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

Supplementary Material

The supplementary and its contents are summarized as follows:

- **Related works:** Brief reviews for the existing semi-supervised learning and state-of-the-art methods.
- **Details of architecture:** Detail structures of the encoder, projector, and classifier. The number of trainable parameters.
- **Distance measurement:** Extensive experiments to compare the distance measurement on the embedding space.
- **Training descriptions:** Learning rate, batch size, optimizer parameters, the number of data, weak and strong augmentations, learning rate scheduler, and iteration steps.

6. Related Works

Semi-supervised learning. The optimization formula of consistency regularization on the semi-supervised learning for image classification can be summarized as follows:

$$\mathcal{L}_{SSL} = \mathcal{L}_{sup} + \mathcal{L}_{unsup}, \tag{15}$$

where $\mathcal{L}_{sup}$ represents the supervised learning loss such as the cross-entropy. $\mathcal{L}_{unsup}$ is the regularization term to encourage the consistent prediction of the unlabeled data.

Pseudo labeling [19] of the unlabeled data applies a self-supervision of the high-confident samples as follows:

$$\mathcal{L}_{unsup} = \frac{1}{\mu B}\sum_{i=1}^{\mu B} \mathbb{I}_{[\max(\mathbf{h}_i)\geq\tau]} H(\hat{\mathbf{y}}_i, f(p(x_i^u))), \tag{16}$$

where $\mathbf{h}_i$ is the probability prediction of the unlabeled image (x_i^u). $\hat{\mathbf{y}}_i$ is the one-hot-encoded vector of $\mathbf{h}_i$. τ represents a threshold value. p represents the stochastic perturbation described in Sec. 1. μB is the batch size of the unlabeled data. H represents a criterion that measures the distance or discrepancy between two vectors such as the cross-entropy. Thus, high-confident unlabeled samples are used as the labeled data to extend the training data with their predicted class by the trained model.

FixMatch [30] uses cross-entropy loss between the pseudo label derived from the weakly augmented version $(\hat{\mathbf{y}}_\alpha)$ of the unlabeled sample $\alpha(x^u)$ and the prediction of the strongly augmented version $(\hat{\mathbf{y}}_\mathcal{A})$ of the unlabeled sample $(\mathcal{A}(x^u))$ as follows:

$$\mathcal{L}_{unsup} = \frac{1}{\mu B}\sum_{i=1}^{\mu B} \mathbb{I}_{[\max(\mathbf{h}_i)\geq\tau]} H(\hat{\mathbf{y}}_i, f(\mathcal{A}(x_i^u))), \tag{17}$$

Group name	Output size	Block type
Input	$H \times W$	-
Conv_1	$H \times W$	$\begin{bmatrix} 3\times 3, & 16 \end{bmatrix}$
Conv_2	$H \times W$	$\begin{bmatrix} 3\times 3, & 32 \\ 3\times 3, & 32 \end{bmatrix} \times N$
Conv_3	$H/2 \times W/2$	$\begin{bmatrix} 3\times 3, & 64 \\ 3\times 3, & 64 \end{bmatrix} \times N$
Conv_4	$H/4 \times W/4$	$\begin{bmatrix} 3\times 3, & 128 \\ 3\times 3, & 128 \end{bmatrix} \times N$
Average Pooling	1×1	$\begin{bmatrix} H/4 \times W/4 \end{bmatrix}$

Table 7. Structure of (a) feature extractor in Wide-ResNet-28-w2.

where $\mathcal{A}$ is the strong augmentation. Unlike the pseudo labeling, $\mathbf{h}_i$ represents the prediction of the weakly augmented samples. Thus, FixMatch employs consistency regularization by minimizing the discrepancy between two differently augmented images.

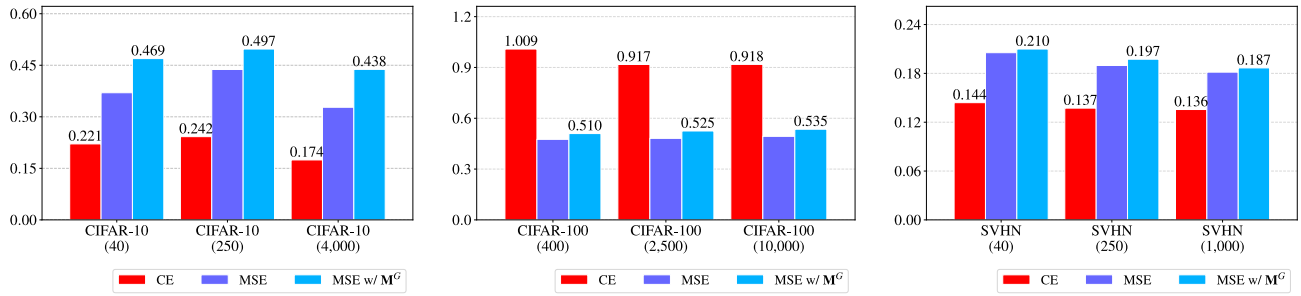
Dash [37] applies pseudo labeling to the unlabeled data having a lower consistency loss than the cross-entropy loss of the supervised learning as follows:

$$\begin{aligned} \mathcal{L}_{sup} &= \frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C} -\mathbf{y}(i,c)\log\hat{\mathbf{y}}_l(i,c), \\ \mathcal{L}_{unsup} &= \frac{1}{\mu B}\sum_{i=1}^{\mu B} \mathbb{I}_{[H(\hat{\mathbf{y}}_\alpha(i),\hat{\mathbf{y}}_\mathcal{A}(i))\leq\tau]} H(\hat{\mathbf{y}}_\alpha(i), \hat{\mathbf{y}}_\mathcal{A}(i)), \\ \tau &= C\gamma^{-(t-1)}\mathcal{L}_{sup}, \end{aligned} \tag{18}$$

where $\mathcal{L}_{sup}$ represents the cross-entropy of the labeled data. C and γ is the predefined constants and t represents a training step. The high-confident unlabeled samples, which have the lower cross-entropy between the $\hat{\mathbf{y}}_\alpha$ and $\hat{\mathbf{y}}_\mathcal{A}$ than the $\mathcal{L}_{sup}$, are used for the pseudo labeling. By this, the threshold value is dynamically set depending on the learning status of a model with the labeled data.

7. Details of Architecture

Encoder. Tables 7 and 8 show the encoder (g) structures of Wide-ResNet-28-w2 and Wide-ResNet-28-w8, respectively. N was the number of blocks in a group and was set to 4 in our experiments. The down-sampling is performed by the first layers in groups Conv_3 and Conv_4. A batch normalization and a rectified linear unit (ReLU) activation



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

Figure 4. The comparison of global distance measurement with $\mathbf{M}^G$ on the test phase in the embedding space. The cross-entropy (CE) between predictions and ground-truth was provided as the index of the classification performance of the model rather than the purpose of the distance comparison.

Group name	Output size	Block type
Input	$H \times W$	-
Conv_1	$H \times W$	$\begin{bmatrix} 3\times 3, & 16 \end{bmatrix}$
Conv_2	$H \times W$	$\begin{bmatrix} 3\times 3, & 128 \\ 3\times 3, & 128 \end{bmatrix} \times N$
Conv_3	$H/2 \times W/2$	$\begin{bmatrix} 3\times 3, & 256 \\ 3\times 3, & 256 \end{bmatrix} \times N$
Conv_4	$H/4 \times W/4$	$\begin{bmatrix} 3\times 3, & 512 \\ 3\times 3, & 512 \end{bmatrix} \times N$
Average Pooling	1×1	$[H/4 \times W/4]$

Table 8. Structure of feature extractor in Wide-ResNet-28-w8.

Dataset	CIFAR-10	CIFAR-100	SVHN	STL-10
#Params.	1.5M	23M	1.5M	1.5M

Table 9. The number of the trainable parameters.

are applied between each convolution. H and W represent the height and width of an input image.

Projector. Table 10 shows the structure of the projector ($\mathcal{P}$) consisting of three linear layers. The L2-normalization (Norm.) layer is placed as the last layer.

Classifier is constructed only with one linear layer without an activation function.

Table 9 shows the number of trainable parameters depending on the conventional datasets. In particular, the number of parameters in the CIFAR-10 experiment is larger because the Wide-ResNet-28-w8 was used as the backbone, which is a large-depth version compared to the Wide-ResNet-28-w2.

Layer name / **Widen_factor** (W)	**Input size** $W=2$	**Input size** $W=8$	**Output size** $W=2$	**Output size** $W=8$
Input	128	512	-	-
Dense_1	128	512	128	512
Batch Norm.	-	-	128	512
ReLU	-	-	128	512
Dense_2	128	512	128	512
Batch Norm.	-	-	128	512
ReLU	-	-	128	512
Dense_3	128	512	128	512
L2-Norm.	-	-	128	512

Table 10. The structure of the projector in the proposed method.

8. Distance Measurement

We measured the distance between different views in the embedding space on the test data. This depicts the remarkable effectiveness of the proposed metric tensor in terms of distance measurement. Specifically, we calculated the distance between two differently augmented test images on the embedding space with two measurements: mean squared errors (MSE) in Eq. 1 and MSE with metric tensor ($\mathbf{M}^G$) in Eq. 2. We generated the two different views by using weak and strong augmentations.

Figure 4 shows the results of the distance measurement on the embedding space depending on the datasets. We displayed the average cross-entropy and distances from the test images. The cross-entropies between predictions and ground truth were provided as the index of the classification performance of the model rather than the purpose of the comparison. We compared the distance values by using the proposed method. Note that, it is difficult to compare the proposed and the benchmark methods because all the feature spaces from the different models were quite different. Nevertheless, the comparison results provided a consider-

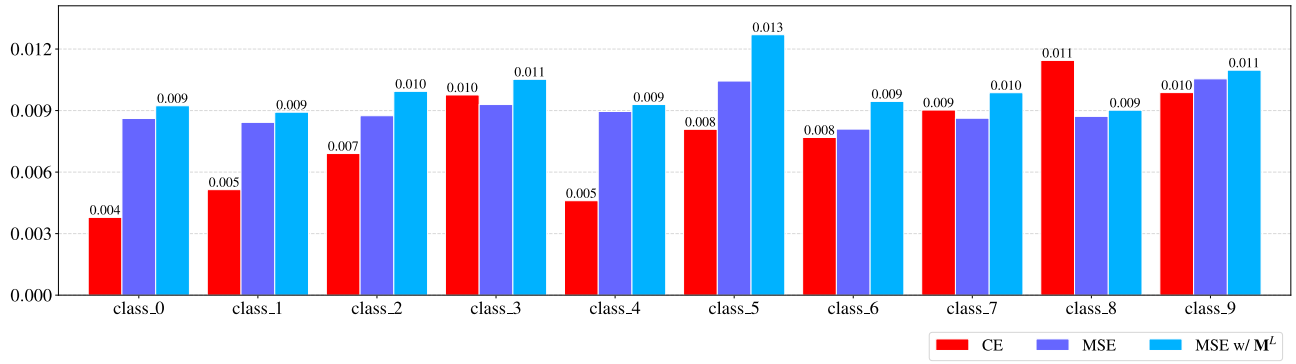


Figure 5. The comparison of class-wise distance measurement with $\mathbf{M}^L$ on STL-10 dataset in the embedding space. This figure is extracted by the number of 1,000 labeled data scenario.

Key	CIFAR-10	CIFAR-100	SVHN	STL-10
N	32	32	32	32
μB	64	64	64	64
Optimizer	SGD	SGD	SGD	SGD
Learning rate	0.03	0.03	0.03	0.03
Momentum	0.9	0.9	0.9	0.9
Weight decay	0.0005	0.001	0.0005	0.0005
m	0.99	0.99	0.99	0.99
Iteration	1M	1M	1M	1M

Table 11. Important parameters setting for the proposed method.

able aspect. For example, Fig. 4 showed that the distance between two views returns a smaller MSE than MSE with $\mathbf{M}^G$. These occurred for all extensive experiments. These depicted that there is a more complex relationship between two embeddings in terms of the semantic distance rather than expressing the Euclidean distance. In other words, the proposed method calculates the distance by considering the geometric curvature of the embedding space rather than calculating the distance in Euclidean space, so it can be interpreted that the MSE with metric tensor ($\mathbf{M}^G$)-based distance is derived to be larger than the MSE-based distance.

Figure 5 shows the distance measurements on the STL-10 dataset. We compared the distance in the same manner in Fig. 4. The only difference was that each class was separately calculated with a class-wise metric tensor ($\mathbf{M}_c^L$). As shown in this figure, the distance between the two views was differently observed which means the local metric tensor induced the different geometric structure in the embedding space. Thus, the class-wise distance empirically showed the variety of the embedding spaces depending on the class.

9. Training Descriptions

Data populations. The number of training and test data in the CIFAR family are 50K and 10K, respectively. The SVHN composes 73,257 samples for training and 26,032 samples for testing. The STL-10 dataset involves 105,000 samples for training and 8,000 samples for testing. The labeled and unlabeled data are divided into two groups based on the predefined SSL scenarios.

Hyper-parameters. Table 11 shows the specification of the training setup depending on the dataset. We unified the almost hyper-parameters as much as possible based on the previous works [33].

Augmentations. We applied the *RandomHorizontalFlip*, *RandomCrop*, and *Normalize* as the weak augmentation. In addition, *AutoContrast*, *Brightness*, *Color*, *Contrast*, *Equalize*, *Identity*, *Posterize*, *Rotate*, *Sharpness*, *Shearing*, *Solarize*, and *Translate* were additionally used as the strong augmentation. The strongly augmented image is generated by combining the three augmentations. We applied the warmup cosine schedule [28] with ten warmup steps.