SEAL: Simultaneous Label Hierarchy Exploration And Learning

Anonymous authors
Paper under double-blind review

Abstract

Label hierarchy is an important source of external knowledge that can enhance classification performance. However, most existing methods rely on predefined label hierarchies that may not match the data distribution. To address this issue, we propose **S**imultaneous label hierarchy **E**xploration **A**nd **L**earning (SEAL), a new framework that explores the label hierarchy by augmenting the observed labels with latent labels that follow a prior hierarchical structure. Our approach uses a 1-Wasserstein metric over the tree metric space as an objective function, which enables us to simultaneously learn a data-driven label hierarchy and perform (semi-)supervised learning. We evaluate our method on several standard benchmarks and show that it achieves improved results in semi-supervised image classification scenarios.

1 Introduction

Labels play a crucial role in machine learning. They provide the supervision signal for learning models from annotated data. However, obtaining label annotations is often costly and time-consuming, which motivates the study of semi-supervised learning that leverages both labeled and unlabeled data (Chen et al., 2020; Assran et al., 2021; Wang et al., 2021a). A common technique for semi-supervised learning is also related to the label, specifically, using (pseudo-)labels (Berthelot et al., 2019b; Sohn et al., 2020; Gong et al., 2021). Unlabeled data is augmented in different ways (Xie et al., 2020), and pseudo labels are then generated from model predictions for different augmentations of the same data. The model is updated by enforcing the consistency of pseudo labels across augmentations. This technique is known as "consistency regularization" (Rasmus et al., 2015b).

Labels are also important for understanding data, as they link real-world observations with abstract semantics. It has been shown that exploiting hierarchical structures of label semantics can enhance the performance of supervised and semi-supervised learning. These structures can be obtained from external sources such as decision trees (Wan et al., 2020) and knowledge graphs (Miller, 1998; Speer et al., 2017). Once the label hierarchy is available, models can be trained by either (1) predicting hierarchical semantic embeddings jointly with labels (Deng et al., 2010; 2012; Barz & Denzler, 2019; Liu et al., 2020; Wang et al., 2021b; Karthik et al., 2021; Nassar et al., 2021) or (2) optimizing hierarchical objective functions that incorporate label relations (Bertinetto et al., 2020; Bilal et al., 2017; Garnot & Landrieu, 2020; Garg et al., 2022b). Alternatively, the structure can also be incorporated into the model architecture itself (Kontschieder et al., 2015; Wu et al., 2016; Wan et al., 2020; Chang et al., 2021; Garg et al., 2022a).

Although predefined label hierarchies are frequently used in the existing literature, they cannot always match the actual data distribution. However, not much effort has been made to derive the label hierarchy from a data-driven perspective. To address this issue, we propose **S**imultaneous label hierarchy **E**xploration **A**nd **L**earning (SEAL), which achieves two goals by incorporating an additional regularization term.

The first goal of SEAL is to identify the data-driven label hierarchy. This goal differs from hierarchical clustering, which discovers hierarchical structures from data that do not align with labels. SEAL expands the label set by adding unobserved *latent* labels to the *observed* label set. The data-driven label hierarchy is modeled by combining the predefined hierarchical structure of *latent* labels with the optimizable assignment

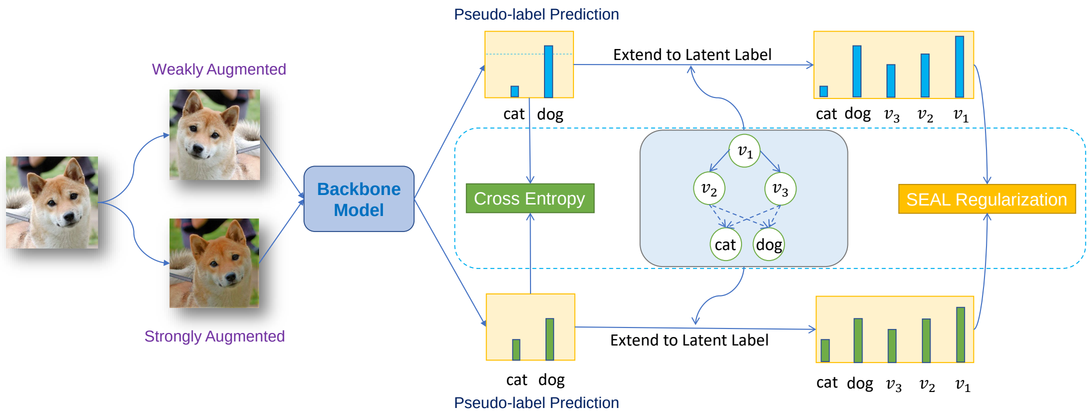


Figure 1: An illustration of the SEAL framework. After getting the predicted probability of a weakly augmented image on *observed* labels, the model shall use any pseudo-labeling techniques to give it a pseudo-label. Then the cross entropy loss between the pseudo label and predicted probability on the strong-augmented image will comprise part of the SEAL regularization. Given a (soft) tree hierarchy can introduce the *latent* labels. Extending the predicted probability on strong-augmented image and pseudo label to *total* labels will give a SEAL regularization loss which is another part of the SEAL regularization. Note that the backbone model and the SEAL regularization model (label hierarchies) can be updated via gradient descent simultaneously.

between observed and latent labels. The observation that inspires this approach is that the labels are typically a subset of concepts in a larger knowledge graph (Miller, 1998).

The second goal of SEAL is to improve the classification performance of state-of-the-art methods. To accomplish this goal, we propose a new regularization term that uses the model predictions and the label annotations on the *observed* label set to encourage agreement on both *observed* and *latent* label set. The confidence on *latent* labels is estimated by simulating the Markov chain based on the label hierarchy between *observed* and *latent* labels. This regularization term can be added to all existing approaches because of the universality of comparing the prediction and labels on the *observed* label set. A diagram of SEAL is shown in Figure 1.

SEAL's soundness and effectiveness are validated theoretically and empirically. The regularization term can be interpreted as a relaxation of the tree Wasserstein metric (Le et al., 2019), and it can be used for optimization. Empirical evaluation demonstrates that adding the SEAL framework consistently and significantly improves the classification performance on supervised learning and various semi-supervised learning methods (Sohn et al., 2020; Zhang et al., 2021; Wang et al., 2022a). SEAL also achieved a new state-of-the-art for semi-supervised learning on standard datasets. Additionally, our case study shows that the alignment between *observed* and *latent* labels also yields a meaningful label hierarchy.

2 Background

In this section, we introduce the notations used throughout this paper first and then define the Tree-Wasserstein distance, which we use to define label hierarchy metrics in Section 3.

2.1 Notations

We consider supervised learning, where $D = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ is the labeled training set. Here, $\boldsymbol{x}_i$ is an image, and y_i is the corresponding (class) label in a set $\mathcal{O}$, where $\mid \mathcal{O} \mid= K$. Without further justification, y is the categorical label, and δ_y is the one-hot label. Our goal is to learn a backbone model f_θ parameterized by θ, which maps each image $\boldsymbol{x}_i$ to a probability over $\mathcal{O}$. We denote the predicted probability vector of image

$\boldsymbol{x}$ as $f_\theta(\boldsymbol{x})$ and define the pseudo label of $\boldsymbol{x}$ as $\hat{y} = \text{argmax}_j\, f(\boldsymbol{x})^T e_j = \text{argmax}_j \Pr(j|\boldsymbol{x})$, where e_j is the coordinate vector. The objective of supervised learning is to make the ground truth label y and pseudo label $\hat{y}$ consistent.

We also consider semi-supervised learning, where an additional unlabeled dataset D^u is provided. During training, we define the relative ratio μ as the number of unlabeled data to labeled data in a mini-batch. Following RandAugment (Cubuk et al., 2020), we shall define weak augmentation $P_{\text{wAug}}(\boldsymbol{x})$ and strong augmentation $P_{\text{sAug}}(\boldsymbol{x})$. These two augmentations are probability distributions of the views of $\boldsymbol{x}$ derived by argumentation while keeping the same pseudo label, the terms weak and strong describe the distortion density. Moreover, $\boldsymbol{x}'$ and $\boldsymbol{x}''$ denote the argument data and y' denotes the pseudo-label.

2.2 Tree-Wasserstein distance

The Tree-Wasserstein distance (Le et al., 2019) is a 1-Wasserstein metric (Peyré et al., 2019) on a tree metric space $(\mathcal{X}, E, w)$, where $\mathcal{X}$ is the node set of a directed rooted tree, E is the (weighted) edge set, and $w = (w_e)_{e\in E}$ denotes the weights of edges $e \in E$. The tree metric $d_\mathcal{X}$ between any two nodes of a tree is defined as the length of the shortest path between them. Given node $v \in \mathcal{X}$, let $\Gamma(v)$ be the set of nodes in the subtree of $\mathcal{X}$ whose root node is v. For each weighted edge $e \in E$, we denote the deeper (away from the root) level endpoints of weighted edge e as v_e. Then, the Tree-Wasserstein metric can be computed in closed form, as shown in Theorem 1.

Theorem 1 (Tree-Wasserstein distance (Le et al., 2019)) *Given two probability measures μ, ν supported on a directed rooted tree $\mathcal{X}$, and choosing the ground metric as tree metric $d_\mathcal{X}$, then the Wasserstein-1 distance under Definition 4 can be reformulated as follows:*

$$W_{d_\mathcal{X}}(\mu,\nu) = \sum_{e\in E} w_e \left|\mu\left(\Gamma\left(v_e\right)\right) - \nu\left(\Gamma\left(v_e\right)\right)\right|. \tag{1}$$

3 The SEAL framework

We propose a novel framework called Simultaneous label hierarchy Exploration and Learning (SEAL), which is motivated by probabilistic models that leverage latent structures to improve classification performance. In the SEAL framework, we introduce *latent* labels that are aligned with *observed* labels to capture the data-driven label hierarchy.

3.1 Why latent structure?

Example 1 *Let us consider a classifier that predicts the probability of an image belonging whether an image belongs to the "apple" or "paint" class. Although the two classes may seem unrelated, they may share some hidden structures, such as the colors "red" and "green". Knowing the conditional probabilities of colors given each class, we can calculate the probability of an image being red given its probability of belonging to the "apple" class. Suppose we know the conditional probabilities* $\Pr(\textit{red}|\textit{apple}) = 0.9$, $\Pr(\textit{green}|\textit{apple}) = 0.1$, $\Pr(\textit{red}|\textit{paint}) = 0.5$, *and* $\Pr(\textit{green}|\textit{paint}) = 0.5$. *In this scenario, if the classifier assigns a probability of* 0.8 *for an image being an apple, the question arises as to what is the probability that this image is red. By applying the law of probability, the answer is* $0.8 \times 0.9 + 0.2 \times 0.5 = 0.82$. *For a pictorial understanding, please see Figure 2.*

3.2 Latent structure and SEAL regularization

The Example 1 illustrates that considering latent variables, such as color in this case, can provide more information to the classifier, leading to better performance. In the context of image classification, these latent variables can represent various factors, such as textures, shapes, and semantic meanings. However, identifying these latent variables and modeling their relationships with observed variables is not always straightforward, which is the focus of the SEAL framework.

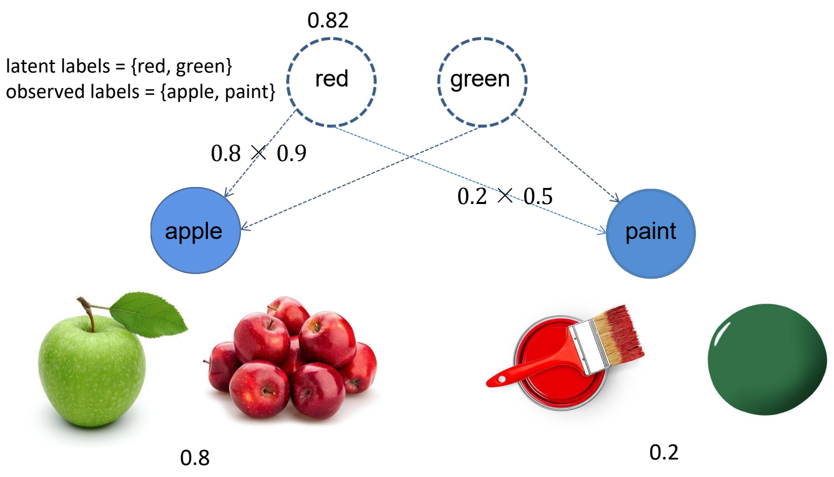

0.82
latent labels = {red, green}
observed labels = {apple, paint}
red
green
0.8 × 0.9
0.2 × 0.5
apple
paint
0.8
0.2

Figure 2: An illustrative example

Then we formally present the following definitions below. The set of labels $\mathcal{O}$ in the dataset is denoted as *observed* label set. The set of latent labels $\mathcal{L}$ is denoted as *latent* label set. We call $\mathcal{O} \cup \mathcal{L}$ *total* label set. Let $|\mathcal{O}| = K$ and $|\mathcal{O} \cup \mathcal{L}| = N$ be the sizes of observed and total label alphabets. The relationship between *observed* and *latent* labels are described by (directed) graphs. Specifically, let $\mathbf{A}_1 \in \mathbb{R}^{(N-K)\times(N-K)}$ be the adjacency matrix for latent labels in $\mathcal{L}$ and $\mathbf{A}_2 \in \mathbb{R}^{(N-K)\times K}$ be the connection matrix between $\mathcal{L}$ and $\mathcal{O}$. The adjacency matrix $\mathbf{A}$ for $\mathcal{L} \cup \mathcal{O}$ characterizes the latent structure.

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ 0 & 0 \end{pmatrix}. \tag{2}$$

It is assumed that no connection exists inside $\mathcal{O}$. SEAL is targeted to discover the hierarchical structure of labels, therefore, additional assumptions are imposed on matrix $\mathbf{A}$. The key assumption of $\mathbf{A}$ follows the following theorem.

Theorem 2 ((Takezawa et al., 2021)) *Suppose a directed graph G having a total of N nodes, which we denote as $\{v_1, v_2, ..., v_N\}$. If the adjacency matrix $\mathbf{A} \in \{0,1\}^{N\times N}$ of this graph satisfies the following conditions:*

1. *$\mathbf{A}$ is a strictly upper triangular matrix.*
2. *$\mathbf{A}^T \mathbf{1}_N = (0, 1, \cdots, 1)^\top$.*

then G is a directed rooted tree with v_1 as the root and $\mathbf{1}_N$ is an all-one vector.

The hierarchical structure of observed labels is then described by the graph defined by $\mathbf{A}$, satisfying conditions in Theorem 2.

We introduce a weight matrix $\alpha = [\alpha_{sr}]_{s,r} \in \mathbb{R}^{(N-K)\times K}$ that describes the connection from $s \in \mathcal{L}$ to $r \in \mathcal{O}$ to further quantify how much an observed label contributes to a latent label. A **SEAL extension** is then defined by a five-tuple $(\mathcal{O}, \mathcal{L}, \mathbf{A}_1, \mathbf{A}_2, \alpha)$. We note that α should be related to the $\mathbf{A}_2$ and its specific formulation will be detailed in the following parts. Then, we are able to extend the model's prediction on the observed label set $\mathcal{O}$ to the total label set.

Definition 1 (Total prediction and total target) *Let p be a probability vector on $\mathcal{O}$ and p_r be the probability of the label $r \in \mathcal{O}$, vector q on* total *label set $\mathcal{O} \cup \mathcal{L}$ is*

$$[q(p)]_s = \begin{cases} p_s & \text{if } s \in \mathcal{O} \\ \sum_{r\in\mathcal{O}} \alpha_{sr} p_r & \text{if } s \in \mathcal{L} \end{cases}, \tag{3}$$

where $s \in \mathcal{O} \cup \mathcal{L}$. Given a sample of input and (pseudo-)label $(\boldsymbol{x}, y)$, we note that the p_r could be derived by both the model prediction $f_\theta(\boldsymbol{x})$, the one-hot label δ_y, or the pseudo-label $\delta_{y'}$. Moreover, $q(f_\theta(\boldsymbol{x}))$ is the total prediction while $q(\delta_y)$ is denoted as the total target.

We note that q is not the probability in any case, since it extends the original probability p over $\mathcal{O}$ by further considering the aggregations over $\mathcal{L}$. However, it is also sufficient to define objective functions to minimize the differences between total prediction and total target, which is SEAL regularization.

Definition 2 (SEAL regularization) *Given input $\boldsymbol{x}$, target y, model f_θ, and a SEAL extension $(\mathcal{O}, \mathcal{L}, \mathbf{A}_1, \mathbf{A}_2, \alpha)$, the SEAL regularization is defined as $\phi(f_\theta(\boldsymbol{x}), \delta_y) = D(q(f_\theta(\boldsymbol{x})), q(\delta_y))$, where D is a distance function.*

In this paper, we consider SEAL regularization where D is the weighted ℓ_1 metric:

$$\phi(f_\theta(\boldsymbol{x}), \delta_y) = D(q(f_\theta(\boldsymbol{x})), q(\delta_y)) = \sum_{s \in \mathcal{O} \cup \mathcal{L}} w_s |[q(f_\theta(\boldsymbol{x}))]_s - [q(\delta_y)]_s|, \tag{4}$$

where w_s is the weight for each observed or latent label.

We have presented the basic framework of SEAL, then we detail how SEAL is used to explore label hierarchy and improve learning in the next parts.

3.3 Label Hierarchy Exploration with SEAL

In this section, we explore the label hierarchy under the SEAL extension $(\mathcal{O}, \mathcal{L}, \mathbf{A}_1, \mathbf{A}_2, \alpha)$. To achieve this, we first specify $\mathcal{L}$, $\mathbf{A}_1$, $\mathbf{A}_2$, and α, which breakdowns into two tasks. The first task is to specify $\mathcal{L}$ and $\mathbf{A}_1$ to define the prior structure inside $\mathcal{L}$, while the second task is to specify how α and $\mathbf{A}_2$ are related to defining how the structure is optimized.

Task (a): Prior Structure Specification. To specify the prior structure inside $\mathcal{L}$, we choose $\mathbf{A}_1$ to be a binary tree or trees derived from prior knowledge such as a part of a knowledge graph or a decision tree. This choice of $\mathbf{A}_1$ allows us to control the prior structure of the label hierarchy and incorporate prior domain knowledge. Additionally, we can use the hierarchical structure of $\mathbf{A}_1$ to guide the training of the model to improve performance.

Task (b): Structure Optimization Specification. To specify how α and $\mathbf{A}_2$ are related to defining how the structure is optimized, we note that α and $\mathbf{A}_2$ both reflect how the structure interacts with the model and data. Specifically, we compute α from $\mathbf{A}_2$ from the Markov chain on trees. This choice of α emphasizes more on the prediction of the model while $\mathbf{A}_2$ emphasizes more on the interpretation of the label hierarchy.

In summary, by specifying $\mathcal{L}$, $\mathbf{A}_1$, $\mathbf{A}_2$, and α, we can explore the label hierarchy under the SEAL extension $(\mathcal{O}, \mathcal{L}, \mathbf{A}_1, \mathbf{A}_2, \alpha)$. This approach allows us to incorporate prior domain knowledge and guide the training of the model to improve performance while also providing a framework for interpreting the label hierarchy.

Random Walk Construction of α. We observe that the matrix $\mathbf{A}$ satisfies the conditions in Theorem 2, and can be viewed as a Markov transition matrix on $\mathcal{L} \cup \mathcal{O}$ that follows the top-down direction over a tree. Therefore, the probability of a random walk from a node $s \in \mathcal{L}$ to a node $r \in \mathcal{O}$ can be computed by simulating the Markov chain. We define α_{sr} to be the probability of random walks starting from s and ending at r, which can be interpreted as the probability that node r is contained in the subtree of node s. Specifically, we have:

$$\alpha_{sr} = \left[\sum_{k=1}^{\infty} \mathbf{A}^k\right]_{sr} = \left[(\mathbf{I} - \mathbf{A})^{-1}\right]_{sr}, \tag{5}$$

where $\mathbf{I}$ is the identity matrix.

Moreover, we can further simplify the above equation by noting that $(\mathbf{I}-\mathbf{A})^{-1}$ can be precomputed. Specifically, we have: $(\mathbf{I}-\mathbf{A})^{-1} = \begin{pmatrix} (\mathbf{I}-\mathbf{A}_1)^{-1} & (\mathbf{I}-\mathbf{A}_1)^{-1}\mathbf{A}_2 \\ 0 & \mathbf{I} \end{pmatrix}$.

This construction of α is based on the Markov chain on trees and has been applied in other applications such as document distance (Takezawa et al., 2021) and hierarchical node clustering (Zügner et al., 2021). However, it is the first time that this construction has been used to train a deep neural network.

Optimizing SEAL regularization Once α is defined explicitly through $\mathbf{A}_2$, the expression of SEAL regularization Φ is also well defined. It simplifies to

$$\phi(f_\theta(\boldsymbol{x}), \delta_y) = (w^\top \begin{pmatrix} (\mathbf{I}-\mathbf{A}_1)^{-1}\mathbf{A}_2 \\ \mathbf{I} \end{pmatrix} (f_\theta(\boldsymbol{x}) - \delta_y))^{|\cdot|}, \tag{6}$$

where ${}^{|\cdot|}$ denotes taking the element-wise absolute value. We set $w = \mathbf{1}_N$ for simplicity.

One could jointly optimize θ and $\mathbf{A}_2$ (or α). Particular attention should be paid to $\mathbf{A}_2$ since it is discrete and required to satisfy the conditions in Theorem 2, making the optimization very hard. In this paper, we relax $\mathbf{A}_2 \in \{0,1\}^{(N-K)\times K}$ to $\mathbf{A}_2^{\text{soft}} \in [0,1]^{(N-K)\times K}$. One could employ projected gradient descent on each column of $\mathbf{A}_2^{\text{soft}}$ to ensure those conditions. More investigation on optimization can be found in Section 6.4.

For clarity, we denote the SEAL regularization as $\phi_\Theta(f_\theta(\boldsymbol{x}), \delta_y)$, where the suffix Θ denotes the parameters defining latent hierarchy.

Interpreting SEAL results After $\mathbf{A}_2^{\text{soft}}$ is optimized, we can reconstruct $\mathbf{A}_2$ to interpret the explored label hierarchy. Specifically

$$(\mathbf{A}_2)_{sr} = \begin{cases} 1 & \text{if } s = \arg\max_{k\in\mathcal{L}} (\mathbf{A}_2^{soft})_{kr} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

Then the matrix $\mathbf{A}$ is derived after optimization.

3.4 Learning with SEAL

We have already defined SEAL regularization based on the model output $f_\theta(\boldsymbol{x})$ and the target y. Then it is natural to apply SEAL regularization to various learning scenarios.

We consider the learning process in a typical mini-batch setting. Given a batch of samples $B \subset D$, we consider the averaged summation Φ of SEAL regularization ϕ over the batch.

$$\Phi(\theta, \Theta; B) = \frac{1}{|B|} \sum_{(\boldsymbol{x},y)\in B} \phi_\Theta(f_\theta(\boldsymbol{x}), \delta_y). \tag{8}$$

We note that δ_y could be the one-hot labels of the labeled data or pseudo-labels on unlabeled data.

3.4.1 Semi-supervised learning with SEAL

Consider a general scheme (Gong et al., 2021) that unifies many prior semi-supervised algorithms. For the $n+1$-th iteration, the model parameter is derived based on supervised loss $L(\theta)$ and a consistency regularization term $\Psi(\theta;\theta_n)$. Specifically

$$\theta_{n+1} \leftarrow \arg\min_\theta \{L(\theta) + \gamma\Psi(\theta;\theta_n)\}, \tag{9}$$

where θ_n denotes the model parameters at the n-th iteration and γ is the loss balancing coefficient.

Adding SEAL to semi-supervised learning is no more than applying SEAL regularization as an additional consistency regularization term apart from the initial $\Psi(\theta;\theta_n)$, which will be described below.

Usually speaking, the computation $\Psi(\theta;\theta_n)$ is conducted over a batch of unlabeled data $B^u \subset D^u$. For each sample $\boldsymbol{x} \in B^u$, the computation follows the following process:

1. **Pseudo-label prediction on weak augmentation** Computing the prediction of weakly-augmented image $\boldsymbol{x}' \in P_{\mathrm{wAug}}(\boldsymbol{x})$ with model f_{θ_n} in the last iteration, which will be used to generate pseudo-labels in the next steps.
2. **Strong augmentation** For the input data $\boldsymbol{x}$, we sample strong argumentation $\boldsymbol{x}'' \in P_{\mathrm{sAug}}(x)$.
3. **Selection** Some selection processes are applied to select the samples and assign them meaningful pseudo-labels y's. This results in a new pseudo-labeled dataset $\hat{B}^u := \{(\boldsymbol{x}'', y')\}$.

Consistency regularization minimizes the differences between the prediction by the model and the pseudo-label y', for example, prior works usually merely use the cross entropy as $\Psi(\theta; \theta_n)$:

$$\Psi(\theta, \theta_n) = \frac{1}{|\hat{B}^u|} \sum_{(\boldsymbol{x}'', y') \in \hat{B}^u} \mathrm{CE}(f_{\theta_n}(\boldsymbol{x}''), \delta_{y'}). \tag{10}$$

Adding SEAL regularization is simply adding another term $\Phi(\theta, \Theta, \hat{B}^u)$. Then, we obtain the updating rule of semi-supervised learning with SEAL:

$$\theta_{n+1} \leftarrow \arg\min_{\theta} \left\{ L(\theta) + \gamma \left[\Psi(\theta; \theta_n) + \lambda \Phi(\theta, \Theta; \hat{B}^u) \right] \right\}. \tag{11}$$

Many existing approaches (Sohn et al., 2020; Zhang et al., 2021; Wang et al., 2022a) fit into this paradigm. So SEAL can be applied easily to such approaches.

Summary for SEAL We have demonstrated how to plugin SEAL with supervised and semi-supervised learning in equations (16) and (11), respectively. The key observation is that SEAL regularization can be applied as long as there is an objective function between model prediction $f(\boldsymbol{x})$ and target y.

4 Theoretical analysis of SEAL regularization

We find equation (6) has a similar structure to that of Tree-Wasserstein distance, so we shall first extend the definition of Tree-Wasserstein distance.

We shall first rewrite the Tree-Wasserstein distance's summation using the node as indices. Note the lower endpoint of each edge has a one-to-one correspondence with each node, thus we can see the weight of each edge as the weight of each node. Denote the tree as $\mathcal{X}$ and leaf nodes as $\mathcal{X}_{leaf}$. We can rewrite the expression of Tree-Wasserstein distance into $W_{d_{\mathcal{X}}}(\mu, \nu) = \sum_{v \in \mathcal{X}} w_v |\mu(\Gamma(v)) - \nu(\Gamma(v))|$. When μ and ν are supported only on the leaf set, we can rewrite $\Gamma(v)$ using the ancestor-child relationship. That is,

$$W_{d_{\mathcal{X}}}(\mu, \nu) = \sum_{v \in \mathcal{X}} w_v | \sum_{x \in \mathcal{X}_{\mathrm{leaf}}} (\mu(x) - \nu(x)) \mathbb{I}_{\mathrm{v\ is\ ancestor\ of\ x}} |. \tag{12}$$

If a directed graph has its adjacency matrix A satisfying the conditions in Theorem 2 except relaxing the hard constraint $\{0, 1\}$ to $[0, 1]$, we shall call it a soft tree. Recall the subtree probabilistic interpretation of α in 3.3, we can define relaxed Tree-Wasserstein distance (RTW) as below.

Definition 3 (Relaxed Tree-Wasserstein distance) *Assume $\mathcal{X}$ is a soft tree and denote the leaf nodes as $\mathcal{X}_{leaf}$. For any two probability measures supported on $\mathcal{X}_{leaf}$. The relaxed tree Wasserstein distance $W^{relax}_{d_{\mathcal{X}}}(\mu, \nu)$ is given as follows:*

$$W^{relax}_{d_{\mathcal{X}}}(\mu, \nu) = \sum_{v \in \mathcal{X}} w_v \left| \sum_{x \in \mathcal{X}_{leaf}} \alpha_{vx} (\mu(x) - \nu(x)) \right|. \tag{13}$$

If we let $\mathcal{X}_{\mathrm{leaf}}$ be the set of *observed* labels $\mathcal{O}$ and $\mathcal{X}$ be *total* labels $\mathcal{O} \cup \mathcal{L}$. We can then show the connection between the relaxed tree Wasserstein distance and the weighted total classification error given by equation (6).

Theorem 3 *The weighted total classification loss described by Eqn. (6) under* $\mathbf{A}_2^{soft}$ *coincides with* $W_{d_{\mathcal{X}}}^{relax}(f_\theta(\boldsymbol{x}), \delta_y)$*.*

Proof 1 *Please see Appendix A.4.3.*

5 Applying SEAL to semi-supervised learning

6 Implementation details

6.0.1 Implementation of SEAL

Defining SEAL extension $(\mathcal{O}, \mathcal{L}, \mathbf{A}_1, \mathbf{A}_2, \alpha)$**.** For fair comparison and injecting no prior knowledge, $\mathbf{A}_1$ and $\mathbf{A}_2^{soft}$ are both randomly initialized. For CIFAR10 and STL-10, $K = 10$ and $N = 21$. For CIFAR100, $K = 100$ and $N = 130$. More ablations and details can be found in Section 6.4.1 and Appendix A.2.

Combining SEAL with other learning methods. The default setting of SEAL for semi-supervised learning is adopted from the same configuration and hyper-parameters used in FixMatch (Sohn et al., 2020). SEAL (Curriculum) adopts the curriculum pseudo-labeling technique and hyper-parameters used in FlexMatch (Zhang et al., 2021). SEAL (Debiased) adopts the debiasing trick and hyper-parameters used in DebiasPL (Wang et al., 2022a). Note in experiments we only conduct SEAL Debiased on CIFAR 10 as the initial paper DebiasPL (Wang et al., 2022a) did not conduct experiments on CIFAR 100 and STL-10.

Optimizing with SEAL regularization. Specifically, we use a (batch) stochastic gradient descent (SGD) optimizer with a momentum of 0.9. We set the learning rate scheduler as the cosine decay scheduler, where the learning rate β can be expressed as $\beta = \beta_0 \mathbf{cos}(\frac{7\pi}{16}\frac{s}{S})$. Here, β_0 is the initial learning rate set to 0.03, s is the current optimization step, and S is the total number of optimization steps set to 2^{20}. We set the batch size of the labeled training data to 64, and the ratio of unlabeled training data to labeled data μ is set to 7. We set the threshold τ to 0.95, and the weak and strong augmentation functions used in our experiments are based on RandAugment(Cubuk et al., 2020). We use WideResNet-28-2 as the backbone model for our experiments. Also, SEAL improves standard supervised learning, details can be found in Appendix A.5.

6.1 Baselines

The baselines we consider in our experiments are those prior works similar to FixMatch, such as Π-Model (Laine & Aila, 2016), Pseudo Label (Lee et al., 2013), Mean Teacher (Tarvainen & Valpola, 2017), MixMatch (Berthelot et al., 2019b), ReMixMatch (Berthelot et al., 2019a), VAT (Miyato et al., 2018), UDA (Xie et al., 2020), FlexMatch (Zhang et al., 2021)and DebiasPL (Wang et al., 2022a). However, we find that our proposed method SEAL is lightweight yet effective and outperforms all of these baselines on all three datasets in nearly all settings.

6.2 Findings

SEAL is lightweight and effective. SEAL is easy to implement and with the aid of SEAL and its variants, we can achieve state-of-art results on all three datasets under all label amount settings. Results can be found in Tables 1 and 2.

The fewer labeled data, the more significant improvements. Interestingly, we observe that the fewer labeled data available, the more significant gains we can achieve using SEAL. For instance, on CIFAR10, we obtain a remarkable 7.52 accuracy gain with only 40 labeled data, while we only see a 0.64 accuracy gain with 250 labeled data. This finding highlights the effectiveness of our proposed method in situations where labeled data is scarce.

SEAL can be boosted by various techniques. Moreover, we demonstrate that our proposed method can be further enhanced by incorporating various existing semi-supervised learning techniques, such as **Curriculum** Pseudo Label (Zhang et al., 2021) and **Debiased** Pseudo Label (Wang et al., 2022a), into

Table 1: Error rates (100% - accuracy) on CIFAR-10. **bold** means the best.

Dataset	CIFAR-10	
# Label	40	250
Π Model (Rasmus et al., 2015a)	74.34±1.76	46.24±1.29
Pseudo Label (Lee et al., 2013)	74.61±0.26	46.49±2.20
VAT (Miyato et al., 2018)	74.66±2.12	41.03±1.79
MeanTeacher (Tarvainen & Valpola, 2017)	70.09±1.60	37.46±3.30
MixMatch (Berthelot et al., 2019b)	36.19±6.48	13.63±0.59
ReMixMatch (Berthelot et al., 2019a)	9.88±1.03	6.30±0.05
UDA (Xie et al., 2020)	10.62±3.75	5.16±0.06
FixMatch (Sohn et al., 2020)	7.47±0.28	5.07±0.05
Dash (Xu et al., 2021)	8.93±3.11	5.16±0.23
MPL (Pham et al., 2021)	6.93±0.17	5.76±0.24
FlexMatch (Zhang et al., 2021)	4.97±0.06	4.98±0.09
DebiasPL (Wang et al., 2022a)	5.40± 1.30	5.60± 0.10
SEAL	6.29± 0.58	4.43± 0.55
SEAL (Curriculum)	4.90± 0.09	4.92± 0.04
SEAL (Debiased)	**4.66± 0.07**	**4.41± 0.18**

Table 2: Error rates (100% - accuracy) on CIFAR-10, CIFAR-100, and STL-10 dataset of state-of-the-art methods for semi-supervised learning. **bold** means the best.

Dataset	CIFAR-100		STL-10	
# Label	400	2500	40	250
Π Model (Rasmus et al., 2015a)	86.96±0.80	58.80±0.66	74.31±0.85	55.13±1.50
Pseudo Label (Lee et al., 2013)	87.45±0.85	57.74±0.28	74.68±0.99	55.45±2.43
VAT (Miyato et al., 2018)	85.20±1.40	46.84±0.79	74.74±0.38	56.42±1.97
MeanTeacher (Tarvainen & Valpola, 2017)	81.11±1.44	45.17±1.06	71.72±1.45	56.49±2.75
MixMatch (Berthelot et al., 2019b)	67.59±0.66	39.76±0.48	54.93±0.96	34.52±0.32
UDA (Xie et al., 2020)	46.39±1.59	27.73±0.21	37.42±8.44	9.72±1.15
Dash (Xu et al., 2021)	44.82±0.96	27.15±0.22	34.52±4.30	-
MPL (Pham et al., 2021)	46.26±1.84	27.71±0.19	35.76±4.83	9.90±0.96
FixMatch (Sohn et al., 2020)	46.42±0.82	28.03±0.16	35.97±4.14	9.81±1.04
FlexMatch (Zhang et al., 2021)	39.94±1.62	26.49±0.20	29.15±4.16	8.23±0.39
SEAL	45.18±0.79	26.99±0.23	33.27±3.21	9.77±0.54
SEAL (Curriculum)	**39.73±1.58**	**26.39±0.29**	**16.15±2.63**	**7.92±0.47**

SEAL framework with minimal effort. This implies that any future work on improving the quality of pseudo labels can be easily adapted into our SEAL framework.

Meaningful posterior tree structure. We present a learned tree structure in Figure 3. We can find semantically close classes are indeed close on the tree. For example, animals cat, dog, and deer are all child nodes of node 5.

6.3 Ablation Studies

In this section, we focus on analyzing the influence of different parameters on the performance of our proposed method, SEAL (Debiased), using the CIFAR10 dataset with only 40 labeled samples.

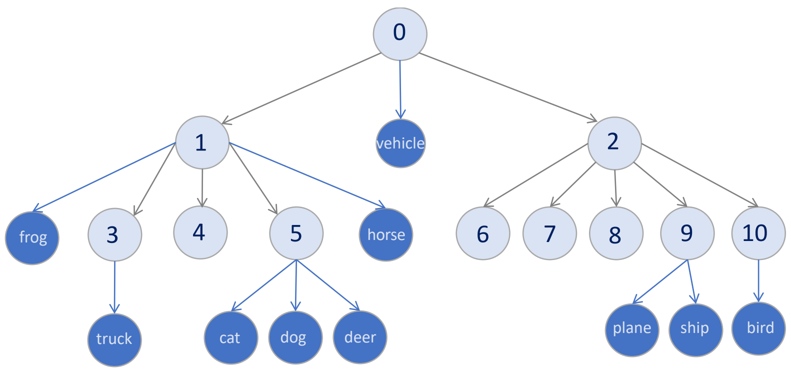


Figure 3: A hierarchy of posterior tree structure

6.3.1 Different regularizer

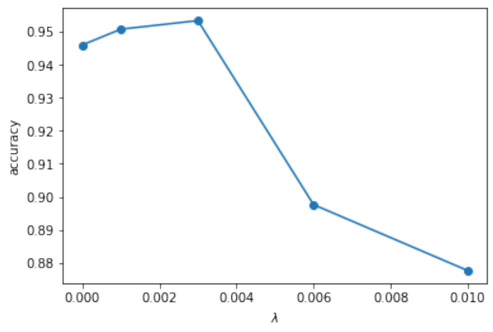


Figure 4: The influence of different hyper-parameter

Lastly, we explore the impact of the regularizer on classification accuracy by experimenting with 5 different λ values, including the case of $\lambda = 0$ to illustrate the necessity of the regularizer. As shown in Figure 4, when λ approaches 0.003, the accuracy increases, while it decreases as λ deviates from 0.003. Therefore, we conclude that the optimal value of λ is around 0.003. We note that the accuracy drops significantly when λ is too large, which may be attributed to the imbalance in loss scale.

6.4 Different updating rule for the adjacency matrix

Efficient updating of the soft adjacency matrix $\mathbf{A}_2$ is essential in the experiments. Two popular approaches have been used for updating $\mathbf{A}_2$. One approach is to consider each column of $\mathbf{A}_2$ as the realization of softmax mapping. The other approach is to use projected gradient descent (PGD) to update the matrix, projecting each column onto the probability simplex. The classification accuracies of both methods are summarized in Table 3.

Table 3: Different methods for updating adjacency matrix

Update Method	CIFAR10-40 labels
PGD	95.34
Softmax	86.46

It is evident that PGD has a significant advantage over the softmax mapping-based approach. This observation is also reported in (Zügner et al., 2021). The poor performance of the softmax-based approach may be attributed to a bad initialization, where the optimization is trapped by a bad starting point.

6.4.1 Different tree structure

Table 4: Classification under different tree structure

Tree Name	Acc. on CIFAR10 (40 labels)
Without Tree	94.60
Trivial Tree	95.18
Random Tree	95.34
NBDT (Wan et al., 2020) Tree	95.39

Next, we examine how the choice of tree structure affects the results. We compare three different trees: a trivial tree with all 10 classes as leaf nodes besides one root node, a randomly generated depth-4 tree with 21 nodes, and the NBDT tree proposed in (Wan et al., 2020), which has a well-designed hierarchy as shown in Figure 5. We use the adjacency matrix of the internal nodes induced subtree as the adjacency matrix A_1 in our method for each tree.

The results of the classification accuracies under these different trees are presented in Table 4. As we can see from the table, using tree structures consistently improves the classification accuracies compared to the vanilla cases. It is worth noting that the NBDT tree, which is carefully designed, achieves the highest accuracy, while our randomly generated tree performs better than the other cases.

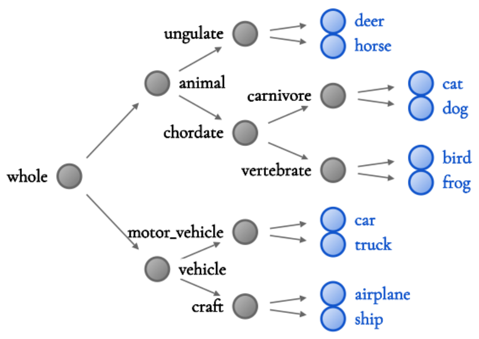


Figure 5: The hierarchy of NBDT Tree

7 Related Work

7.1 Semi-supervised Learning

Most existing methods for consistency regularization aim to improve the quality of the pseudo labels generated from unlabeled data. For example, SimPLE (Hu et al., 2021) introduces a paired loss that minimizes the statistical distance between confident and similar pseudo labels. Dash (Xu et al., 2021) and FlexMatch (Zhang et al., 2021) propose dynamic and adaptive strategies for filtering out unreliable pseudo labels during training. MaxMatch (Li et al., 2022) proposes a worst-case consistency regularization technique that minimizes the maximum inconsistency between an original unlabeled sample and its multiple augmentations with theoretical guarantees. A notable exception is SemCo (Nassar et al., 2021), which leverages external label semantics to prevent the deterioration of pseudo-label quality for visually similar classes in a co-training framework.

Though proposed in different techniques, all these methods rely on a fixed objective function to define the consistency, which is usually the cross-entropy over the label space. Our work differs from these methods by proposing a novel way to extend the cross-entropy with latent labels and hierarchical structures. Therefore, our method can complement existing methods whenever cross-entropy is used.

7.2 Label hierarchies

Label relationships are essential prior knowledge for improving model performance, and can be represented by semantic structures among labels. One prominent form of label structure is the label hierarchy (Garnot & Landrieu, 2020), which can be obtained from external sources like decision trees (Wan et al., 2020) and knowledge graphs (Miller, 1998; Speer et al., 2017). This information can be leveraged to train models as semantic embeddings (Deng et al., 2010; 2012; Barz & Denzler, 2019; Liu et al., 2020; Wang et al., 2021b; Karthik et al., 2021; Nassar et al., 2021) or objective functions (Bertinetto et al., 2020; Bilal et al., 2017; Garnot & Landrieu, 2020; Garg et al., 2022b). Additionally, the hierarchical information can also be incorporated as part of the model structure (Kontschieder et al., 2015; Wu et al., 2016; Wan et al., 2020; Chang et al., 2021; Garg et al., 2022a).

Pre-defined label hierarchies are widely acknowledged as an essential source of prior knowledge for the label space in classification. This has been extensively discussed in the literature. The label hierarchy information can be used to improve the model training process, such as by embedding the hierarchical labels to maximize the similarity between the latent embedding of the input image and the embedding of its label (Bengio et al., 2010; Deng et al., 2012; Frome et al., 2013). This idea has been generalized to various embedding spaces (Barz & Denzler, 2019; Liu et al., 2020; Garnot & Landrieu, 2020) and the joint learning scheme where image embeddings and label embeddings are both optimized (Wu et al., 2016; Chang et al., 2021). Additionally, hierarchical structures can also be explicitly used to make the training process hierarchical-aware (Deng et al., 2010; Bilal et al., 2017; Bertinetto et al., 2020; Karthik et al., 2021; Garg et al., 2022a;b). However, existing work typically treats the hierarchical label structure as prior knowledge. In contrast, our approach leverages the posterior latent label structures given the presence of labeled and unlabeled samples.

8 Conclusion and Future Work

In this paper, we propose a framework SEAL to jointly train the model of high performances and the label structure of significance. The SEAL framework is flexible to be adapted to various learning schemes, and can even incorporate the prior structure given by the external knowledge and the information given by the data. Experimental results support the effectiveness of the SEAL framework. Theoretical understanding of SEAL via optimal transport theory is also discussed. Future works may include incorporating more complex prior knowledge or applying the SEAL framework to self-supervised learning.

References

Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Armand Joulin, Nicolas Ballas, and Michael Rabbat. Semi-supervised learning of visual features by non-parametrically predicting view assignments with support samples. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8443–8452, 2021.

Björn Barz and Joachim Denzler. Hierarchy-based image embeddings for semantic image retrieval. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 638–647. IEEE, 2019.

Samy Bengio, Jason Weston, and David Grangier. Label embedding trees for large multi-class tasks. *Advances in Neural Information Processing Systems*, 23, 2010.

Christian Berg, Jens Peter Reus Christensen, and Paul Ressel. *Harmonic analysis on semigroups: theory of positive definite and related functions*, volume 100. Springer, 1984.

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*, 2019a.

David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in neural information processing systems*, 32, 2019b.

Luca Bertinetto, Romain Mueller, Konstantinos Tertikas, Sina Samangooei, and Nicholas A Lord. Making better mistakes: Leveraging class hierarchies with deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12506–12515, 2020.

Alsallakh Bilal, Amin Jourabloo, Mao Ye, Xiaoming Liu, and Liu Ren. Do convolutional neural networks learn class hierarchy? *IEEE transactions on visualization and computer graphics*, 24(1):152–162, 2017.

Dongliang Chang, Kaiyue Pang, Yixiao Zheng, Zhanyu Ma, Yi-Zhe Song, and Jun Guo. Your" flamingo" is my" bird": Fine-grained, or not. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11476–11485, 2021.

Hao Chen, Ran Tao, Yue Fan, Yidong Wang, Jindong Wang, Bernt Schiele, Xing Xie, Bhiksha Raj, and Marios Savvides. Softmatch: Addressing the quantity-quality trade-off in semi-supervised learning. *International Conference on Learning Representations (ICLR)*, 2023.

Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33:22243–22255, 2020.

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 215–223. JMLR Workshop and Conference Proceedings, 2011.

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*, pp. 702–703, 2020.

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*, pp. 248–255. Ieee, 2009.

Jia Deng, Alexander C Berg, Kai Li, and Li Fei-Fei. What does classifying more than 10,000 image categories tell us? In *European conference on computer vision*, pp. 71–84. Springer, 2010.

Jia Deng, Jonathan Krause, Alexander C Berg, and Li Fei-Fei. Hedging your bets: Optimizing accuracy-specificity trade-offs in large scale visual recognition. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3450–3457. IEEE, 2012.

Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. *Advances in neural information processing systems*, 26, 2013.

Ashima Garg, Shaurya Bagga, Yashvardhan Singh, and Saket Anand. Hiermatch: Leveraging label hierarchies for improving semi-supervised learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1015–1024, 2022a.

Ashima Garg, Depanshu Sani, and Saket Anand. Learning hierarchy aware features for reducing mistake severity. *arXiv preprint arXiv:2207.12646*, 2022b.

Vivien Sainte Fare Garnot and Loic Landrieu. Leveraging class hierarchies with metric-guided prototype learning. *arXiv preprint arXiv:2007.03047*, 2020.

Chengyue Gong, Dilin Wang, and Qiang Liu. Alphamatch: Improving consistency for semi-supervised learning with alpha-divergence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13683–13692, 2021.

Zijian Hu, Zhengyu Yang, Xuefeng Hu, and Ram Nevatia. Simple: similar pseudo label exploitation for semi-supervised classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15099–15108, 2021.

Pranav Jeevan and Amit Sethi. Vision xformers: Efficient attention for image classification. *arXiv preprint arXiv:2107.02239*, 2021.

Shyamgopal Karthik, Ameya Prabhu, Puneet K Dokania, and Vineet Gandhi. No cost likelihood manipulation at test time for making better mistakes in deep networks. *arXiv preprint arXiv:2104.00795*, 2021.

Peter Kontschieder, Madalina Fiterau, Antonio Criminisi, and Samuel Rota Bulo. Deep neural decision forests. In *Proceedings of the IEEE international conference on computer vision*, pp. 1467–1475, 2015.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Citeseer*, 2009.

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

Tam Le, Makoto Yamada, Kenji Fukumizu, and Marco Cuturi. Tree-sliced variants of wasserstein distances. *Advances in neural information processing systems*, 32, 2019.

Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, pp. 896, 2013.

Muyang Li, Runze Wu, Haoyu Liu, Jun Yu, Xun Yang, Bo Han, and Tongliang Liu. Instant: Semi-supervised learning with instance-dependent thresholds. *Advances in Neural Information Processing Systems*, 36, 2024.

Yangbangyan Jiang Xiaodan Li, Yuefeng Chen, Yuan He, Qianqian Xu, Zhiyong Yang, Xiaochun Cao, and Qingming Huang. Maxmatch: Semi-supervised learning with worst-case consistency. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

Shaoteng Liu, Jingjing Chen, Liangming Pan, Chong-Wah Ngo, Tat-Seng Chua, and Yu-Gang Jiang. Hyperbolic visual embedding learning for zero-shot recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9273–9281, 2020.

George A Miller. *WordNet: An electronic lexical database.* MIT press, 1998.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018.

Islam Nassar, Samitha Herath, Ehsan Abbasnejad, Wray Buntine, and Gholamreza Haffari. All labels are not created equal: Enhancing semi-supervision via label grouping and co-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7241–7250, 2021.

Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

Hieu Pham, Zihang Dai, Qizhe Xie, and Quoc V Le. Meta pseudo labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11557–11568, 2021.

Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. Semi-supervised learning with ladder networks. *Advances in Neural Information Processing Systems*, 28:3546–3554, 2015a.

Antti Rasmus, Harri Valpola, Mikko Honkala, Mathias Berglund, and Tapani Raiko. Semi-supervised learning with ladder network. *ArXiv*, abs/1507.02672, 2015b.

Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020.

Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

Yuki Takezawa, Ryoma Sato, and Makoto Yamada. Supervised tree-wasserstein distance. In *International Conference on Machine Learning*, pp. 10086–10095. PMLR, 2021.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017.

Alvin Wan, Lisa Dunlap, Daniel Ho, Jihan Yin, Scott Lee, Henry Jin, Suzanne Petryk, Sarah Adel Bargal, and Joseph E Gonzalez. Nbdt: neural-backed decision trees. *arXiv preprint arXiv:2004.00221*, 2020.

Xudong Wang, Long Lian, and Stella X Yu. Data-centric semi-supervised learning. *arXiv preprint arXiv:2110.03006*, 2021a.

Xudong Wang, Zhirong Wu, Long Lian, and Stella X Yu. Debiased learning from naturally imbalanced pseudo-labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14647–14657, 2022a.

Yidong Wang, Hao Chen, Qiang Heng, Wenxin Hou, Marios Savvides, Takahiro Shinozaki, Bhiksha Raj, Zhen Wu, and Jindong Wang. Freematch: Self-adaptive thresholding for semi-supervised learning. *arXiv preprint arXiv:2205.07246*, 2022b.

Yu Wang, Zhou Wang, Qinghua Hu, Yucan Zhou, and Honglei Su. Hierarchical semantic risk minimization for large-scale classification. *IEEE Transactions on Cybernetics*, 2021b.

Hui Wu, Michele Merler, Rosario Uceda-Sosa, and John R Smith. Learning to make better mistakes: Semantics-aware visual food recognition. In *Proceedings of the 24th ACM international conference on Multimedia*, pp. 172–176, 2016.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems*, 33:6256–6268, 2020.

Yi Xu, Lei Shang, Jinxing Ye, Qi Qian, Yu-Feng Li, Baigui Sun, Hao Li, and Rong Jin. Dash: Semi-supervised learning with dynamic thresholding. In *International Conference on Machine Learning*, pp. 11525–11536. PMLR, 2021.

Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Advances in Neural Information Processing Systems*, 34:18408–18419, 2021.

Daniel Zügner, Bertrand Charpentier, Morgane Ayle, Sascha Geringer, and Stephan Günnemann. End-to-end learning of probabilistic hierarchies on graphs. In *International Conference on Learning Representations*, 2021.

A Appendix

A.1 Comparision with more advanced methods

We will present more comparisons with more recently introduced methods like FreeMatch (Wang et al., 2022b), SoftMatch (Chen et al., 2023), and InstanT (Li et al., 2024). We add our SEAL regularization to the existing method FreeMatch. In Table 5, we can see our method shows improvements upon baseline FreeMatch and also behaves good even considering the other two methods SoftMatch and InstanT, showing the effectiveness of our method.

Table 5: Error rates compared with more advanced algorithms.

	CIFAR10 40 labels	CIFAR10 250 labels	CIFAR100 400 labels	CIFAR100 2500 labels
FreeMatch	4.9	4.88	37.98	26.47
SoftMatch	4.91	4.82	**37.10**	26.66
InstanT	5.17	5.28	46.06	32.91
FreeMatch + SEAL	**4.8**	**4.71**	37.12	**26**

A.2 More ablation studies

A.2.1 Different threshold

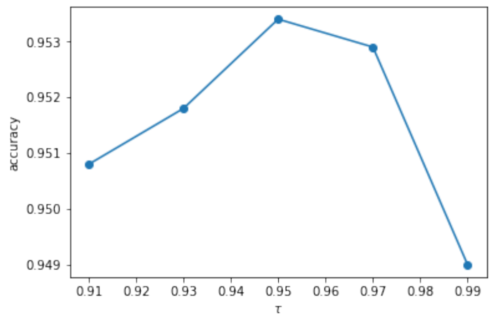


Figure 6: Varying different threshold for pseudo labeling

We first investigate the impact of threshold choice on the accuracy of our method. We experiment with five different values of τ and plot the results in Figure 6. As shown in the figure, the accuracy increases as τ approaches 0.95, and decreases when τ deviates from 0.95. This suggests that choosing a threshold of 0.95 yields the best performance.

A.2.2 Convergence Speedup

Figure 7 displays the Top-1 accuracy of CIFAR100-2500 labels, showcasing that with the aid of SEAL, convergence becomes faster and more stable. With the addition of SEAL, the model consistently outperforms the initial training process in all epochs.

A.2.3 Computation overhead of introducing SEAL

We investigate the additional computation time required when applying SEAL to our method. Table 6 shows the results of these experiments, which were conducted using an Nvidia GeForce RTX 2080 Ti. We observe that the computation of SEAL incurs only a marginal increase in the computation time, demonstrating its efficiency in practice.

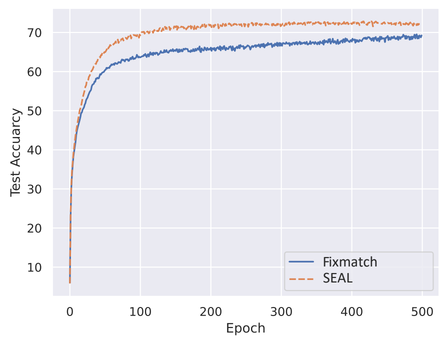


Figure 7: Top-1 accuracy on CIFAR100 dataset (2500 labels)

Table 6: Computation Time of 1 epoch

Update Method	Computation Time (Minutes)
with SEAL	7.11
without SEAL	7.10

A.3 Datasets

We evaluate our proposed method on three popular datasets, namely CIFAR10, CIFAR100, and STL-10.

CIFAR10 and CIFAR100. CIFAR10 (Krizhevsky et al., 2009) contains 60,000 colored images in 10 different classes, where each image has a size of 32×32 pixels. The training set consists of 50,000 labeled images and the test set consists of 10,000 labeled images. Similarly, CIFAR100 (Krizhevsky et al., 2009) contains 100 different classes with the same image size and a similar number of images.

STL-10. STL-10 (Coates et al., 2011) is a semi-supervised benchmark that contains 10 classes. It is adapted from ImageNet(Deng et al., 2009) and contains 500 labeled training samples and 800 labeled testing samples per class. Additionally, it has 10,000 unlabeled images, some of which are not from the labeled classes.

Following the standard semi-supervised learning setting to sample, we sample the labeled images equally and randomly from all classes. To ensure statistical significance, we conduct experiments among seeds and calculate the mean and standard deviation of the results.

A.4 More Theoretical Results

A.4.1 A result about determining the structure of tree based on predictions

We will discuss how one can retrieve the latent structure matrix $\mathbf{A}$ based on the total predictions on total labels $\mathcal{O} \cup \mathcal{L}$.

Theorem 4 *Recall that α is defined by equation (5) and the total prediction operation q is given by definition 1 and $\mathbf{A}_1$ is predefined by a tree's adjacency matrix. Assume there are K samples $\boldsymbol{x}_i$ $(i = 1, \cdots, K)$ that make $[f_\theta(\boldsymbol{x}_1), \cdots, f_\theta(\boldsymbol{x}_K)]$ a full rank matrix. Then given the total prediction matrix $[q(f_\theta(\boldsymbol{x}_1)), \cdots, q(f_\theta(\boldsymbol{x}_K))]$ on these samples, we can recover the matrix $\mathbf{A}$.*

Proof 2 *From definition 1, the model-predicted probabilities are part of the total prediction matrix $\mathbf{Q} = [q(f_\theta(\boldsymbol{x}_1)), \cdots, q(f_\theta(\boldsymbol{x}_K))]$ and we can obtain $\mathbf{F} = [f_\theta(\boldsymbol{x}_1), \cdots, f_\theta(\boldsymbol{x}_K)]$. From equation (5), we have $(\mathbf{I} - \mathbf{A}_1)^{-1}\mathbf{A}_2\mathbf{F} = \mathbf{Q}$, therefore $\mathbf{A}_2 = (\mathbf{I} - \mathbf{A}_1)\mathbf{Q}\mathbf{F}^{-1}$. Thus we can recover the matrix $\mathbf{A}$.*

A.4.2 Optimal Transport

Definition 4 (Wasserstein-1 distance) *Consider two probability distribution: $\boldsymbol{x} \sim \mu$, and $\boldsymbol{y} \sim \nu$. The Wasserstein-1 distance between μ and ν can be defined as:*

$$\mathcal{W}(\mu, \nu) = \min_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{X}} c(\boldsymbol{x}, \boldsymbol{y})\, d\pi,$$

where $\mathcal{X}$ is the space that μ and ν supported on, $c(\cdot,\cdot)$ is a cost function defined on the cartesian space $\mathcal{X}\times\mathcal{X}$, and $\Pi(\mu,\nu)$ is the set of all possible couplings of μ and ν ; and π is a joint distribution satisfying $\int_{\mathcal{X}} \pi(\boldsymbol{x},\boldsymbol{y})\, d\boldsymbol{y} = \mu(\boldsymbol{x})$ and $\int_{\mathcal{X}} \pi(\boldsymbol{x},\boldsymbol{y})\, d\boldsymbol{x} = \nu(\boldsymbol{y})$.

A.4.3 The Relation between SEAL and RTW

Note that we assume no correlation between real labels, thus the vector constructed by $\Pr(r|y)$ as its r-th component is exactly δ_y.

As the nodes in $\mathcal{O}\cup\mathcal{L}$ are exactly all the *total* labels, and those in $\mathcal{O}$ are all the *observed* labels. Comparing equations (6) and (13) yields the result. For convenience, we consider the entire tree metric space as $\mathcal{X}$. Then it is easy to see that $\mathcal{X} = \mathcal{O}\cup\mathcal{L}$, and $\mathcal{X}_{\text{leaf}} = \mathcal{O}$ are the leaf nodes of the tree.

A.4.4 Basic Property of RTW

Here we will present some theoretical properties of relaxed Tree-Wasserstein distance next to illustrate why it is a good metric defined on trees.

Theorem 5 *$W_{d_{\mathcal{X}}}^{relax}(\cdot,\cdot)$ defines a metric on the probability space. Furthermore, when $\mathbf{A}$ is the (hard) adjacency matrix of a tree, the relaxed tree Wasserstein distance is exactly the tree Wasserstein distance.*

Proof 3 *The positive definiteness and symmetry of $W_{d_{\mathcal{X}}}^{relax}$ is clear from its definition. Then we show it satisfies the triangular inequality. For any probability measures λ, μ and ν on $\mathcal{X}_{leaf}$.*

$$\begin{aligned}
& W_{d_{\mathcal{X}}}^{relax}(\lambda,\mu) + W_{d_{\mathcal{X}}}^{relax}(\mu,\nu) \\
&= \sum_{v\in\mathcal{X}} w_v \left| \sum_{x\in\mathcal{X}_{leaf}} \alpha_{vx}\left(\lambda(x)-\mu(x)\right)\right| \\
&+ \sum_{v\in\mathcal{X}} w_v \left| \sum_{x\in\mathcal{X}_{leaf}} \alpha_{vx}\left(\mu(x)-\nu(x)\right)\right| \\
&\geq \sum_{v\in\mathcal{X}} w_v \left| \sum_{x\in\mathcal{X}_{leaf}} \alpha_{vx}\left(\lambda(x)-\nu(x)\right)\right| \\
&= W_{d_{\mathcal{X}}}^{relax}(\lambda,\nu).
\end{aligned} \tag{14}$$

Note that when $\mathcal{X}$ is a hard tree, we have $\mathbb{I}_{v \text{ is the ancestor of } x} = \alpha_{vx}$. Then from equations (12) and (13), we know that RTW is exactly the tree Wasserstein distance in this degenerate case.

Theorem 6 *The relaxed tree Wasserstein distance is a negative definite kernel.*

Proof 4 *Please see Appendix A.4.5.*

A.4.5 Kernel Property of RTW

Definition 5 *(Berg et al., 1984) A function $k : \mathcal{M}\times\mathcal{M} \to \mathbb{R}$ is negative definite if for $\forall n \geq 2$, $\forall x_1, x_2, \ldots, x_n \in \mathcal{M}$ and $\forall c_i \in \mathbb{R}$ such that $\sum_{i=1}^n c_i = 0$, we have $\sum_{i,j} c_i c_j k(x_i, x_j) \leq 0$.*

Table 7: ViN best knn classification

initial feature	smooth feature	SEAL feature	prob space hard treedis
0.6725	0.6818	0.6831	0.8094

Table 8: ResNet18 best knn classification

initial feature	smooth feature	SEAL feature	prob space hard treedis
0.9542	0.9551	0.9572	0.9574

We shall prove that RTW is a negative definite kernel on the tree leaf Wasserstein space $\mathcal{M} = \mathbb{P}(\mathcal{X}_{\text{leaf}})$. We define a mapping Φ where

$$\Phi(x) = (\begin{matrix} (\mathbf{I} - \mathbf{A}_1)^{-1} \mathbf{A}_2 \\ \mathbf{I} \end{matrix})x. \tag{15}$$

Note $k(x_i, x_j) = \|w \circ \Phi(x_i) - w \circ \Phi(x_j)\|_1$. Since the definition of negative definiteness is only related to the value of k, we can transform the problem of considering Φ only. Note k is only a weighted l_1 distance between Φ, from the separability of l_1 norm and (Le et al., 2019)'s Lemma A.2, it is clear that RTW is negative definite.

A.5 A Closer Look at the Supervised Settings

In this section, we shall show the performance of SEAL regularization on two backbones. One is a backbone with fewer parameters, another is the standard ResNet 18 backbone.

Consider a supervised learning objective $L(\theta)$ over a batch, such as Cross-Entropy (CE) to train the neural network θ. One could derive the SEAL regularized objective as

$$\begin{aligned} \mathcal{L}(\theta, \Theta) &= \frac{1}{|B|} \sum_{(\boldsymbol{x},y)\in B} \mathrm{CE}(f_\theta(\boldsymbol{x}), \delta_y)) + \lambda \phi(f_\theta(\boldsymbol{x}), \delta_y)) \\ &= L(\theta) + \lambda \Phi(\theta, \Theta; B) \end{aligned} \tag{16}$$

Optimizing $\mathcal{L}(\theta, \Theta)$ jointly trains the neural network f_θ and the latent hierarchy defined by Θ.

A.5.1 ViN backbone

We train Vision Nystromformer (ViN) (Jeevan & Sethi, 2021) with optimizer AdamW for 60 epochs and get classification accuracy 66.65%, apply the same configuration to label smoothing will give an accuracy of 68.02%. SEAL regularization boosts the accuracy of initial ViN from 66.65% to 68.52% within 5 epochs.

We are also interested in t-SNE visualization of the learned backbone feature, Figure 9a is the initial ViN feature, and Figure 9b ViN with label smoothing $\alpha = 0.1$, Figure 9c is the learned ViN feature with SEAL regularization. Figure 9d is slightly different, and we use the learned relaxed Tree-Wasserstein distance on probability space as the similarity metric.

As for the k-nearest neighbors (kNN) task, we choose the best k for each subtask respectively. We summarize the result in Table 7. Note relaxed Tree-Wasserstein distance is a well-defined metric, so we also calculate the relaxed Tree-Wasserstein distance on probability space to do the knn task. The tree we used is plotted in Fig. 8a. The tree shows some semantic relations between classes, as semantic closer classes have smaller tree distances.

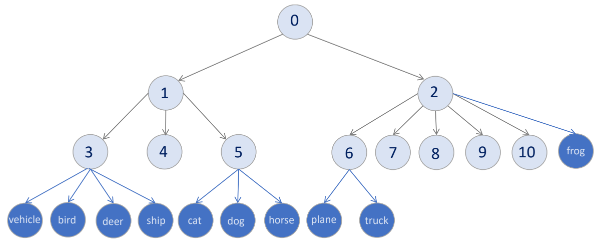
A.5.2 ResNet18 backbone

Inspired by paper NBDT, on CIFAR10 we train 200 epochs, and the origin (trained by cross-entropy loss) accuracy is 95.42%. With label smoothing, the accuracy is 95.54%, while with SEAL regularization, the accuracy is 95.75%. In the above experiments, we train the first 180 epochs using the same loss as the initial and turn the loss to label smoothing or RTW respectively.

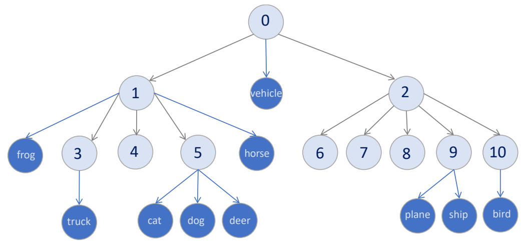
We are also interested in t-SNE visualization of the learned backbone feature, Figure 10a is the initial VIN feature, and Figure 10b VIN with label smoothing $\alpha = 0.1$, Figure 10c is the SEAL boosted VIN feature. Figure 10d is slightly different, and we use the learned tree distance on probability space as the similarity criteria.

As for the k-nearest neighbors (kNN) task, we summarize the result in Table 8. Note relaxed Tree-Wasserstein distance is a well-defined metric, so we also calculate the relaxed Tree-Wasserstein distance on probability space to do the knn task. The tree we used is plotted in Figure 8b. The tree shows some semantic relations between classes, as semantic closer classes have smaller relaxed Tree-Wasserstein distances.

Note that in the original paper NBDT, the initial accuracy is 94.97%, and their method gets 94.82%. Their initial accuracy is slightly lower than ours may be due to the number of epochs they run being smaller.

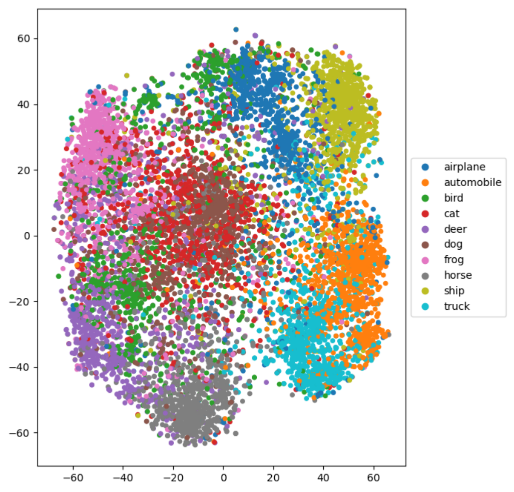


(a) ViN Tree

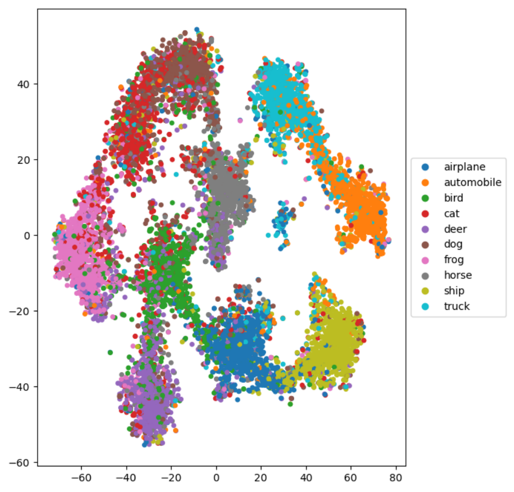


(b) ResNet Tree

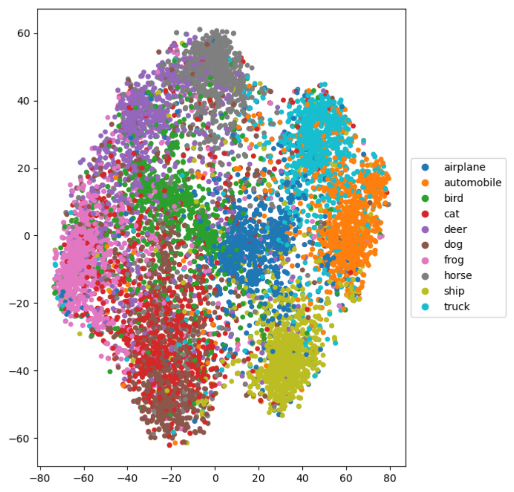
Figure 8: Comparison of ViN Tree and ResNet Tree



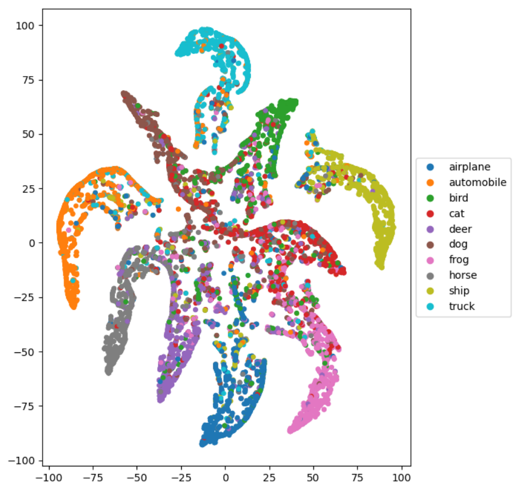
(a) t-SNE visualization of the initial feature of ViN backbone model.



(b) t-SNE visualization of the learned feature of ViN backbone model with label smoothing.

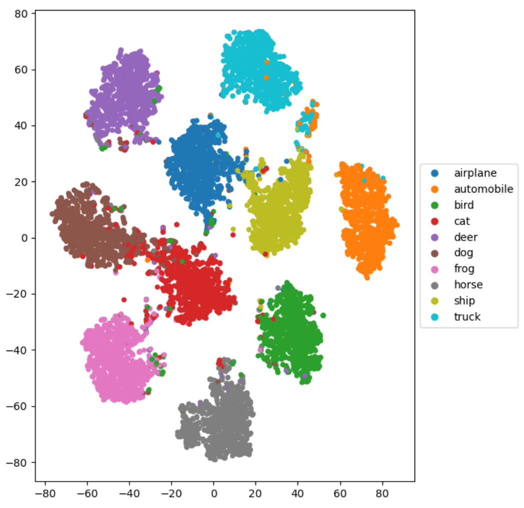


(c) t-SNE visualization of the learned feature of ViN backbone model with SEAL regularization.

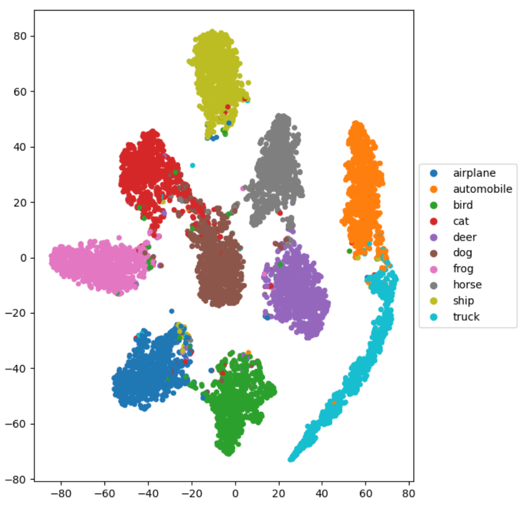


(d) t-SNE visualization of the learned probability of ViN model with SEAL regularization. The similarity metric is defined by relaxed Tree-Wasserstein distance.

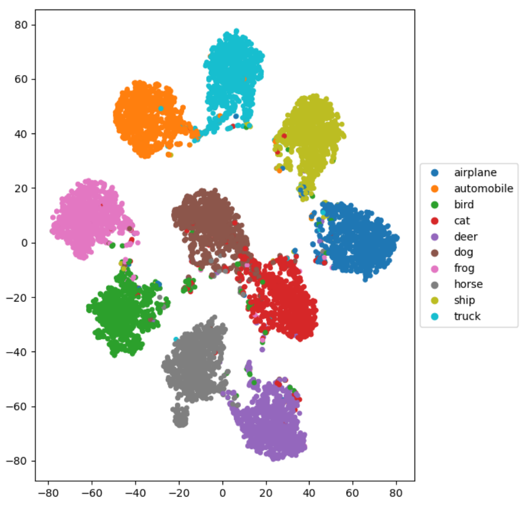
Figure 9: Comparison of t-SNE visualizations of features and probabilities learned by the ViN model with different regularization methods. (a) shows the initial feature of the ViN backbone model. (b) shows the learned feature with label smoothing. (c) shows the learned feature with SEAL regularization. (d) shows the learned probability with SEAL regularization using relaxed Tree-Wasserstein distance as the similarity metric.



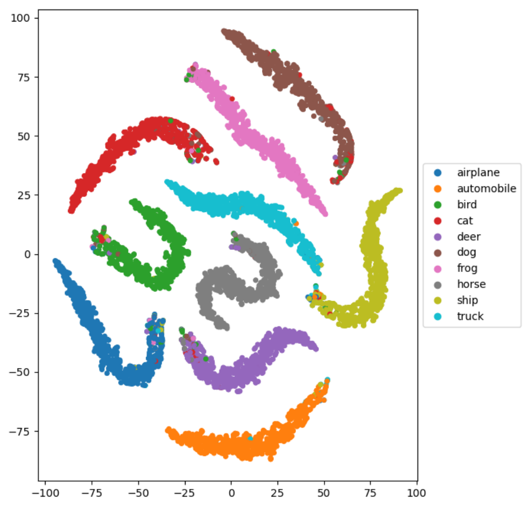
(a) t-SNE visualization of the initial feature of ResNet18 backbone model



(b) t-SNE visualization of the learned feature of ResNet18 backbone model with label smoothing



(c) t-SNE visualization of the learned feature of ResNet18 backbone model with SEAL regularization



(d) t-SNE visualization of the learned probability of ResNet18 backbone model with SEAL regularization. Similarity metric is defined by relaxed Tree-Wasserstein distance.

Figure 10: Visualizations of ResNet18 backbone model features with different regularization methods.