Not All Classes Stand on Same Embeddings: Calibrating a Semantic Distance with Metric Tensor

Jae Hyeon Park[1] Gyoomin Lee[1] Seunggi Park[1] Sung In Cho[1]*
[1]Division of AI Software Convergence at Dongguk University
30, Pildong-ro 1-gil, Jung-gu, Seoul, 04620, Republic of Korea
{pjh0011,exceleaf,psg0912,csi2267}@dongguk.edu

Abstract

The consistency training (CT)-based semi-supervised learning (SSL) bites state-of-the-art performance on SSL-based image classification. However, the existing CT-based SSL methods do not highlight the non-Euclidean characteristics and class-wise varieties of embedding spaces in an SSL model, thus they cannot fully utilize the effectiveness of CT. Thus, we propose a metric tensor-based consistency regularization, exploiting the class-variant geometrical structure of embeddings on the high-dimensional feature space. The proposed method not only minimizes the prediction discrepancy between different views of a given image but also estimates the intrinsic geometric curvature of embedding spaces by employing the global and local metric tensors. The global metric tensor is used to globally estimate the class-invariant embeddings from the whole data distribution while the local metric tensor is exploited to estimate the class-variant embeddings of each cluster. The two metric tensors are optimized by the consistency regularization based on the weak and strong augmentation strategy. The proposed method provides the highest classification accuracy on average compared to the existing state-of-the-art SSL methods on conventional datasets.

1. Introduction

Most semi-supervised learning (SSL) methods [2, 7, 8, 13–15, 20, 30, 38] on image classification make extensive use of the equality of multiple views of the input. Specifically, they encourage the consistency of the model's prediction vectors for two different versions of the input image derived from the same input image. FixMatch [30] which utilizes pseudo labeling on weakly and strongly augmented images is the most famous consistency training technique and it suggests the research direction of numerous SSL studies based on consistency training [37]. Furthermore,

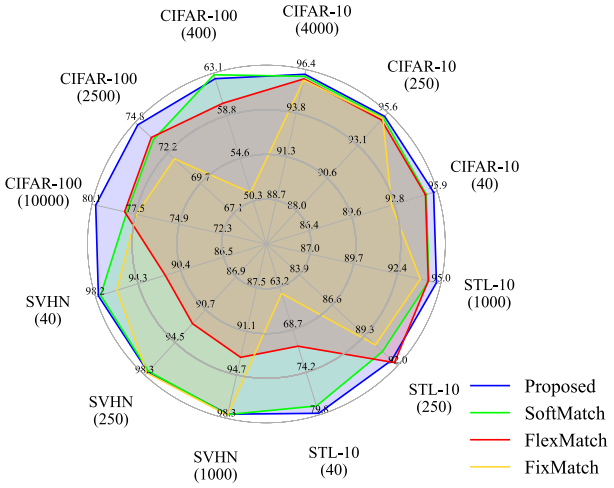

Figure 1. The performance of benchmarks on a broad range of conventional datasets compared with existing SSL. The numerical values under the dataset name (CIFAR-10, CIFAR-100, SVHN, STL-10) with bracket represent the scenarios of the SSL.

this multi-view strategy is used for contrastive learning and contributes to greatly improving classification accuracy. In contrastive learning, positive and negative pairs are made from the input anchor image and its augmented versions, the input anchor image and other images in a batch, respectively. SimCLR [7, 8] achieved excellent classification performance by initializing the backbone network with a pretext stage that learns the similarity or dissimilarity relationship in positive or negative pairs.

Most of these SSL techniques strongly adopt the manifold hypothesis [18, 26] that assumes the original high-dimensional data in the real world can be expressed on the relatively low-dimensional latent manifold (embedding space). In this manifold hypothesis, it is believed that the overall structure of the manifold (non-Euclidean space) can be found by clustering the positive pairs in a local region

*Corresponding author.

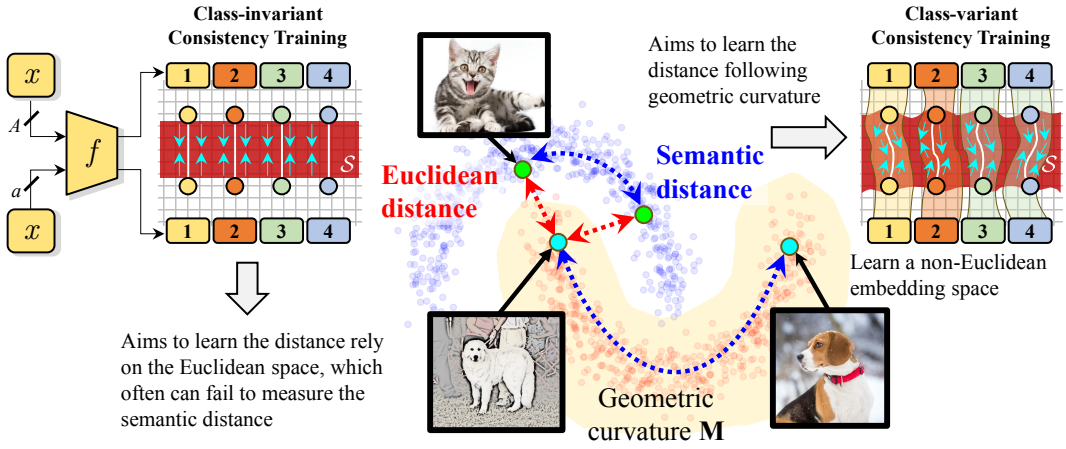

Figure 2. Conceptual illustrations of class invariant and class-variant consistency training on embedding spaces.

(analogous Euclidean space). However, this consistency training still does not fully utilize both global and local geometric information of the manifold space. We highlight the importance of the distance measurement considering the geometrical curvature of the embedding space. In addition, to maximize the effectiveness of consistency training, the distance measurement must be adaptively applied to each class. To deal with this, we proposed a novel consistency regularization based on the metric tensor representing an intrinsic geometry on embedding spaces. The proposed method minimizes the prediction discrepancy between different views of positive pairs based on the estimated geometric structure. The proposed method aims to find the geometric structure of an embedding space based on two metric tensors: global (overall distribution) and local (class-wise) metric tensors. The global metric tensor introduces the metric function to estimate the geometric structure of embedding space from the whole data distribution. In addition, the local metric tensor, representing the class-wise geometric structure, is estimated with a pseudo-labeling technique. By estimating the geometric structure of the manifold, the proposed method can estimate the semantic distance in the manifold space more accurately and provide the highest classification accuracy for the conventional datasets as shown in Fig. 1. The main contributions of the proposed method are summarized as follows:

- **Estimation of intrinsic geometric structure in the embedding space via metric tensor:** We represent the intrinsic geometric structure of embedding space as the global and local metric tensors. The two metric tensors are employed for consistency regularization to precisely measure the discrepancy of the multi-view in the embedding space.
- **Novel consistency regularization with global and local metric tensors:** We propose a new consistency regularization technique for SSL that estimates the metric tensors in a high-dimensional embedding space and simultaneously induces consistent prediction of multi-view of the unlabeled data.

2. Preliminary

Consistency training leads to the consistent prediction of different views of the same input image. Early consistency training methods [18, 21, 34, 37] generate different views through stochastic perturbation such as dropout, Gaussian noise, and random cropping. Virtual adversarial training (VAT) [21] employs a novel data augmentation technique using an adversarial noise that is aggressive to the model optimization as the perturbation. In FixMatch [30], the discrepancy of predictions for weakly and strongly augmented inputs is minimized based on pseudo-labeling. Furthermore, in recent consistency training-based pseudo labeling, adaptive thresholding depending on the learning status of the model is being studied actively [6, 16, 34, 37, 41].

Metric tensor is a matrix representing the intrinsic geometry on a multi-dimensional manifold [35]. It addresses the curvature of space for difference measurement in the non-Euclidean space. Specifically, the shortest distance between any two vectors in a two-dimensional Euclidean space can be expressed as follows:

$$\begin{aligned} s^2 &= (p_1 - q_1)^2 + (p_2 - q_2)^2, \\ &= (\mathbf{p} - \mathbf{q})(\mathbf{p} - \mathbf{q})^\top, \\ &= \mathbf{u}\mathbf{u}^\top, \end{aligned} \tag{1}$$

where $\mathbf{p} = (p_1, p_2)$ and $\mathbf{q} = (q_1, q_2)$ are two-dimensional vectors. $\mathbf{u} = \mathbf{p} - \mathbf{q}$ is a difference vector. Eq. 1 can be generalized in d-dimensional manifold space as follows:

$$s^2 = \mathbf{u}\mathbf{M}\mathbf{u}^\top, \tag{2}$$

where $\mathbf{u} \in \mathbb{R}^{1\times d}$ is a difference vector between two vectors $\{\mathbf{p}, \mathbf{q}\} \in \mathbb{R}^{1\times d}$. $\mathbf{M} \in \mathbb{R}^{d\times d}$ is the metric tensor. $\mathbf{M}$ is

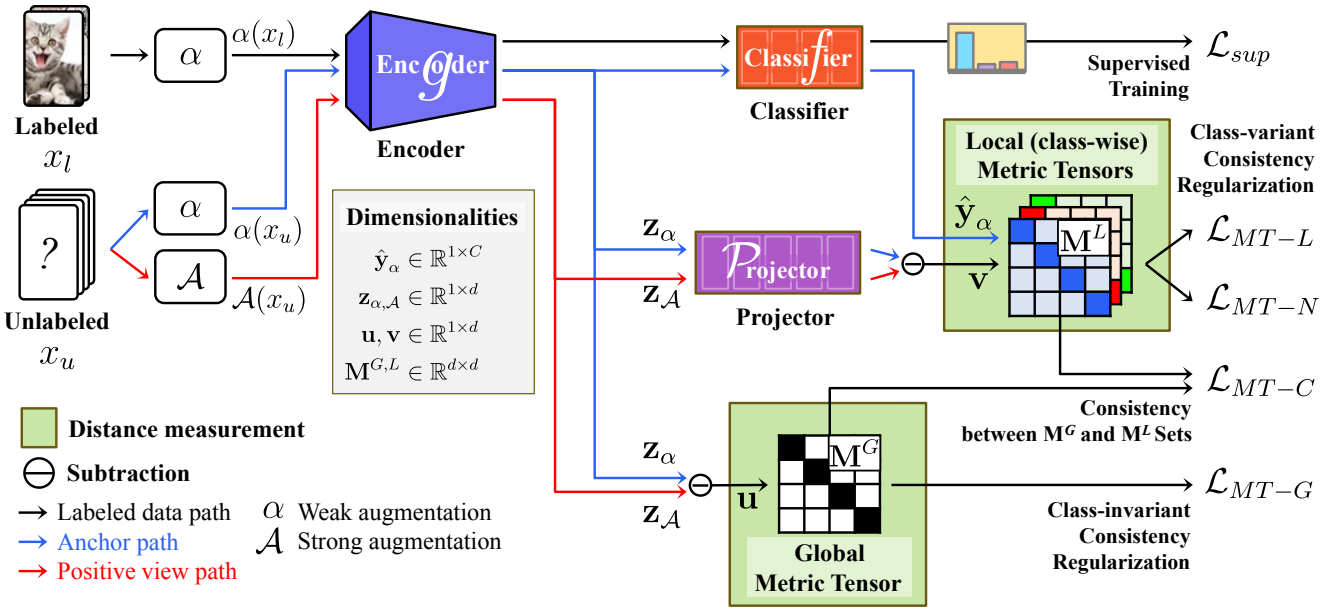

Figure 3. Overall architecture of the proposed metric-tensor-based consistency regularization.

the identity matrix (**I**) in flatten Euclidean space. Therefore, Eq. 1 is the unique case that can be derived when the metric tensor **M** is **I**. Here, we note that the shortest distance between two points in a high-dimensional manifold can be derived through linear transformation by the metric tensor **M**. By estimating the metric tensor of the manifold space, we further boost the existing consistency training effect.

3. Proposed Method

Motivations. Existing consistency regularization assumes that the semantic distance of neighbors (same class or different views) can be approximated by the Euclidean distance in the local region. However, in practice, most embedding spaces used in the SSL do not follow the Euclidean space properties. Thus, the distance calculated by following the surface of the embedding space can represent a more accurate semantic distance. The proposed method induces the attractive relation between the manifold hypothesis and consistency regularization as a form of the metric tensor. In addition, we attached a new trial to estimate class-wise metric tensors. Figure 2 depicts the motivation of the proposed method using four example classes. Here, in the class-invariant consistency training, distances between different views (circles with the same color) are regarded as the same in the embedding space (gray grid) even if they are different classes. However, as shown in class-variant consistency training, the distances between different views can be estimated differently due to the different geometric structures on embedding space (color region) depending on the class.

Problem settings. We build a convolutional encoder g and two linear modules (classifier f and projector $\mathcal{P}$). In addition, we define the two metric tensor sets, the global metric tensor $\mathbf{M}^G \in \mathbb{R}^{d\times d}$ and local metric tensor set $\mathbf{M}^L$. $\mathbf{M}^L$ is a set of metric tensors $\mathbf{M}^L_c \in \mathbb{R}^{d\times d}$ for a class index c. d is the number of feature dimensions.

The class predictions of the labeled and unlabeled samples are extracted as follows:

$$\hat{\mathbf{y}} = f(g(x)), \tag{3}$$

where the x represents an input image. The classification model $f \circ g$ maps x into a C-dimensional prediction vector, $\hat{\mathbf{y}} \in \mathbb{R}^{1\times C}$. x can be samples among the weakly augmented labeled samples $\alpha(x_l)$, weakly augmented unlabeled samples $a(x_u)$, and strongly augmented unlabeled samples $\mathcal{A}(x_u)$. a and A represent the weak and strong augmentations, respectively. To be more specific, the predictions of the $\alpha(x_l)$, $\alpha(x_u)$, and $\mathcal{A}(x_u)$ are denoted as $\hat{\mathbf{y}}_l$, $\hat{\mathbf{y}}_\alpha$, and $\hat{\mathbf{y}}_\mathcal{A}$, respectively. Note that, all the training images are regarded as the unlabeled data (x_u), i.e., consistency regularization will be applied for all the training labeled and unlabeled images.

Overview of the proposed method. Figure 3 shows the overall architecture of the proposed method. First, we generate the differently augmented views $(\mathcal{A}(x_u), \alpha(x_u))$ of x_u. The difference vector $(\mathbf{u})$ between the $\mathcal{A}(x_u)$ and $\alpha(x_u)$ on the embedding space is then used for the optimization of $\mathbf{M}^G$ (the dashed green arrows). In addition, the embedding vectors (outputs by g) are projected to a new feature space via the $\mathcal{P}$ module. The difference vector $(\mathbf{v})$ between projected embeddings of $\mathcal{A}(x_u)$ and $\alpha(x_u)$ is extracted. Then, class-wise $\mathbf{M}^L_c$ is optimized based on the

pseudo labeling using $\hat{\mathbf{y}}_\alpha$. Finally, consistency loss is applied for parallelism of the optimization trajectories of $\mathbf{M}^G$ and $\mathbf{M}_c^L$.

3.1. Supervised Learning on the Labeled Set

The supervised learning with the labeled data is conducted as follows:

$$\mathcal{L}_{sup} = \frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C} -\mathbf{y}^{LS}(i,c)\log\hat{\mathbf{y}}_l(i,c), \tag{4}$$

where N and C are the numbers of labeled samples and classes, respectively. $\mathbf{y}^{LS}$ is a label smoothed supervision of a one-hot-encoded ground-truth vector $\mathbf{y}$. This supervised learning plays an important role as an indicator that maps the unlabeled data into its latent class. Thus, we applied label smoothing [22] to alleviate the overfitting and overconfident problems, that often occur in supervised learning with a small amount of labeled data, as follows:

$$\mathbf{y}^{LS}(c) = \mathbf{y}(c)(1-\alpha) + \alpha/C, \tag{5}$$

where $\mathbf{y}(c)$ is the ground-truth value of the c-th class. $\mathbf{y}^{LS}$ depicts the smoothed ground-truth with smoothing factor α. The parameters are introduced in the implementation details section.

With supervised learning on the labeled data, we will apply the metric tensor-based consistency training in this work. The details of the proposed method will be described in the following subsections.

3.2. Global Metric Tensor

Global metric tensor, $\mathbf{M}^G$, is introduced to estimate the geometric structure of the manifold space from the entire distribution of embedding vectors extracted through g. That is, we estimate the class-invariant embedding space that shares a common geometry structure for all classes. We utilize the consistency regularization technique to estimate $\mathbf{M}^G$ in high-dimensional feature space. Specifically, weak and strong augmentations are applied on unlabeled samples to generate different views of those samples as follows:

$$\begin{aligned} \mathbf{z}_a &= g(\alpha(x_u)),\\ \mathbf{z}_A &= g(\mathcal{A}(x_u)),\\ \mathbf{u} &= \mathbf{z}_\alpha - \mathbf{z}_\mathcal{A}, \end{aligned} \tag{6}$$

where $\{\mathbf{z}_\alpha, \mathbf{z}_\mathcal{A}\} \in \mathbb{R}^{1\times d}$ represents the d-dimensional embedding vectors. Then, the loss function of the trainable matrix module $\mathbf{M}^G$ for the difference vector ($\mathbf{u}$) of two views is calculated as follows:

$$\mathcal{L}_{MT-G} = \frac{1}{\mu B}\sum_{i=1}^{\mu B}\mathbf{u}_i\mathbf{M}^G\mathbf{u}_i^\top, \tag{7}$$

where μB is the batch size of the unlabeled samples. $\mathcal{L}_{MT-G}$ depicts the shortest distance between two different views on the embedding space. $\mathbf{M}^G$ is then optimized by the Eq. 7. This allows the consistency regularization between weakly and strongly augmented views in addition to the optimization of the $\mathbf{M}^G$. By this, we incorporate the intrinsic geometry of class-invariant embedding space by the $\mathbf{M}^G$ during consistency regularization and this can provide the precise distance on the calibrated embedding space.

3.3. Local Metric Tensor

We mainly deal with class-variant embedding estimation in this work. Therefore, we access the adaptive consistency regularization that can reflect the geometric structure of different clusters via the class-wise metric tensor. The loss function for estimating $\mathbf{M}_c^L$ is derived as follows:

$$\begin{aligned} \mathbf{v} &= \mathcal{P}(\mathbf{z}_\alpha) - \mathcal{P}(\mathbf{z}_\mathcal{A}),\\ \hat{\mathbf{v}}_1 &= \mathbf{v}_1\mathbf{M}_1^L\mathbf{v}_1^\top,\\ \hat{\mathbf{v}}_2 &= \mathbf{v}_2\mathbf{M}_2^L\mathbf{v}_2^\top,\\ &\vdots\\ \hat{\mathbf{v}}_C &= \mathbf{v}_C\mathbf{M}_C^L\mathbf{v}_C^\top,\\ \mathcal{L}_{MT-L} &= \frac{1}{C\times\mu B_c}\sum_{c=1}^{C}\sum_{i=1}^{\mu B_c}\hat{\mathbf{v}}_c(i), \end{aligned} \tag{8}$$

where $\mathcal{P}$ is the projection module consisting of three linear layers. Projector $\mathcal{P}$ allows the model to capture a more precise representation of the unlabeled data than global geometry for structuring the class-wise $\mathbf{M}_c^L$. $\mathbf{v} \in \mathbb{R}^{n\times d}$ represents the difference between elaborated embeddings via the projection module $\mathcal{P}$. $\mathbf{M}^L_{c=[1,2,\dots,C]}$ and $\mathbf{v}_{c=[1,2,\dots,C]}$ are the metric tensor and difference vectors of the c-th class, respectively. $\hat{\mathbf{v}}_{c=[1,2,\dots,C]}$ represents the class-wise distance value between the two projected embeddings. μB_c is the number of samples predicted to the c-th class and not specified. To obtain the class prediction for the unlabeled samples, we utilize the output $\hat{\mathbf{y}}_\alpha$ of the classification network for the unlabeled data as follows:

$$\begin{aligned} \hat{\mathbf{y}}_\alpha^{max} &= \underset{c^*}{\operatorname{argmax}}\,\hat{\mathbf{y}}_\alpha(c),\\ \mathbf{v}_c &= \mathbb{I}_{[\hat{\mathbf{y}}_\alpha^{max}=c]}\odot\mathbf{v}, \end{aligned} \tag{9}$$

where $\hat{\mathbf{y}}_\alpha^{max}$ represents the pseudo class having the largest probability value from $\hat{\mathbf{y}}_\alpha$. $\mathbb{I}_{[\cdot]}$ is the indicator function, $\odot$ means element-wise multiplication. Thus, $\mathbf{v}_c$ is the class-wise difference between the predictions of weakly augmented unlabeled samples having the class c. With the pseudo labeling, the local metric loss ($\mathcal{L}_{MT-L}$) in Eq. 7 is used to minimize the distance between two projected data depending on the intrinsic geometric characteristics of the manifold of each class. Therefore, by applying $\mathcal{L}_{MT-G}$,

we can apply consistency regularization between projected points, and also find the optimal metric tensor for each class.

3.4. Consistency between Global and Local Metric Tensors

The proposed method aims to derive class-variant embeddings by the local metric tensor while maintaining the global geometric structure in manifold space by the global metric tensor. Therefore, to prevent the two metric tensors from being optimized in different directions, consistency loss is applied as follows:

$$\mathcal{L}_{MT-C} = \frac{1}{Cd^2} \sum_{c=1}^{C} \sum_{i=1}^{d^2} \left(\mathbf{M}^G(i) - \mathbf{M}_c^L(i) \right)^2, \tag{10}$$

where d is the number of feature dimensions, thus, d^2 is the number of elements of one metric tensor. Consistency loss is calculated as mean squared errors (MSE) of $\mathbf{M}^G$ and $\mathbf{M}_c^L$.

3.5. Informative Local Metric Tensor

As shown in Sec. 3.3, to establish the discriminative class-wise metric tensor, we made the local metric tensor, $\mathbf{M}_c^L$ depending on the class. When the local metric tensor $\mathbf{M}_c^L$ is estimated, the importance of geometric information of each axis in the original manifold space should be considered because the curvature information of the axis itself generally has a greater impact on distance measurement than the curvature related to the relationship between any two axes. In other words, the diagonal elements in the metric tensor include the important properties of the embedding space since they indicate whether the space lies on the Euclidean or not. Thus, by the following loss function, we regularize the $\mathbf{M}_c^L$ so that the proposed model can consider the curvature information of each axis more importantly, rather than the relationship between different axes:

$$\mathcal{L}_{MT-N} = -\frac{1}{C} \sum_{c=1}^{C} \text{tr}(\mathbf{M}_c^L). \tag{11}$$

By this, each axis of embeddings is emphasized to prevent excessive axis adjustment by the relationship between different axes. When Eqs. 10 and 11 are utilized together, synergistically contribute to the enhancement of SSL performance (provided in Table 3 of Subsection 4.3).

3.6. Optimization

With the supervised training on the labeled data, the cost of the consistency regularization for the unlabeled data is conducted as follows:

$$\begin{aligned} \mathcal{L}_{total} = \mathcal{L}_{sup} + \lambda_G \mathcal{L}_{MT-G} + \lambda_N \mathcal{L}_{MT-L} \\ + \lambda_L \mathcal{L}_{MT-C} + \lambda_L \mathcal{L}_{MT-N}, \end{aligned} \tag{12}$$

where $\mathcal{L}_{total}$ is the total training loss on a batch. λ_G, λ_L, λ_C, and λ_N represent the constant parameters that manage the strength of $\mathcal{L}_{MT-G}$, $\mathcal{L}_{MT-L}$, $\mathcal{L}_{MT-C}$, and $\mathcal{L}_{MT-N}$, respectively. The values of these hyper-parameters will be described in Subsection. 3.

The proposed method estimates the intrinsic geometrical diversity of each class of the embedding space through a class-wise metric tensor and derives the consistency training of unlabeled data based on this. By this, the proposed method can provide more accurate and consistent predictions for different views.

3.7. Exponential Moving Average

We build an ensemble model according to the training steps by applying the exponential moving average (EMA) techniques [13, 14, 31], which has recently been frequently adopted as a baseline for inference stages. The EMA is an ensemble model that applies a relatively higher weight to the model parameters of the previous step than the parameters of the model currently being trained as follows:

$$\theta_{t+1} \leftarrow m\theta_{t-1} + (1-m)\theta_t, \tag{13}$$

where t is the training step. θ represents the trainable parameters in the model. m is the momentum factor to scale the graduality of ensembling. The trainable parameters of the model are exponentially ensembled at every training step by Eq. 13. We only used the EMA model for the inference stage.

4. Experiments

4.1. Experimental Setup

Datasets. We evaluated the SSL performance of the proposed method on the four conventional datasets: CIFAR-10, CIFAR-100 [17], SVHN [24], and STL-10 [9]. We chose the three SSL scenarios depending on the amounts of the available labeled data. All the SSL scenarios are based on a unified semi-supervised learning benchmark (USB) [33] which is an open-source SSL library implementing the 14 algorithms and comprehensive modules.

Benchmark methods. We used the Pseudo-Labeling [19], mean teacher (MT) [31], VAT [21], ReMixMatch [3], FixMatch [30], UDA [36], DASH [37], FlexMatch [41], SoftMatch [6], ConMatch [16], SimMatch [43], and ShrinkMatch [39] as the benchmark methods. We carefully addressed the performance of the SSL methods based on the mean and standardization values of classification error rates by repeating the experiments three times.

Implementation details. We implemented the proposed method in Pytorch [28]. Stochastic gradient descent (SGD) [4] was used as an optimizer. To establish the encoder g, we used the trunk of the Wide-ResNet [40], a wide backbone

Dataset / Method	CIFAR-10 40	CIFAR-10 250	CIFAR-10 4,000	CIFAR-10 Avg.	CIFAR-100 400	CIFAR-100 2,500	CIFAR-100 10,000	CIFAR-100 Avg.
Fully-supervised		*4.91±0.33		-		*19.27±0.05	-	
Pseudo-Labeling [19]	74.61±0.26	46.49±2.20	15.08±0.19	45.39±0.88	87.45±0.85	57.74±0.28	36.55±0.24	60.58±0.45
Mean Teacher [31]	70.09±1.60	37.46±3.30	8.10±0.21	38.55±1.70	81.11±1.44	45.17±1.06	31.75±0.23	52.67±0.91
VAT [21]	74.66±2.12	41.03±1.79	10.51±0.12	42.07±1.34	85.20±1.40	46.84±0.79	32.14±0.19	54.73±0.79
UDA [36]	10.62±3.75	5.15±0.06	4.29±0.07	6.69±1.29	46.39±1.59	27.73±0.21	22.49±0.23	32.20±0.67
DASH [37]	9.16±4.31	**4.56±0.13**	4.08±0.06	5.93±1.50	44.76±0.96	27.18±0.21	21.97±0.14	31.30±0.43
FixMatch [30]	7.47±0.28	*5.07±0.65	*4.26±0.05	5.60±0.33	*48.85±1.75	*28.29±0.11	*22.60±0.12	33.25±0.66
ReMixMatch [3]	9.88±1.03	6.30±0.05	4.84±0.01	7.01±0.36	42.75±1.05	26.03±0.35	**20.02±0.27**	29.60±0.56
FlexMatch [41]	4.97±0.06	4.98±0.09	4.19±0.01	4.71±0.05	39.94±1.62	26.49±0.20	21.90±0.15	29.44±0.65
SimMatch [43]	5.60±1.37	4.84±0.39	3.96±0.01	4.80±0.59	37.81±2.21	**25.07±0.32**	20.58±0.11	**27.82±0.87**
ConMatch [16]	**4.43±0.13**	**4.70±0.25**	**3.92±0.08**	**4.34±0.15**	38.89±2.18	25.39±0.20	-	-
SoftMatch [6]	4.91±0.12	4.82±0.09	4.04±0.02	4.59±0.07	37.10±0.77	26.66±0.25	22.03±0.03	28.60±0.35
ShrinkMatch [39]	5.08±0.18	4.74±0.25	-	-	**35.36±1.04**	**25.17±0.20**	-	-
Proposed	**4.37±0.21**	4.71±0.06	**3.92±0.09**	**4.33±0.12**	**37.49±1.92**	25.43±0.07	**20.18±0.25**	**27.70±0.74**

Table 1. Top-1 classification error rates (%). The smaller value indicates the more accurate. Avg. is the average value of the error rates depending on the method.

Dataset / Method	SVHN 40	SVHN 250	SVHN 1,000	SVHN Avg.	STL-10 40	STL-10 250	STL-10 1,000	STL-10 Avg.
Fully-supervised		*2.13±0.01		-		None		-
Pseudo-Labeling [19]	64.61±5.60	15.59±0.95	9.40±0.32	29.87±2.29	74.68±0.99	55.45±2.43	32.64±0.71	54.26±1.37
Mean Teacher [31]	36.09±3.98	3.45±0.03	3.27±0.05	14.27±1.35	71.72±1.45	56.49±2.75	33.90±1.37	54.04±1.85
VAT [21]	74.75±3.38	4.33±0.12	4.11±0.20	27.73±1.23	74.74±0.38	56.42±1.97	37.95±1.12	56.37±1.15
UDA [36]	5.12±4.27	**1.92±0.05**	**1.89±0.01**	2.98±1.44	37.42±8.44	9.72±1.15	6.64±0.17	17.93±3.25
DASH [37]	3.03±1.59	2.17±0.10	2.03±0.06	2.41±0.58	-	-	**3.96±0.25**	-
FixMatch [30]	*3.96±2.17	*2.48±0.38	*2.28±0.11	2.91±0.89	35.97±4.14	9.81±1.04	6.25±0.33	17.34±1.83
ReMixMatch [3]	24.04±9.13	6.36±0.22	5.16±0.31	11.85±3.22	32.12±6.24	12.49±1.28	6.74±0.14	17.12±2.55
FlexMatch [41]	8.19±3.20	6.59±2.29	6.72±0.30	7.17±1.93	29.15±4.16	**8.23±0.39**	5.77±0.18	14.38±1.57
ConMatch [16]	3.14±0.57	3.13±0.72	-	-	-	-	**5.26±0.04**	-
SoftMatch [6]	**2.33±0.25**	*2.21±0.00	**2.01±0.01**	**2.18±0.08**	21.42±3.48	*9.22±0.01	5.73±0.24	**12.12±1.24**
ShrinkMatch [39]	2.51±0.56	**1.96±0.04**	-	-	**14.02±0.41**	8.45±0.62	-	-
Proposed	**2.13±0.26**	2.14±0.06	2.04±0.04	**2.10±0.12**	**20.47±5.44**	**8.45±0.30**	5.27±0.11	**11.39±1.95**

Table 2. Top-1 classification error rates (%). The smaller value indicates the more accurate. Avg. is the average value of the error rates depending on the method.

for the SSL benchmarks. Specifically, we used a Wide-ResNet-24-w2 as the encoder for CIFAR-10, SVHN, and STL-10 datasets. Wide-ResNet-24-w8, a deeper channel-depth version of Wide-ResNet-24-w2, was used for the CIFAR-100 experiments. The metric tensors were invented by the trainable array. We used RandAug [10] to generate the strongly augmented version of the unlabeled data. The smoothing factor (α) in Eq. 5 was set to 0.1. All the extensive experiments were conducted in the NVIDIA GeForce RTX 3090 GPU without the distributed learning on the multiple GPUs. The details of network architecture and hyper-parameters were described in Supplementary.

4.2. Comparison Results on Semi-supervised Benchmarks

Results on the classification performance. Tables 1 and 2 showed the average of the classification error rates depending on the dataset. The numerical numbers in the second row of the table represent the number of labeled data used for training. The asterisk symbol ($*$) in the table indicated that we extracted results by ourselves using the released code based on the well-crafted SSL libraries [41]. We provided the five iterated experimental results on the same scenario under the different random seeds. The bold letter in red color indicates the highest classification performance and the bold letter is the secondary highest classification performance. As shown in this table, the proposed method achieved the lowest classification error rates compared to the benchmark methods. Furthermore, in the CIFAR-10

Losses				CIFAR-10			CIFAR-100			SVHN			STL-10		
$\mathcal{L}_{MT-G}$	$\mathcal{L}_{MT-L}$	$\mathcal{L}_{MT-C}$	$\mathcal{L}_{MT-N}$	40	250	4,000	400	2,500	10,000	40	250	1,000	400	250	1,000
✓				7.87	7.11	6.32	51.44	40.19	38.36	3.28	2.75	2.31	28.10	11.85	10.99
✓	✓			5.19	5.02	4.72	42.81	34.27	28.41	2.67	2.63	2.17	23.61	9.89	6.02
✓	✓	✓		4.88	4.96	4.25	40.54	29.38	24.33	2.38	2.47	2.17	22.63	9.00	5.97
✓	✓		✓	**4.34**	**4.85**	**4.01**	**39.10**	**24.99**	**21.44**	**2.33**	**2.29**	**2.11**	**21.69**	**8.74**	**5.88**
✓	✓	✓	✓	**4.37**	**4.71**	**3.92**	**37.49**	**25.43**	**20.18**	**2.13**	**2.14**	**2.04**	**20.47**	**8.45**	**5.27**

Table 3. Top-1 classification error rates (%) on the benchmark datasets.

Hyper-parameters				CIFAR-10			CIFAR-100			SVHN			STL-10		
λ_G	λ_L	λ_C	λ_N	40	250	4,000	40	250	10,000	40	250	1,000	40	250	1,000
1.0	1.0	1.0	1.0	4.68	4.82	4.06	40.19	25.87	21.05	2.38	3.11	2.21	22.35	10.85	5.43
1.0	1.0	0.1	0.1	4.49	4.79	**3.98**	38.25	**25.67**	20.40	2.29	2.38	**2.07**	23.90	9.03	5.33
1.0	1.0	0.3	0.1	4.45	**4.80**	4.02	37.66	25.71	**20.29**	**2.11**	2.16	2.09	21.85	**8.72**	5.37
1.0	0.7	0.3	0.1	4.51	4.84	4.08	37.61	25.73	20.33	**2.12**	**2.14**	2.08	**17.87**	8.76	**5.22**
0.7	0.7	0.1	0.1	5.00	4.93	4.12	**37.26**	26.24	20.67	2.24	2.25	2.19	21.53	9.05	5.69
0.8	0.7	0.1	0.3	**4.80**	4.96	4.43	38.74	26.17	20.86	2.15	2.19	2.09	23.11	8.92	5.81
1.0	0.7	0.3	0.3	**4.37**	**4.71**	**3.92**	**37.49**	**25.43**	**20.18**	2.13	**2.14**	**2.04**	**20.47**	**8.45**	**5.27**
Average				4.61	4.84	4.13	38.17	25.83	20.54	2.20	2.34	2.11	21.01	9.11	5.45
Standard Deviation				0.21	0.08	0.19	0.95	0.27	0.30	0.10	0.32	0.06	1.54	0.74	0.20

Table 4. Top-1 classification error rates (%) depending on the coefficient values of losses.

experiment with the 40 labeled data, the proposed method achieved a lower error rate of up to 1.4% compared to that of previous state-of-the-art SSL methods.

4.3. Analysis and Ablation

Contribution of the investigated component. Table 3 provided the extensive ablation experiments to provide the effectiveness of the modules invested in the proposed method. $\mathcal{L}_{MT-G}$, $\mathcal{L}_{MT-L}$, $\mathcal{L}_{MT-C}$, and $\mathcal{L}_{MT-N}$ indicate the losses in Eqs. 7, 8, 10 and 11, respectively. As shown in this table, we observed that all the investigations collaboratively improved the classification performance. In particular, when the $\mathbf{M}^G$ is only used ($\mathcal{L}_{MT-G}$), it provided poor classification performance. However, the classification performance is significantly improved when the $\mathbf{M}_c^L$ is used to training together ($\mathcal{L}_{MT-G} + \mathcal{L}_{MT-L}$). Thus, consistency regularization across the class-variant embeddings can activate the precise distance measurement.

Variations depending on the loss weighting. Table 4 showed the classification results of extensive experiments depending on the coefficients of losses in Eq. 12. In addition, this table explains the decision of hyper-parameter setting in the proposed method. We started the experiments with the same values (1.0) for all losses. We seek the optimal setting to provide the best classification performance of the proposed method. As shown in the table, no single parameter setting gives the highest performance for all datasets. That is, the parameter dependency of the proposed method is low thus adaptation for other tasks can be convenient. Consequently, the λ_G, λ_L, λ_C, and λ_N were set to 1.0, 0.7, 0.3, and 0.3, respectively. The decision of hyper-parameters can be further developed by the hyper-parameter optimization techniques [1, 11].

Dataset		Class-(c) 0	1	2	3	4	5	6	7	8	9	Avg.
CIFAR-10 (40)	w/o $\mathbf{M}_c^L$	6.72	10.27	8.37	4.97	2.13	5.32	2.47	3.68	7.19	7.23	5.84
	w/ $\mathbf{M}_c^L$	**8.94**	9.31	**9.65**	4.72	**2.85**	**6.07**	**3.39**	**3.76**	6.45	6.29	**6.14**
SVHN (40)	w/o $\mathbf{M}_c^L$	8.43	9.76	12.65	16.90	7.33	9.47	11.77	6.61	12.28	3.85	9.91
	w/ $\mathbf{M}_c^L$	**8.75**	**9.77**	**12.78**	**19.41**	**9.54**	**10.61**	**12.01**	**8.73**	12.27	**4.92**	**10.88**
STL10 (40)	w/o $\mathbf{M}_c^L$	2.86	4.54	11.24	9.18	3.73	7.67	8.20	4.59	9.94	10.30	7.23
	w/ $\mathbf{M}_c^L$	**3.38**	**5.12**	10.37	8.48	**5.47**	7.21	**10.42**	**5.32**	**11.15**	**10.32**	**7.72**

Table 5. Ratio between inter-and intra-class distance depending on the $\mathbf{M}_c^L$ usage for distance measurement.

Effectiveness of local metric tensors. We conducted extensive experiments to provide the effectiveness of the proposed $\mathbf{M}_c^L$ with a reliable evaluation metric. Here, Table 5 showed the ratio (r_c) of the average distance between samples of the target (c-th) class and samples of different classes (inter-class distance) to the average distance between samples of the same class (intra-class distance) in embedding space. r_c is defined as follows:

$$r_c = \frac{\frac{1}{NM}\sum_{i=1}^{N}\sum_{m=1}^{M} d(\mathbf{z}_i, \mathbf{z}_m)}{\frac{1}{N(N-1)}\sum_{i=1}^{N}\sum_{j=1}^{N-1} d(\mathbf{z}_i, \mathbf{z}_j)}, \text{ where } i \neq j, m \tag{14}$$

where d, $\mathbf{z}$, N, and M are the distance measure, embeddings, the number of samples in the c-th class ($c \in$[0,9]), and the total number of samples in other classes, respectively. As shown in this table, the ratios were increased when the $\mathbf{M}_c^L$ was used in distance measurement. This depicts that the proposed $\mathbf{M}_c^L$ can measure the better seman-

m	CIFAR-10 (40)	CIFAR-100 (400)	SVHN (40)	STL-10 (40)
0.99	4.37	37.49	2.14	20.47
0.97	4.42	37.89	2.10	19.27
0.95	4.61	38.21	2.17	23.58

Table 6. Top-1 classification error rates (%) depending on the momentum in EMA.

tic distances between data points than the Euclidean distance by understanding the intrinsic geometry on embedding spaces depending on the classes.

Evaluation depending on EMA momentum. Following the conventional evaluation trend [13, 14, 31], we employed the EMA evaluation described in Subsection 3.7. Table 6 showed the classification error rates depending on the EMA momentum (m) for the specific SSL scenarios. To compose the different ensembled model, we conducted training on the proposed method with three momentums 0.99, 0.97, and 0.95. Experimental results showed the proposed setting ($m = 0.99$) can be optimal for the overall datasets.

5. Discussion and Conclusion

We introduced the metric-tensor-based consistency regularization inducing the class-variant embedding space for the consistency training of unlabeled data. The main contributions are the usage of global and local metric tensors representing the intrinsic geometry structure on the high-dimensional embedding space for the SSL. Moreover, we designed a simple framework and architecture for the SSL scenarios. We strongly believe that the proposed method can be available for various practical applications such as autonomous manufacturing systems and future work. Extensive experiments on various datasets showed stable and high-performance results without the explosion of training parameters.

Limitations. We briefly listed the limitations of this work to discuss the direction of future work. First, it is difficult to introduce the metric tensor for the analysis of manifold space due to the heuristic property of the manifold hypothesis. Second, the pseudo-labeling in the proposed method is quite simple and it's not far from the naive techniques.

Future work. Previous works [12, 23, 27, 32] for the analysis of feature space are well-designed and still remain attractive research topics such as diffusion [29], representation learning [2, 8, 13], and multi-modal large language model [5, 42]. We hope this work can be extended to those foundation models. In addition, the proposed method can be applied to other SSL frameworks such as object detection and semantic segmentation in the future. Moreover, the metric tensor-based consistency regularization can be further developed for domain adaptation [25] due to its powerful capability of extracting geometry on the embedding space.

Acknowledgement

This research was supported by National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) under Grant RS2023-00208763, Institute of Information & communications Technology Planning & Evaluation (IITP) under the Artificial Intelligence Convergence Innovation Human Resources Development (IITP-2023-RS-2023-00254592) grant funded by the Korea government (MSIT), and LG Display (C2023001588).

References

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631, 2019. 7

[2] 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*, pages 8443–8452, 2021. 1, 8

[3] David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring. In *International Conference on Learning Representations*. 5, 6

[4] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186, Heidelberg, 2010. Physica-Verlag HD. 5

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 8

[6] 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. *arXiv preprint arXiv:2301.10921*, 2023. 2, 5, 6

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 1

[8] 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. 1, 8

[9] 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*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011. 5

[10] Ekin Dogus Cubuk, Barret Zoph, Jon Shlens, and Quoc Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Advances in Neural Information Processing Systems*, pages 18613–18624. Curran Associates, Inc., 2020. 6

[11] Xingping Dong, Jianbing Shen, Wenguan Wang, Yu Liu, Ling Shao, and Fatih Porikli. Hyperparameter optimization for tracking with continuous deep q-learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 7

[12] Felipe L Gewers, Gustavo R Ferreira, Henrique F De Arruda, Filipi N Silva, Cesar H Comin, Diego R Amancio, and Luciano da F Costa. Principal component analysis: A natural approach to data exploration. *ACM Computing Surveys (CSUR)*, 54(4):1–34, 2021. 8

[13] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020. 1, 5, 8

[14] 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. 5, 8

[15] Zhiqi Kang, Enrico Fini, Moin Nabi, Elisa Ricci, and Karteek Alahari. A soft nearest-neighbor framework for continual semi-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11868–11877, 2023. 1

[16] Jiwon Kim, Youngjo Min, Daehwan Kim, Gyuseong Lee, Junyoung Seo, Kwangrok Ryoo, and Seungryong Kim. Conmatch: Semi-supervised learning with confidence-guided consistency regularization. In *European Conference on Computer Vision*, pages 674–690. Springer, 2022. 2, 5, 6

[17] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

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

[19] 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*, page 896. Atlanta, 2013. 5, 6

[20] Zekun Li, Lei Qi, Yinghuan Shi, and Yang Gao. Iomatch: Simplifying open-set semi-supervised learning with joint inliers and outliers utilization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15870–15879, 2023. 1

[21] 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. 2, 5, 6

[22] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? *Advances in neural information processing systems*, 32, 2019. 4

[23] Ashwin Narayan, Bonnie Berger, and Hyunghoon Cho. Assessing single-cell transcriptomic variability through density-preserving data visualization. *Nature Biotechnology*, 2021. 8

[24] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011. 5

[25] Ba Hung Ngo, Yeon Jeong Chae, Jung Eun Kwon, Jae Hyeon Park, and Sung In Cho. Improved knowledge transfer for semi-supervised domain adaptation via trico training strategy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19214–19223, 2023. 8

[26] Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. *Advances in neural information processing systems*, 31, 2018. 1

[27] So Jeong Park, Hae Ju Park, Eun Su Kang, Ba Hung Ngo, Ho Sub Lee, and Sung In Cho. Pseudo label rectification via co-teaching and decoupling for multisource domain adaptation in semantic segmentation. *IEEE Access*, 10:91137–91149, 2022. 8

[28] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2019. 5

[29] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 8

[30] 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. 1, 2, 5, 6

[31] 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. 5, 6, 8

[32] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9 (11), 2008. 8

[33] Yidong Wang, Hao Chen, Yue Fan, Wang Sun, Ran Tao, Wenxin Hou, Renjie Wang, Linyi Yang, Zhi Zhou, Lan-Zhe Guo, et al. Usb: A unified semi-supervised learning benchmark for classification. *Advances in Neural Information Processing Systems*, 35:3938–3961, 2022. 5

[34] Yidong Wang, Hao Chen, Qiang Heng, Wenxin Hou, Yue Fan, Zhen Wu, Jindong Wang, Marios Savvides, Takahiro Shinozaki, Bhiksha Raj, Bernt Schiele, and Xing Xie. Freematch: Self-adaptive thresholding for semi-supervised

learning. In *The Eleventh International Conference on Learning Representations*, 2023. 2

[35] Hao Wu, Yongqiang Cheng, Xixi Chen, Xiang Li, and Hongqiang Wang. Adaptive matrix information geometry detector with local metric tensor. *IEEE Transactions on Signal Processing*, 70:3758–3773, 2022. 2

[36] 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. 5, 6

[37] 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*, 2021. 1, 2, 5, 6

[38] Lihe Yang, Lei Qi, Litong Feng, Wayne Zhang, and Yinghuan Shi. Revisiting weak-to-strong consistency in semi-supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7236–7246, 2023. 1

[39] Lihe Yang, Zhen Zhao, Lei Qi, Yu Qiao, Yinghuan Shi, and Hengshuang Zhao. Shrinking class space for enhanced certainty in semi-supervised learning. In *ICCV*, 2023. 5, 6

[40] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference 2016*. British Machine Vision Association, 2016. 5

[41] 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. 2, 5, 6

[42] Renrui Zhang, Jiaming Han, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, Peng Gao, and Yu Qiao. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint arXiv:2303.16199*, 2023. 8

[43] Mingkai Zheng, Shan You, Lang Huang, Fei Wang, Chen Qian, and Chang Xu. Simmatch: Semi-supervised learning with similarity matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14471–14481, 2022. 5, 6