2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) | 978-1-6654-4509-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/CVPR46437.2021.01209

Self-Supervised Wasserstein Pseudo-Labeling for Semi-Supervised Image Classification

Fariborz Taherkhani, Ali Dabouei, Sobhan Soleymani, Jeremy Dawson, Nasser M. Nasrabadi
West Virginia University
{ft0009, ad0046, ssoleyma} @ mix.wvu.edu, {jeremy.dawson, nasser.nasrabadi} @ mail.wvu.edu

Abstract

The goal is to use Wasserstein metric to provide pseudo labels for the unlabeled images to train a Convolutional Neural Networks (CNN) in a Semi-Supervised Learning (SSL) manner for the classification task. The basic premise in our method is that the discrepancy between two discrete empirical measures (e.g., clusters) which come from the same or similar distribution is expected to be less than the case where these measures come from completely two different distributions. In our proposed method, we first pre-train our CNN using a self-supervised learning method to make a cluster assumption on the unlabeled images. Next, inspired by the Wasserstein metric which considers the geometry of the metric space to provide a natural notion of similarity between discrete empirical measures, we leverage it to cluster the unlabeled images and then match the clusters to their similar class of labeled images to provide a pseudo label for the data within each cluster. We have evaluated and compared our method with state-of-the-art SSL methods on the standard datasets to demonstrate its effectiveness.

1. Introduction

CNN models have enabled breakthroughs in computer vision and machine learning. However, training a CNN model relies on a large-scale annotated datasets which are usually tedious and labor intensive to collect [38]. Considering the vast amounts of unlabeled data available on the web, the idea to use the unlabeled data without human effort to annotate them has become very appealing [77, 11]. In this work, we tackle the challenge of deep SSL, the task of which is to use the unlabeled data in conjunction with the labeled data to train a better CNN classifier. Conventionally, we are given a dataset $\mathcal{D} = \mathcal{D}_l \cup \mathcal{D}_u$ where the data in $\mathcal{D}_l$ are annotated by labels while the data in $\mathcal{D}_u$ are not. The goal is to train a CNN classifier on the known categories in $\mathcal{D}_l$ using the data in $\mathcal{D}$. The test data involves only the classes that are present in $\mathcal{D}_l$. The main challenge in SSL is to efficiently leverage the unlabeled $\mathcal{D}_u$ to help learning on $\mathcal{D}_l$. To make use of unlabeled data in the general setting of SSL challenge, there are two fundamental assumptions that must be taken into the consideration [11]: 1) We assume that labeled and unlabeled data come from the same or similar underlying distribution and there is no class distribution mismatch between the labeled and unlabeled sets. 2) We presume that the underlying distribution of data has some structure. SSL algorithms considers at least one of these structural assumptions: consistency, manifold and cluster.

In consistency assumption [5, 8, 9, 60, 66], data samples in a small neighbourhood have the same class label. In cluster assumption [53, 12, 75, 62], data tends to construct discrete clusters in some geometric sense, and data within the same cluster are more probably to have the same class label. In manifold assumption [49, 59, 70], data lie in the neighbourhood of a low-dimensional and well-defined manifold which can be classified by meaningful distances on the manifold. For all of these assumptions, it is important to consider the geometry of the data when designing an SSL method. For example, popular mean teacher [63] and π-model [39] leverage different data augmentations approaches, each of which uses a different strategy to explore the local geometry of the labeled data for generating new data.

Recently, the theory of Optimal Transport (OT) [57, 64] is used as a tool in machine learning algorithms to consider the geometry of the data. For example, the Wasserstein distance in OT uses the geometry of the metric space to provide a meaningful distance between two distributions even if the supports of these distributions do not overlap. This property of the Wasserstein distance has made it useful and practical for many computer vision and machine learning applications such as clustering [18, 37, 31, 46], generative models [4, 27], loss function [22], semi-supervised learning [61, 23, 69, 43, 62], and domain adaptation [16, 36, 58, 67, 19, 40].

In this work, we propose a new SSL method based on the Wasserstein metric which follows the general assumptions in SSL. Inspired by the effectiveness of Self-Supervised learning in many tasks including SSL [72, 35, 32], we first

978-1-6654-4509-2/21/$31.00 ©2021 IEEE
DOI 10.1109/CVPR46437.2021.01209

pre-train our CNN using a self-supervised learning method, MoCo v2 [28, 13, 14]. This process potentially enforces a clustered structure in the feature space for the unlabeled data which motivates us to perform a clustering on the feature of unlabeled data and then infer a pseudo-label for them.

Specifically, using the self-supervised pre-training on the CNN, we make a cluster assumption about the unlabeled data in which clusters are identified by the Wasserstein barycenter of the unlabeled data. Then, we leverage the Wasserstein metric to match the clusters of unlabeled data to their most similar classes of labeled data to provide pseudo-labels for the unlabeled data. Here, the Wasserstein distance is a measure of similarity between two sets of data points where one of them contains labeled data while the other one consists of unlabeled data. This matching is based on the assumption that the labeled and unlabeled data within the same class have the same or similar distribution. Therefore, we would expect that the similarity between two sets of data which come from the same or similar distribution is more than the case where these sets of data come from completely two different distributions. Finally, depending on the matching, we infer a pseudo label for the unlabeled data within each cluster, which are used along with the initially labeled data to train our CNN classifier.

2. Related Work

2.1. Semi Supervised Learning for Deep Models

There are many SSL algorithms in the literature [78, 11, 52]. However, we briefly review the methods based on the pseudo labeling and consistency regularization which have been incorporated with deep learning models.

Pseudo-Labeling was initially proposed in [41]. In SSL models based on the pseudo-labeling, the model usually relies on its own prior belief about the label of unlabeled data to obtain supplementary information over the course of training [41, 55, 21, 42, 34]. The main drawback of these methods is susceptibility to confirmation bias such that the model is confident about its incorrect prediction, and then overfits to incorrect pseudo-labels during the training [3]. Therefore, in these models, the incorrect pseudo-labels not only can not provide useful information during the training but also error of the model's prediction is accumulated in the model and results in overfitting. This downside even gets worse in cases where the discrepancy between the domain of the unlabeled data is significant from that of labeled data.

Consistency-based SSL models perform based on the assumption that the model should be generally consistent with its predictions between a given data and its meaningfully-distorted versions [7]. This simple criterion on the models output has provided promising results in the SSL literature such as stochastic perturbations models [56], π-model [39], mean teacher [63], and virtual adversarial training (VAT) [47], Mixmatch [9], Remixmatch [8], and Fixmatch [60]. The primary idea in stochastic perturbations and π-model was initially proposed in [6] and is known as pseudo-ensembles. The pseudo-ensemble regularization techniques usually perform in such a way that under realistic perturbations of input x: ($x \sim x'$), the prediction of the model $g(x, \theta)$ should not vary drastically. This objective is achieved by considering a weighted loss term such as $d(g(x, \theta), g(x', \theta))$ during the training of model, where $d(.,.)$ denotes MSE or KL divergence which calculates a distance or divergence between outputs of the prediction function. The main problem in pseudo-ensemble approaches, including π-model is that they highly depend on a likely unstable prediction, which can instantly deviate significantly over the course of training.

To solve this issue, two approaches including temporal ensembling [39] and mean teacher [63], were introduced to achieve a more stable target output $g'(x, \theta)$. In temporal ensembling, the model uses an exponentially accumulated average of outputs, $g(x, \theta)$, to produce a smooth and consistent target output while in mean teacher, the model uses a prediction function parametrized by an accumulated average of the model parameters θ during the training. Contrary to the stochastic perturbation methods mentioned earlier, VAT initially estimates a small perturbation r to add it to x which drastically changes the model prediction, $g(x, \theta)$. Then, a consistency regularization term, $d(g(x, \theta), g(x + r, \theta))$ is considered as a loss term during the training.

Following the advance in consistency regularization, and pseudo-labeling for SSL, MixMatch integrates data augmentation, consistency regularization [56], entropy minimization [26], and mixup [73]. ReMixMatch enhanced on MixMatch by including augmentation anchors and distribution alignment. Augmentation anchors are performs similar to pseudo-labeling. FixMatch which is the sate-of-the art and the most recent approach in this line of research combines consistency regularization, and pseudo-labeling with a threshold of confidence on the output of the model.

2.2. Self-Supervised Learning

The idea behind self-supervised learning (Self-SL) is to take large amount of readily and available unlabeled data and use it to understand itself [13, 14, 28, 50, 65]. Generally, the purpose of Self-SL for images is to create image representations that are semantically meaningful via pretext tasks that do not need human-annotations for a large training dataset. Pretext tasks usually guide the model towards learning meaningful representations that are covariant with image transformations such as rotations [25], and jigsaw transformations [50], and affine transformations [51, 74]. Recently, it has been shown that Self-SL approaches can be simply used to leverage all unlabeled data for learning and can be incorporated by SSL models [72, 35, 32]. For example, the

work in [72] demonstrated that integrating simple Self-SL losses such as rotation is useful for a SSL approach.

3. Wasserstein Distance

For any subset $\theta \subset \mathbb{R}^d$, assume that $\mathcal{P}(\theta)$ represents the space of Borel probability measures on θ. The Wasserstein space of order $k \in [1, \infty)$ of probability measures on θ is defined as follows: $\mathcal{P}_k(\theta) = \{\mathbb{F} \in \mathcal{P}(\theta) : \int ||x||^k d\mathbb{F}(x) < \infty\}$, where $||.||$ is the Euclidean distance in $\mathbb{R}^d$. Let $x \sim \mathbb{P} \in \mathcal{P}(\theta)$, $y \sim \mathbb{Q} \in \mathcal{P}(\theta)$ and $\mathcal{J}(\mathbb{P}, \mathbb{Q})$ denote all the joint distributions J for (x, y) on $\theta \times \theta$ that have marginals $\mathbb{P}$ and $\mathbb{Q}$ for x and y, respectively, and also assume that $\delta(x, y)$ is a distance measure between two instances x and y. Then, the Wasserstein distance is defined as follows:

$$W_k(\mathbb{P}, \mathbb{Q}) = \left(\inf_{J \in \mathcal{J}(\mathbb{P}, \mathbb{Q})} \int \delta(x, y)^k dJ(x, y) \right)^{1/k}, \quad (1)$$

where $k \geq 1$. In case $k = 1$, this is also called the Earth Mover distance. The term $J(x, y)$ can be considered as a plan that transports a unit of mass from location x to another location y such that the marginal constraints are satisfied. The minimizer J^* in Eq. (1) is called the optimal transport plan. In the case where transporting cost of a unit of mass from $x \sim \mathbb{P}$ to $y \sim \mathbb{Q}$ is equal to $\delta(x, y)^k$, then $W_k(\mathbb{P}, \mathbb{Q})$ is the minimum expected transportation cost. The Kantorovich-Rubinstein dual theorem [64] indicates that in the special case where $k = 1$, the Wasserstein distance has a closed form of an integral probability metric as follows:

$$W_1(\mathbb{P}, \mathbb{Q}) = \sup_{||f||_L \leq 1} \mathbb{E}_{x \sim \mathbb{P}}[f(x)] - \mathbb{E}_{x \sim \mathbb{Q}}[f(x)], \quad (2)$$

where the supremum is over all 1-Lipschitz functions $f : \mathcal{X} \to \mathbb{R}$, and Lipschitz semi-norm is defined as follows: $||f||_L = \sup |f(x) - f(y)|/\delta(x, y)$.

4. Wasserstein Barycenter

Wasserstein Barycenter was initially introduced by [1], and provided an efficient role in clustering methods based on OT [18, 37, 31, 46]. Let θ denote a Polish space, and $P(\theta)$ represent the space of probability measures on this space. Moreover, let's assume that we are given $M \geq 1$ probability measures $\mathcal{P}_1, \mathcal{P}_2, ..., \mathcal{P}_M \in P(\theta)$ with finite second moments, then the Wasserstein barycenter of these measures is defined as follows:

$$B(\tilde{\mathcal{P}}) = \inf_{\tilde{\mathcal{P}} \in P(\theta)} \frac{1}{M} \sum_{i=1}^{M} W_2^2(\tilde{\mathcal{P}}, \mathcal{P}_i), \quad (3)$$

it has been demonstrated by [2] that the problem of exploring Wasserstein barycenter on the space of $P(\theta)$ in Eq. (3) boils down to search only on a reduced space $\mathcal{O}_r(\theta)$ where $r = \sum_{i=1}^{M} e_i - M + 1$ and e_i is the number of elements in $\mathcal{P}_i$

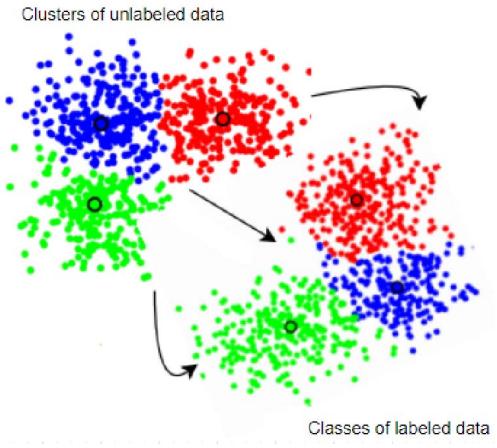


Figure 1. An illustration of mapping clusters to classes.

for all $1 \leq i \leq M$. Moreover, several practical and effective algorithms have been recently proposed in [1, 18, 68] that provide proper local solutions for the Wasserstein barycenter problem over the space of $\mathcal{O}_r(\theta)$. These algorithms such as the one in [18] have been a building block for many interesting clustering algorithms based on OT such as [31].

5. Proposed Method

Here, we describe the outline of our SSL method. Our SSL model contains three steps as follows: in step (1), we initially pre-train our CNN using a self-SL method on the unlabeled data and then fine-tune it using the initially labeled data. This operation potentially encourages the CNN model to construct cluster structure when representing the data. For example, in Self-SL based on contrastive learning paradigm [28, 13, 14], the goal is to learn similarities/dissimilarities such that the model is able to understand that the similar data should be closer to each other while dissimilar data should be far away from each other in terms of their representations. Therefore, pre-training the CNN motivates us to make a cluster assumption on the unlabeled data and then annotate each cluster with a unique pseudo-label.

In step (2), we use Wasserstein distance as a metric of similarity between two discrete probability measures to match each cluster of the unlabeled data to the most similar class of the labeled data for pseudo-labeling (see Fig. 1). This pseudo-labeling is based on the SSL assumption in which the labeled and unlabeled data within the same class should come from the same or similar distribution. Thus, we would expect that the similarity between two clouds of data which come from the same or similar distribution is more than the case where these clouds come from completely two different distributions.

Finally, in step (3), we use the unlabeled data annotated with the pseudo labels obtained from step (2) in conjunction

with the initially labeled data to train our CNN classifier.

5.1. Self-Supervised Learning and Clustering via Wasserstein Barycenter

As discussed earlier, in step (1), we initially pre-train our CNN model using a Self-SL paradigm on the unlabeled data to make a cluster assumption for them. Here, we use MoCo v2 Self-SL [14] as it is a strong and efficient Self-SL method. Specifically, we use SimCLR [13] style data augmentation for the unlabeled images in the contrastive loss, and follow the implementation details in MoCo v2 where we use a two-layers MLP on the top of the last feature layer to map image features to a 128 dimensions, and then use a momentum updated model to calculate the key features in the memory bank.

After pre-training, we use the Wasserstein metric to perform a clustering on the unlabeled features extracted from the network. Following the previous clustering method based on OT [18, 37, 31, 46], here we relate the clustering algorithm to the problem of exploring Wasserstein barycenter of the unlabeled data to find the clusters underlying them. The K-means objective is an optimization problem that has come up in the quantization problem [54].

Given n unlabeled data $\{x_1, ..., x_n\} \in \mathbb{R}^d$, suppose that these data are grouped into k clusters where $k \geq 1$. The K-means algorithm aims to find a set C which contains k elements $\{c_1, ..., c_k\}$ that minimizes the following objective:

$$F(C) = \inf_{C} \frac{1}{n} \sum_{i=1}^{n} D^2(x_i, C), \tag{4}$$

let $\mathcal{P}_n = \frac{1}{n}\sum_{i=1}^{n} \delta_{x_i}$ be a probability measure where δ_{x_i} is the Dirac function on x_i. Then, problem (4) is equal to exploring a probability measure $\mathcal{Q}$ with k finite atoms that minimizes the following objective:

$$B(\mathcal{Q}) = \inf_{\mathcal{Q} \in \mathcal{O}_k(\theta)} \sum_{i=1}^{n} W_2^2(\mathcal{Q}, \mathcal{P}_n), \tag{5}$$

this optimization problem can also be thought as a Wasserstein barycenter problem when $M = 1$ in Eq. (3). From this prospective, as introduced by [18], the algorithm for exploring the Wasserstein barycenter is an alternative for the well-known Loyd's algorithm to obtain local minimum for the K-means. In this work, we use [18] to find the Wasserstein barycenter of the unlabeled data for clustering.

5.2. Matching Clusters to Classes via WGAN

After clustering the unlabeled data, in step (2), we follow the cluster assumption in SSL where data within the same cluster more likely should have the same class label. Moreover, in the general setting of SSL, data within the same class in both labeled and unlabeled sets have the same or similar distribution. Therefore, by considering the Wasserstein distance as a metric of similarity between two discrete probability measures, label of each cluster can be predicted based on the closest Wasserstein distance that the cluster has with a class of labeled data in the labeled set. This is because we would expect that the similarity between two sets of data coming from the same or similar distribution is more than the case where they come from completely two different distributions. Since we usually deal with large scale datasets, and CNN model is usually trained by stochastic gradient descent, we follow the standard training procedure, and use an approach based on gradient descent [4, 58, 24] to compute the Wasserstein distance.

Suppose that $\mathcal{P}_i = \frac{1}{n_i}\sum_{j=1}^{n_i} \delta_{x_{ij}}$ denotes a labeled discrete measure which is constructed by labeled data x_{ij} belonging to the i-th class; and $\mathcal{Q}_i = \frac{1}{n'_i}\sum_{j=1}^{n'_i} \delta_{x'_{ij}}$ denotes an unlabeled discrete measure which is constructed by unlabeled data x'_{ij} belonging to the i-th cluster. In step (2) of our algorithm, we aim to match each of $\mathcal{Q}_1, ..., \mathcal{Q}_k$ to one of the labeled measures $\mathcal{P}_1, ..., \mathcal{P}_c$, so that we can infer a label for each cluster. Therefore, we use the empirical Wasserstein distance as a measure of similarity between each pair ($\mathcal{Q}_i$, $\mathcal{P}_j$) to match the pairs. For example, if the labeled measure $\mathcal{P}_m$ is the closest measure to the unlabeled measure $\mathcal{Q}_i$, we annotate the data within the i-th cluster with label m.

In our SSL method, we use the CNN pre-trained via Self-SL to extract the feature for a given sample. Given an image $x \in \mathbb{R}^{m \times n}$, the CNN as a function $f_n : \mathbb{R}^{m \times n} \to \mathbb{R}^d$ with parameters θ_n maps sample x to a d-dimensional representation. Inspired by the Wasserstein Generative Adversarial Network (WGAN) [4], we use a critic layer to compute the Wasserstein distance between each pair ($\mathcal{Q}_i$, $\mathcal{P}_j$). Given a feature $z = f_n(x)$ obtained by the CNN, the critic layer in our model learns a function $f_c : \mathbb{R}^d \to \mathbb{R}$ with parameters θ_c that maps a feature to a real number. Therefore, the Wasserstein distance between two discrete measures $\mathcal{P}_i$ and $\mathcal{Q}_j$, where $z = f_n(x)$, $z' = f_n(x')$, $x \in \mathcal{P}_i$ and $x' \in \mathcal{Q}_j$ can be calculated by using Eq. (2) as follows:

$$\begin{aligned} W_1(\mathcal{P}_i, \mathcal{Q}_j) &= \sup_{||f_c||_L \leq 1} \mathbb{E}_{\mathcal{P}_i}[f_c(z)] - \mathbb{E}_{\mathcal{Q}_j}[f_c(z')] \\ &= \sup_{||f_c||_L \leq 1} \mathbb{E}_{\mathcal{P}_i}[f_c(f_n(x))] - \mathbb{E}_{\mathcal{Q}_j}[f_c(f_n(x'))]. \end{aligned} \tag{6}$$

By considering the parameterized class of critic functions f_c are all 1-Lipschitz, we can then calculate the empirical Wasserstein distance by maximizing the critic loss $\mathcal{L}_w$ with respect to parameters θ_c as follows:

$$\mathcal{L}_w(\mathcal{P}_i, \mathcal{Q}_j) = \frac{1}{|\mathcal{P}_i|}\sum_{x \in \mathcal{P}_i} f_c(f_n(x)) - \frac{1}{|\mathcal{Q}_j|}\sum_{x' \in \mathcal{Q}_j} f_c(f_n(x')). \tag{7}$$

Now, we need to force the Lipschitz constraint. In WGAN [4], it is suggested to clip the weights of critic layer in a compact interval $[-c, c]$ after each gradient update. However,

weight clipping causes some issues including capacity underuse, and exploding problems or gradient vanishing [27]. Therefore, we use the technique used [27] to force a gradient penalty $\mathcal{L}_{grad}$ for critic parameters θ_c as follows:

$$\mathcal{L}_{grad}(\hat{z}) = (||\nabla_{\hat{z}} f_c(\hat{z})||_2 - 1)^2, \tag{8}$$

where the features $\hat{z}$ on which to penalize the gradients are the features of the labeled and unlabeled data, and also the random points along the line between labeled and unlabeled pairs. Therefore, we can approximate the Wasserstein distance by optimizing the following objective:

$$W_1(\mathcal{P}_i, \mathcal{Q}_j) = \max_{\theta_c}(\mathcal{L}_w - \alpha\mathcal{L}_{grad}), \tag{9}$$

where α is a coefficient that balances between $\mathcal{L}_w$ and $\mathcal{L}_{grad}$.

5.3. Total Loss for Training the CNN

In step (3), we remove two-layers MLP from top of the last feature layer which we used for Self-SL, and then place a softmax layer for the classification task. In this step, we aim to use the unlabeled data annotated by the pseudo labels in conjunction with the supervision signals of the initially labeled data to train our CNN classifier. Therefore, we use the regular cross entropy loss to train the parameters of our CNN as follows: Let $\mathcal{X}_l$ be all of the labeled training data annotated by true labels $\mathcal{Y}$, and $\mathcal{X}_u$ be the unlabeled training data annotated by pseudo labels $\mathcal{Y}'$, then the total loss function $\mathcal{L}(.)$, for training the CNN in SSL fashion is:

$$\mathcal{L}(\theta_n, \mathcal{X}_l, \mathcal{X}_u, \mathcal{Y}, \mathcal{Y}') = \mathcal{L}_c(\theta_n, \mathcal{X}_l, \mathcal{Y}) + \lambda\mathcal{L}_c(\theta_n, \mathcal{X}_u, \mathcal{Y}'), \tag{10}$$

where $\mathcal{L}_c(.)$ denotes cross entropy loss function, and λ is a hyperparameter that balances between two losses obtained from the labeled and unlabeled data. Our algorithm to train a CNN in the SSL fashion is described in Algorithm 1:

Algorithm 1 Self-Supervised Wasserstein Pseudo-Labeling

input: $\mathcal{X}_l, \mathcal{X}_u, \alpha, \lambda, \beta_1, \beta_2, b, k, m$

initialize: critic layer θ_c with $\mathcal{N}(0, 0.001)$.
pretrain θ_n using MoCo v2 Self-SL.
repeat
  $Z_l = \{z_l\}_{l=1}^m$, $Z_u = \{z'_u\}_{u=1}^m$: where $z_i = f_n(x_i)$.
  $\{\mathcal{Q}_1, ..., \mathcal{Q}_k\} \leftarrow$ cluster Z_u to k groups.
  $\{\mathcal{P}_1, ..., \mathcal{P}_c\} \leftarrow$ cluster Z_l to c classes.
  for each $\mathcal{Q}_i$ and $\mathcal{P}_j$ **do**
    for $i = 1, ..., s$ **do**
      choose a batch: $\{x_i\}_{i=1}^b \subset \mathcal{P}_j$, $\{x'_i\}_{i=1}^b \subset \mathcal{Q}_i$,
      $z'_i \leftarrow f_n(x'_i)$, $z_i \leftarrow f_n(x_i)$,
      $\hat{z} \leftarrow \{z'_i, z_i, \tilde{z}\}$: take sample $\tilde{z}$ randomly on lines between z'_i and z_i pairs,
      $\theta_c \leftarrow \theta_c + \beta_1 \nabla_{\theta_c}[\mathcal{L}_w(z', z) + \alpha\mathcal{L}_{grad}(\hat{z})]$,
    end for
    $S(i, j) \leftarrow \mathcal{L}_w(\mathcal{P}_i, \mathcal{Q}_j)$, by Eq. (7)
  end for
  $\{y'_u\}_{u=1}^m \leftarrow$ pseudo label data within each cluster $\mathcal{Q}_j$ with the most similar class (i.e., $\operatorname{argmin} S(:, j)$),
  repeat
    choose a batch:$\{x_i\}_{i=1}^b \subset \mathcal{X}_u \cup \mathcal{X}_l$,
    $\theta_n \leftarrow \theta_n - \beta_2 \nabla_{\theta_n}[\mathcal{L}(\theta_n, x, x', y, y')]$, by Eq. (10)
  until for an epoch
until θ_n converge

6. Experiments

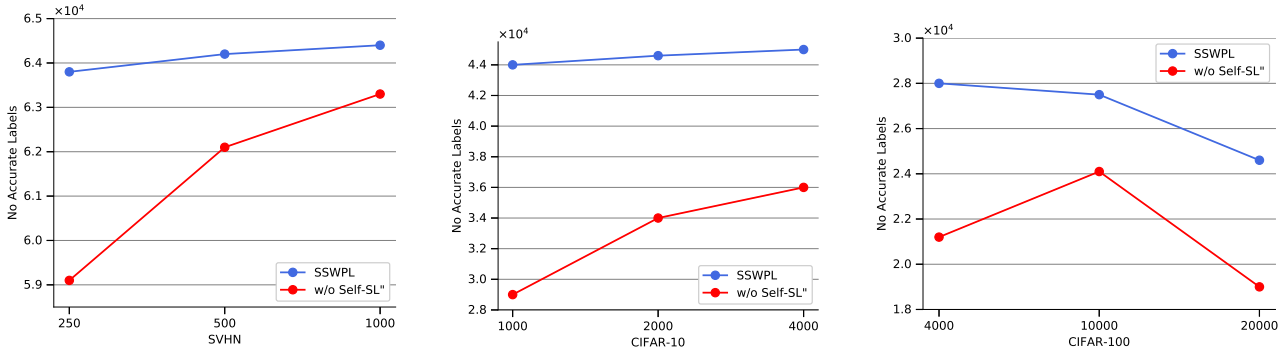
We carry out empirical analysis to show the effectiveness and benefit of our SSL algorithm over other state-of-the-art methods [55, 41, 66, 63, 9, 8, 60]. Here, we perform following studies: 1) We report results for supervised-baseline where the CNN is only trained by initially labeled data, this is because the goal of SSL is to greatly improve the supervised-baseline. 2) We change number of the labeled and unlabeled data and report the results as an efficient SSL method should still perform well even by using a small number of labeled data and extra amount of unlabeled data. 3) We replace our OT-base clustering method with the popular k-means and report the results to demonstrate the importance of the Wasserstein metric in our SSL algorithm. 4) We conduct an analysis on the clustering resolution (i.e., k in Alg 1) to see its importance in our model.

Following the compared methods, we have been consistent in CNN network and used the 'WRN-28-2' [71], including leaky ReLU nonlinearities [45] and batch normalization [33]. We performed our experiments on the widely used CIFAR-10/100 [38], SVHN [48], and ImageNet [20] datasets. We note that in all of our experiments, we consider the general SSL setting where the labeled and unlabeled data coming the same or similar distribution, and a given unlabeled data belongs to one of the classes in the labeled set and there is no class distribution mismatch. Furthermore, for each of aforementioned datasets, we split the training set into two different sets of labeled and unlabeled data. We make sure that all classes are balanced such that each class should have the same number of labeled data.

For training, we set hyperparameter λ to 0.7 in all of our experiments. We use the regular SGD optimizer with momentum 0.9, and weight decay 10^{-4}. We set the learning rate β_2 in Alg 1 to 3×10^{-3} in all of our experiment. The batch size in the experiments (b in Alg 1) is set to 128. We note that our batch size for training the CNN (b) is different from the batch size that we map the unlabeled data to the labeled data (m in Alg 1). The batch size for mapping the unlabeled data to the labeled data is the size of initially labeled data ($|\mathcal{X}_l|$). In other words, each time, we select $|\mathcal{X}_l|$ unlabeled data to cluster them. Then, we use WGAN to map these clusters to the groups of data formed by $\mathcal{X}_l$.

Datasets	CIFAR-10		CIFAR-100		SVHN	
Labels	250	4000	2500	10000	250	1000
Supervised	56.85 ± 1.34	19.74 ± 0.23	59.47 ± 0.56	40.97 ± 0.22	24.95 ± 0.49	12.91 ± 0.26
π model [55]	54.26 ± 3.97	14.01 ± 0.38	57.25 ± 0.48	37.88 ± 0.11	18.96 ± 1.92	7.54 ± 0.36
Pseudo-Labeling [41]	49.78 ± 0.43	16.09 ± 0.28	57.38 ± 0.46	36.21 ± 0.19	20.21 ± 1.09	9.94 ± 0.61
UDA [66]	8.82 ± 1.08	4.88 ± 0.18	33.13 ± 0.22	24.50 ± 0.25	5.69 ± 2.76	2.46 ± 0.245
MT [63]	32.32 ± 2.30	9.19 ± 0.19	53.91 ± 0.57	35.83 ± 0.24	3.57 ± 0.11	3.42 ± 0.07
MixMatch [9]	11.05 ± 0.86	6.42 ± 0.10	39.94 ± 0.37	28.31 ± 0.33	3.98 ± 0.23	3.50 ± 0.28
ReMixMatch [8]	5.44 ± 0.05	4.72 ± 0.13	27.43 ± 0.31	23.03 ± 0.56	2.92 ± 0.48	2.65 ± 0.08
FixMatch [60]	5.07 ± 0.33	4.31 ± 0.15	28.64 ± 0.24	23.18 ± 0.11	$\mathbf{2.64 \pm 0.64}$	2.36 ± 0.19
SSWPL (k-means)	9.62 ± 0.47	7.74 ± 0.73	30.19 ± 0.35	25.75 ± 0.60	6.16 ± 0.18	4.59 ± 0.34
SSWPL	$\mathbf{4.11 \pm 0.15}$	$\mathbf{3.18 \pm 0.09}$	$\mathbf{26.52 \pm 0.45}$	$\mathbf{20.88 \pm 0.85}$	2.71 ± 0.25	$\mathbf{2.27 \pm 0.07}$

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



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

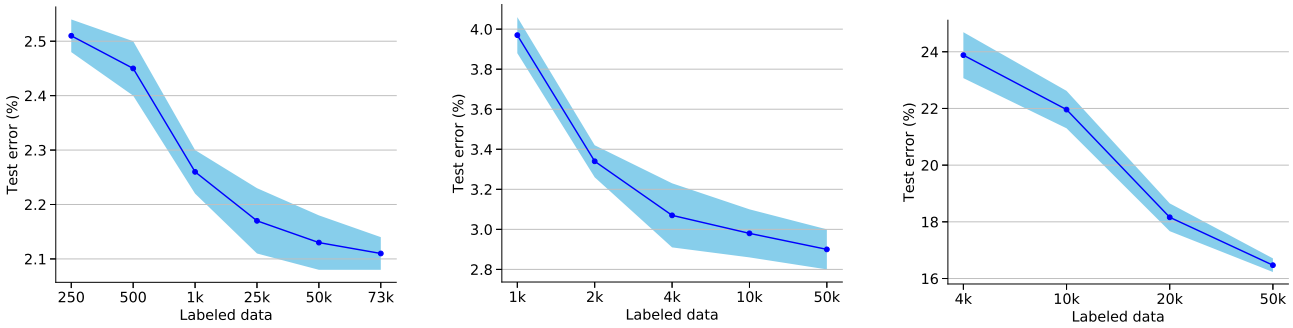
Figure 2. Number of accurate predicted labels by SSWPL in case (1) and (2).

The parameters of the network (θ_n in Alg 1) are initialized by pre-training via MoCo v2 Self-SL [14]. Here, we follow the implementation details from MoCo-v2 but we use a memory bank of size 16384. We initialized the parameters of the critic layer (θ_c in Alg 1) by sampling randomly from $\mathcal{N}(0, 0.001)$. The critic layer parameters usually requires around 10 epochs (s in Alg 1) to converge in our experiments but we set it to 20 epochs for a sufficient optimization guarantee for the parameters of the critic layer. For training the critic layer, the learning rate is also set to $\beta_1 = 3 \times 10^{-3}$. Note that during the training of the critic layer, we penalize the gradients not only at CNN outputs for the unlabeled and labeled data points but also at random points along the line between pairs of labeled and unlabeled data points. The coefficient α is set to 10 as is suggested in [27].

In our experiments, we use the regular data augmentation and standard data normalization techniques. Specifically, for SVHN, we converted and normalized pixel intensity values of the images to floating point values in the range of [-1, 1]. For the data augmentation, we only applied random translation by up to 2 pixels. For CIFAR-10/100, we used global contrast normalization. The data augmentation on CIFAR-10/100 are random translation by up to 2 pixels, random horizontal flipping, and Gaussian input noise with standard deviation 0.15.

6.1. Comparison

The goal in SSL is essentially to obtain a better performance when we use the unlabeled data compared to the case where we use the labeled data alone. Thus, we report the error rate of our 'WRN-28-2' for cases where we only use a limited amount of labeled data (i.e., Supervised in Table. 1), and the case where we leverage the unlabeled data using our SSL method called Self-Supervised Wasserstein Pseudo Labeling (SSWPL) in Table. 1. Furthermore, we report the performance of the other SSL methods including π model [55], Pseudo-Labeling [41], UDA [66], MT [63], MixMatch [9], ReMixMatch [8], and FixMatch [60] in Table. 1. For comparison, we chose 250, and 4000 labeled images for CIFAR-10, 2500, and 10000 labeled images for CIFAR-100, 250, and 1000 labeled images for SVHN. Here, the remaining images of the training set are used as the unlabeled images to train the network . We ran our SSL methods over 5 times with different random splits of labeled and unlabeled sets for each dataset, and we reported the mean and standard deviation of the test error rate in Table. 1. The results on CIFAR-10/100 and SVHN datasets in Table. 1 demonstrate the potential of SSWPL for using the unlabeled data in comparison to other



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

Figure 3. Test error rate of SSWPL by changing the number of labeled data

state-of-the-art SSL algorithms.

6.2. Self-SL Contribution on Clustering

As discussed in step (1), we pre-train our CNN model using Self-SL method and then form the clusters on the unlabeled data. Here, we evaluate the importance of the Self-SL on the clustering performance in our model. This is because one may assume that pre-training of the CNN on the initially labeled data can also enforce a clustered structure in the feature space for the unlabeled data, so it is important to know the benefit of using Self-SL on the clustering performance which plays an essential role in our model. Therefore, we conducted experiments to compare two different cases where in case (1), we fine-tune the network using initially labeled data without considering the Self-SL while in case (2) we consider the Self-SL for clustering. To compare these two cases and indicate the positive influence of the Self-SL on clustering, we changed the number of initially labeled data in the training set and reported the number of accurately predicted pseudo labels using our SSL method in case (1) and (2) on the remaining unlabeled training data. The significant gap between case (1) and case (2) which are respectively indicated by SSWPL and w/o Self-SL in Fig. 2 show that for CIFAR-10/100 and SVHN datasets, the labels predicted by our SSL method on the unlabeled training data are more accurate in case (1) than case (2), which means that the entire CNN network can benefit from Self-SL.

6.3. Analysis in Limited Label Regime

Here, we investigate that how changing the amount of initially labeled data increase the accuracy of our SSL algorithm in the very limited label scenario, and also at which point our SSL algorithm can recover the performance of training when using all of the labeled data in the dataset. To conduct this evaluation, we moderately increase the number of labeled samples during the training and report the performance of our SSL algorithm on the testing set. In this study, we ran our SSL algorithm over 5 times with different random splits of labeled and unlabeled sets for each dataset, and reported the mean and standard deviation of the error rate in Fig. 3. The results indicate that the performance of our SSL method on CIFAR-10/100 and SVHN inclines to converge as the number of initially labeled data increases.

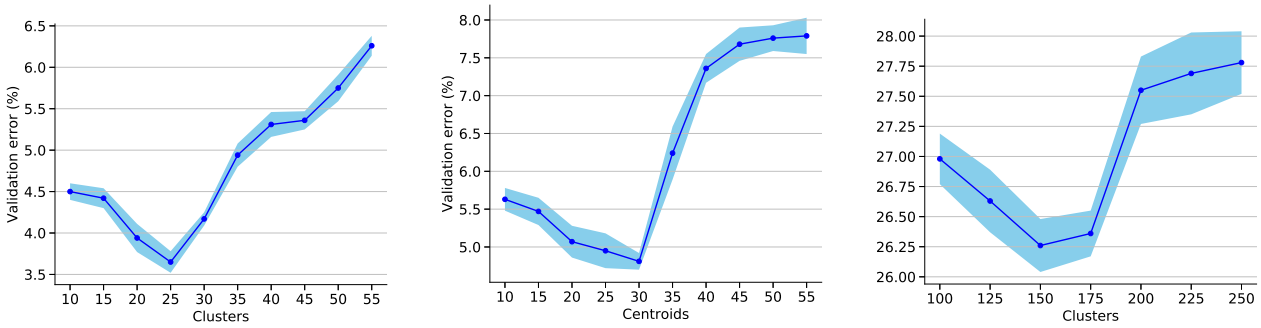
6.4. Varying Number of the Clusters

We evaluate the role of the clustering resolution on the error rate of SSWPL. In this study, we use 500, 1000, and 4000 labeled images from the training sets of SVHN, CIFAR-10, and CIFAR-100 datasets, respectively. We change the number of the clusters in our model, and evaluate error of the model on the validation set. The results on SVHN and CIFAR-10/100 datasets in Fig. 4 demonstrate that as we increase the number of the clusters in our model, the model can benefit from it but performance of the model inclines to degrade as we largely perform over-clustering. The reason can be interpreted by SSL models based on consistency regularization [76, 44, 5]. In other words, if we significantly perform over-clustering, we basically disregard the local geometry or structure of the data when mapping clusters to the label classes using the Wasserstein metric which is not useful in SSL as we neglect the local consistency.

Furthermore, in our other studies, instead of using the Wasserstein metric in the K-means objective for clustering the unlabeled data, we used the generic K-means in SSWPL and reported the test error rate in Table. 1. We call this baseline as SSWPL (K-means). The compared results between SSWPL and SSWPL (K-means) on SVHN, and CIFAR-10/100 datasets in Table. 1 demonstrate advantage of leveraging the Wasserstein-metric in the K-means objective for our SSL model.

6.5. Results on ImageNet

We also conducted an experiment on the large-scale ImageNet dataset to evaluate the performance of our model when using unlabeled unlabeled data in a very limited label regime.



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

Figure 4. Validation error rate of the SSWPL by varying number of clusters

Following the prior work [60, 66], we also used a ResNet-50 architecture and RandAugment [17] data augmentation technique to conduct our experiments. Here, we set the number of the clusters in our method to the number of classes (i.e., 1000). We used 10% of the training set as our initially labeled data and the remaining as the unlabeled data. We ran our model 3 times and reported the mean and standard deviation of top-1 (top-5) error rate. The supervised-baseline top-1 (top-5) error rate using 10% of the training data is $45.64 \pm 0.83\%$ (24.67 ± 0.32) while for our SSL model (i.e., SSWPL), FixMatch[60], and UDA [66] are $26.46 \pm 0.44\%$ $(9.14 \pm 0.26\%)$, $28.54 \pm 0.52\%$ $(10.87 \pm 0.28\%)$, and 31.22% (11.2%), respectively. These results indicate the efficiency and potential of our SSL method compared to other effective SSL approaches for the large-scale datasets.

6.6. Limitation, discussion and Future Work

As mentioned earlier, in this study we consider the general setting of SSL in the literature [78, 11, 52] where there is no class distribution mismatch and the main assumption is that the labeled and unlabeled data coming from the same or similar distribution. Specifically, every given unlabeled data should belong to one of the classes which present in the labeled set. However, the work [52] in Sec. 4.4 showed that using unlabeled data from the mismatched classes essentially has a negative impact on the performance of the studied SSL approaches compared to the case where these approaches do not use any unlabeled data at all. Likewise, our method would also hurt the performance when using the unlabeled data from the mismatched classes. This is because our method will provide a pseudo-label for the unlabeled data whether they belong to the mismatched classes or not. Thus in such a case, our model predicts high confident but incorrect labels for the unlabeled data within the mismatched classes and then use them for training which causes a confirmation bias problem [3]. However, pre-training the network using the Self-SL approach on the unlabeled data as we used in our method potentially can cluster the unlabeled data from mismatched classes as good as the unlabeled data which are not from mismatched classes. Therefore, in such a case, we can perform the clustering approach on the entire data and then disregards the clusters which contain the unlabeled data from mismatched classes during the training. There are many methods in the literature [10, 29, 30, 15] that are proposed to detect out of distribution samples which we can potentially use them to detect out of distribution clusters. We will consider this study as our future work.

7. Conclusion

We proposed a new SSL algorithm that uses the Wasserstein distance and Self-SL technique to provide pseudo labels for the unlabeled data to train a CNN classifier in an SSL fashion. In this work, after pre-training the CNN model using a Self-SL method, we made a cluster assumption about the unlabeled data and then used their Wasserstein barycenter to explore the clusters underlying them. In the next step, we used the Wasserstein GAN to match each of the clusters to the most similar class of labeled data so we can provide a unique label for the data within each cluster. Finally, we used all the unlabeled data annotated by pseudo labels in conjunction with the initially labeled data to train our CNN model. In this study, we conducted empirical analysis to demonstrate the potential and efficiency of our SSL algorithm for leveraging the unlabeled data when labels are limited over the course of training.

References

[1] Martial Agueh and Guillaume Carlier. Barycenters in the wasserstein space. *SIAM Journal on Mathematical Analysis*, 43(2):904–924, 2011.

[2] Ethan Anderes, Steffen Borgwardt, and Jacob Miller. Discrete wasserstein barycenters: optimal transport for discrete data. *Mathematical Methods of Operations Research*, 84(2):389–409, 2016.

[3] Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International*

Joint Conference on Neural Networks (IJCNN), pages 1–8. IEEE, 2020.

[4] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[5] Ben Athiwaratkun, Marc Finzi, Pavel Izmailov, and Andrew Gordon Wilson. There are many consistent explanations of unlabeled data: Why you should average. In *International Conference on Learning Representations*, 2018.

[6] Philip Bachman, Ouais Alsharif, and Doina Precup. Learning with pseudo-ensembles. In *Advances in Neural Information Processing Systems*, pages 3365–3373, 2014.

[7] 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), 2006.

[8] David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *arXiv preprint arXiv:1911.09785*, 2019.

[9] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 5049–5059, 2019.

[10] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 93–104, 2000.

[11] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

[12] Olivier Chapelle, Jason Weston, and Bernhard Schölkopf. Cluster kernels for semi-supervised learning. In *Advances in neural information processing systems*, pages 601–608, 2003.

[13] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

[14] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[15] Yanbei Chen, Xiatian Zhu, Wei Li, and Shaogang Gong. Semi-supervised learning under class distribution mismatch. In *AAAI*, pages 3569–3576, 2020.

[16] Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 39(9):1853–1865, 2017.

[17] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020.

[18] Marco Cuturi and Arnaud Doucet. Fast computation of wasserstein barycenters. In *International Conference on Machine Learning*, pages 685–693, 2014.

[19] Bharath Bhushan Damodaran, Benjamin Kellenberger, Rémi Flamary, Devis Tuia, and Nicolas Courty. Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation. In *European Conference on Computer Vision*, pages 467–483. Springer, 2018.

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

[21] WeiWang Dong-DongChen and Zhi-HuaZhou WeiGao. Trinet for semi-supervised deep learning. IJCAI, 2018.

[22] Charlie Frogner, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya, and Tomaso A Poggio. Learning with a wasserstein loss. In *Advances in Neural Information Processing Systems*, pages 2053–2061, 2015.

[23] Tingran Gao, Shahab Asoodeh, Yi Huang, and James Evans. Wasserstein soft label propagation on hypergraphs: Algorithm and generalization error bounds. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3630–3637, 2019.

[24] Aude Genevay, Marco Cuturi, Gabriel Peyré, and Francis Bach. Stochastic optimization for large-scale optimal transport. In *Advances in neural information processing systems*, pages 3440–3448, 2016.

[25] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

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

[27] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5767–5777, 2017.

[28] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[29] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

[30] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. *arXiv preprint arXiv:1812.04606*, 2018.

[31] Nhat Ho, Xuan Long Nguyen, Mikhail Yurochkin, Hung Hai Bui, Viet Huynh, and Dinh Phung. Multilevel clustering via wasserstein means. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1501–1509. JMLR. org, 2017.

[32] Chao Huang, Hui Tang, Wei Fan, Yuan Xiao, Dingjun Hao, Zhen Qian, Demetri Terzopoulos, et al. Self-supervised, semi-supervised, multi-context learning for the combined classification and segmentation of medical images (student abstract). In

Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pages 13815–13816, 2020.

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

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

[35] Shayan Jawed, Josif Grabocka, and Lars Schmidt-Thieme. Self-supervised learning for semi-supervised time series classification. In *Advances in Knowledge Discovery and Data Mining*, pages 499–511, Cham, 2020. Springer International Publishing.

[36] Soheil Kolouri, Se Rim Park, Matthew Thorpe, Dejan Slepcev, and Gustavo K Rohde. Optimal mass transport: Signal processing and machine-learning applications. *IEEE signal processing magazine*, 34(4):43–59, 2017.

[37] Soheil Kolouri, Yang Zou, and Gustavo K Rohde. Sliced wasserstein kernels for probability distributions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5258–5267, 2016.

[38] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[39] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

[40] Chen-Yu Lee, Tanmay Batra, Mohammad Haris Baig, and Daniel Ulbricht. Sliced wasserstein discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10285–10295, 2019.

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

[42] Xialei Liu, Joost Van De Weijer, and Andrew D Bagdanov. Exploiting unlabeled data in cnns by self-supervised learning to rank. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[43] Yanbin Liu, Makoto Yamada, Yao-Hung Hubert Tsai, Tam Le, Ruslan Salakhutdinov, and Yi Yang. Lsmi-sinkhorn: Semi-supervised squared-loss mutual information estimation with optimal transport. *arXiv preprint arXiv:1909.02373*, 2019.

[44] Yucen Luo, Jun Zhu, Mengxi Li, Yong Ren, and Bo Zhang. Smooth neighbors on teacher graphs for semi-supervised learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8896–8905, 2018.

[45] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3, 2013.

[46] Liang Mi, Wen Zhang, Xianfeng Gu, and Yalin Wang. Variational wasserstein clustering. *arXiv preprint arXiv:1806.09045*, 2018.

[47] Takeru Miyato, Shin-ichi Maeda, Shin Ishii, and Masanori Koyama. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE PAMI*, 2018.

[48] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 5, 2011.

[49] Partha Niyogi. Manifold regularization and semi-supervised learning: Some theoretical analyses. *The Journal of Machine Learning Research*, 14(1):1229–1250, 2013.

[50] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, pages 69–84. Springer, 2016.

[51] David Novotny, Samuel Albanie, Diane Larlus, and Andrea Vedaldi. Self-supervised learning of geometrically stable features through probabilistic introspection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3637–3645, 2018.

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

[53] Mohammad Peikari, Sherine Salama, Sharon Nofech-Mozes, and Anne L Martel. A cluster-then-label semi-supervised learning approach for pathology image classification. *Scientific reports*, 8(1):1–13, 2018.

[54] David Pollard. Quantization and the method of k-means. *IEEE Transactions on Information theory*, 28(2):199–205, 1982.

[55] Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. Semi-supervised learning with ladder networks. In *Advances in Neural Information Processing Systems*, pages 3546–3554, 2015.

[56] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 1163–1171, 2016.

[57] Filippo Santambrogio. Optimal transport for applied mathematicians. *Birkäuser, NY*, 55:58–63, 2015.

[58] Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. Wasserstein distance guided representation learning for domain adaptation. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[59] Vikas Sindhwani, Partha Niyogi, Mikhail Belkin, and Sathiya Keerthi. Linear manifold regularization for large scale semi-supervised learning. In *Proc. of the 22nd ICML Workshop on Learning with Partially Classified Training Data*, volume 28, 2005.

[60] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*, 2020.

[61] Justin Solomon, Raif Rustamov, Leonidas Guibas, and Adrian Butscher. Wasserstein propagation for semi-supervised learning. In *International Conference on Machine Learning*, pages 306–314, 2014.

[62] Fariborz Taherkhani, Ali Dabouei, Sobhan Soleymani, Jeremy Dawson, and Nasser M. Nasrabadi. Transporting labels via hierarchical optimal transport for semi-supervised learning. In *Computer Vision – ECCV 2020*, pages 509–526, Cham, 2020. Springer International Publishing.

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

[64] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

[65] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018.

[66] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. Unsupervised data augmentation for consistency training. *arXiv preprint arXiv:1904.12848*, 2019.

[67] Yuguang Yan, Wen Li, Hanrui Wu, Huaqing Min, Mingkui Tan, and Qingyao Wu. Semi-supervised optimal transport for heterogeneous domain adaptation. In *IJCAI*, pages 2969–2975, 2018.

[68] Jianbo Ye, Panruo Wu, James Z Wang, and Jia Li. Fast discrete distribution clustering using wasserstein barycenter with sparse support. *IEEE Transactions on Signal Processing*, 65(9):2317–2332, 2017.

[69] Xin Yi, Ekta Walia, and Paul Babyn. Unsupervised and semi-supervised learning with categorical generative adversarial networks assisted by wasserstein distance for dermoscopy image classification. *arXiv preprint arXiv:1804.03700*, 2018.

[70] Bing Yu, Jingfeng Wu, Jinwen Ma, and Zhanxing Zhu. Tangent-normal adversarial regularization for semi-supervised learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10676–10684, 2019.

[71] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[72] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. In *Proceedings of the IEEE international conference on computer vision*, pages 1476–1485, 2019.

[73] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[74] Liheng Zhang, Guo-Jun Qi, Liqiang Wang, and Jiebo Luo. Aet vs. aed: Unsupervised representation learning by auto-encoding transformations rather than data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2547–2555, 2019.

[75] Dengyong Zhou, Olivier Bousquet, Thomas Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. *Advances in neural information processing systems*, 16:321–328, 2003.

[76] Denny Zhou, Olivier Bousquet, Thomas N Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *Advances in neural information processing systems*, pages 321–328, 2004.

[77] Xiaojin Zhu. Semi-supervised learning literature survey. *Computer Science, University of Wisconsin-Madison*, 2(3):4, 2006.

[78] Xiaojin Zhu, Zoubin Ghahramani, and John D Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 20th International conference on Machine learning (ICML-03)*, pages 912–919, 2003.