2019)
33. Lee, D.H.: Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In: Workshop on Challenges in Representation Learning, ICML. vol. 3, p. 2 (2013)
34. Lee, J., Dabagia, M., Dyer, E., Rozell, C.: Hierarchical optimal transport for multimodal distribution alignment. In: Advances in Neural Information Processing Systems. pp. 13453–13463 (2019)
35. Liu, X., Van De Weijer, J., Bagdanov, A.D.: Exploiting unlabeled data in cnns by self-supervised learning to rank. IEEE transactions on pattern analysis and machine intelligence (2019)
36. Luo, Y., Zhu, J., Li, M., Ren, Y., Zhang, B.: Smooth neighbors on teacher graphs for semi-supervised learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8896–8905 (2018)
37. Mi, L., Zhang, W., Gu, X., Wang, Y.: Variational wasserstein clustering. arXiv preprint arXiv:1806.09045 (2018)
38. Miyato, T., Maeda, S.i., Ishii, S., Koyama, M.: Virtual adversarial training: a regularization method for supervised and semi-supervised learning. IEEE transactions on pattern analysis and machine intelligence (2018)

39. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning. In: NIPS workshop on deep learning and unsupervised feature learning. vol. 2011, p. 5 (2011)
40. Nguyen, X., et al.: Borrowing strengh in hierarchical bayes: Posterior concentration of the dirichlet base measure. Bernoulli **22**(3), 1535–1571 (2016)
41. Oliver, A., Odena, A., Raffel, C.A., Cubuk, E.D., Goodfellow, I.: Realistic evaluation of deep semi-supervised learning algorithms. In: Advances in Neural Information Processing Systems. pp. 3235–3246 (2018)
42. Pereyra, G., Tucker, G., Chorowski, J., Kaiser, Ł., Hinton, G.: Regularizing neural networks by penalizing confident output distributions. arXiv preprint arXiv:1701.06548 (2017)
43. Pollard, D.: Quantization and the method of k-means. IEEE Transactions on Information theory **28**(2), 199–205 (1982)
44. Rasmus, A., Berglund, M., Honkala, M., Valpola, H., Raiko, T.: Semi-supervised learning with ladder networks. In: Advances in Neural Information Processing Systems. pp. 3546–3554 (2015)
45. Sajjadi, M., Javanmardi, M., Tasdizen, T.: Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In: Advances in Neural Information Processing Systems. pp. 1163–1171 (2016)
46. Santambrogio, F.: Optimal transport for applied mathematicians. Birkäuser, NY **55**, 58–63 (2015)
47. Schmitzer, B., Schnörr, C.: A hierarchical approach to optimal transport. In: International Conference on Scale Space and Variational Methods in Computer Vision. pp. 452–464. Springer (2013)
48. Shen, J., Qu, Y., Zhang, W., Yu, Y.: Wasserstein distance guided representation learning for domain adaptation. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
49. Shi, W., Gong, Y., Ding, C., MaXiaoyu Tao, Z., Zheng, N.: Transductive semi-supervised deep learning using min-max features. In: The European Conference on Computer Vision (ECCV) (September 2018)
50. Solomon, J., De Goes, F., Peyré, G., Cuturi, M., Butscher, A., Nguyen, A., Du, T., Guibas, L.: Convolutional wasserstein distances: Efficient optimal transportation on geometric domains. ACM Transactions on Graphics (TOG) **34**(4), 66 (2015)
51. Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., Liu, C.: A survey on deep transfer learning. In: International Conference on Artificial Neural Networks. pp. 270–279. Springer (2018)
52. Tarvainen, A., Valpola, H.: Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In: Advances in neural information processing systems. pp. 1195–1204 (2017)
53. Villani, C.: Optimal transport: old and new, vol. 338. Springer Science & Business Media (2008)
54. Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al.: Matching networks for one shot learning. In: Advances in neural information processing systems. pp. 3630–3638 (2016)
55. Vural, E., Guillemot, C.: A study of the classification of low-dimensional data with supervised manifold learning. Journal of Machine Learning Research **18**, 157–1 (2017)
56. Wen, Y., Zhang, K., Li, Z., Qiao, Y.: A discriminative feature learning approach for deep face recognition. In: European conference on computer vision. pp. 499–515. Springer (2016)

57. Yan, Y., Li, W., Wu, H., Min, H., Tan, M., Wu, Q.: Semi-supervised optimal transport for heterogeneous domain adaptation. In: IJCAI. pp. 2969–2975 (2018)
58. Ye, J., Wu, P., Wang, J.Z., Li, J.: Fast discrete distribution clustering using wasserstein barycenter with sparse support. IEEE Transactions on Signal Processing **65**(9), 2317–2332 (2017)
59. Yu, B., Wu, J., Ma, J., Zhu, Z.: Tangent-normal adversarial regularization for semi-supervised learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 10676–10684 (2019)
60. Yurochkin, M., Claici, S., Chien, E., Mirzazadeh, F., Solomon, J.M.: Hierarchical optimal transport for document representation. In: Advances in Neural Information Processing Systems. pp. 1599–1609 (2019)
61. Zhou, D., Bousquet, O., Lal, T.N., Weston, J., Schölkopf, B.: Learning with local and global consistency. In: Advances in neural information processing systems. pp. 321–328 (2004)