Dynamic Pseudo Labeling via Gradient Cutting for High-Low Entropy Exploration

Supplementary Material

The supplementary and its contents are summarized as follows:

- **Key concept:** This section provides the additional description for the proposed four learning status.
- **Evaluation Setup:** This section provides the implementation setting for the inference stage in this work.
- **Implementation Details:** This section provides detailed descriptions of data augmentation, learning rate, learning rate schedule, weight decay, and iteration settings for the training of the proposed model.
- **Experimental Results depending on percentile (p):** This section provides the experimental results of the proposed method depending on the percentile value in Subsection 3.6, Eq.(12).
- **Sampling Rate:** This section provides the number of pseudo-labels derived from confident unlabeled data used in each pseudo-labeling (PL) method during training at various steps.
- **t-SNE Visualization:** This section provides the visualization results on the model's embedding space from the benchmark and proposed methods.

A. Key Concept

Figure A provides the data distribution (randomly selected 1K anchor-positive pairs on CIFAR-10) based on the four-LS. FreeMatch provided a more overconfident status (36.7%) than the proposed method (31.0%). We focused on eliminating this overconfidence in DPL training. Figure B shows how HGP and LGP estimate over- and under-confident status based on Fig. A. As shown in Fig. B, PL w/ HGP (clipping low-gradients) relies more on high-gradient samples (A and C) than f. Thus, HGP's learning status is closer to A than that of f. Similarly, PL w/ LGP (clipping high gradients) is closer to D than f. With this, HGP and LGP can induce a relatively more over- and under-confident status than f by clipping the low and high gradients, respectively. The different learning status (LS) between f, LGP, and HGP are provided in Fig. 5 showing PL loss variation. Thus, we could estimate over- and under-confident LS ($\mathcal{T}^{\text{HGP}}$ and $\mathcal{T}^{\text{LGP}}$) and use them to adjust $\mathcal{T}^{f}$ for PL thresholding depending on the degree of the f's overconfidence. Figure B shows an example of threshold adjustment. NEMaxNorm encourages an increase in the quantity of pseudo-labels by reducing $\mathcal{T}^{f}$ for class-adaptively.

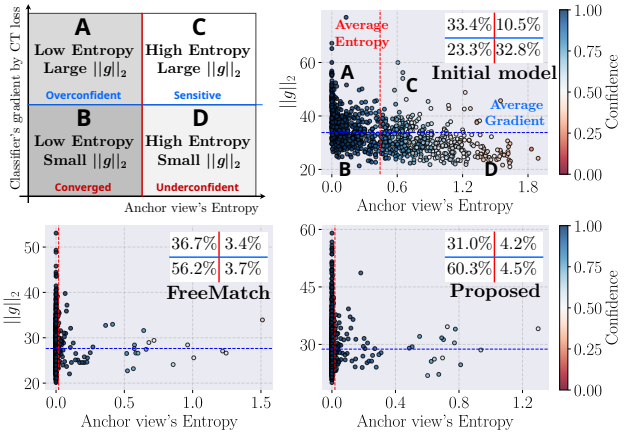

Figure A. Visualization of the four-learning status with CIFAR-10 dataset.

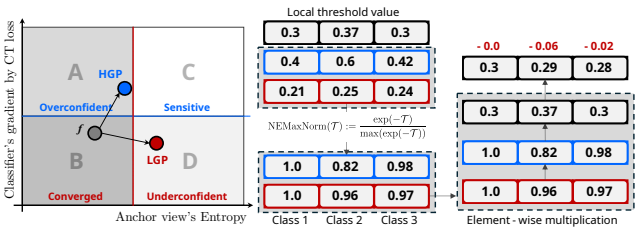

Figure B. (left) The connectivity between GP and learning status, and (right) flowchart of the adjustment of threshold, $\mathcal{T}^{\text{GCD}}$ using $\mathcal{T}^{\text{HGP}}$ and $\mathcal{T}^{\text{LGP}}$.

B. Evaluation Setup

Table A shows the specifications of the inference consisting of four datasets: Canadian institute for advanced research (CIFAR-10, 100), street view house numbers (SVHN), and self-taught learning (STL-10). The mean and standard deviation (Mean and Std.) of images in the dataset serve as normalization factors to transform the image values into a distribution $\mathcal{N}(0, \delta^2)$, calculated as $(\mathbf{x}_c - \mu_c) \cdot \delta_c$ for the c-th channel.

Table A. The specifications of inference depending on the datasets.

Dataset	Image size	# Data	Mean.	Std.	# Class
CIFAR-10	3×32×32	10K	[0.485, 0.456, 0.406]	[0.229, 0.224, 0.225]	10
CIFAR-100	3×32×32	10K	[0.507, 0.487, 0.441]	[0.267, 0.256, 0.276]	100
SVHN	3×32×32	10K	[0.438, 0.444, 0.473]	[0.175, 0.177, 0.174]	10
STL-10	3×96×96	5K	[0.441, 0.428, 0.387]	[0.268, 0.261, 0.269]	10

Table B. The specifications of inference depending on the datasets.

Dataset	CIFAR-10		CIFAR-100	
p	40	250	400	10,000
90	4.51	4.49	31.54	20.97
80 (proposed)	4.52	4.39	31.09	20.13
70	5.75	4.57	32.23	21.58

C. Implementation Details

C.1. Data augmentation

We applied horizontal flipping with a probability of 0.5, followed by randomly cropping 87.5% of the image and padding it to the original size. These augmentations were used to generate the labeled ($\mathbf{s}$) and anchor sets ($\mathbf{x}$) in Subsection 3.1. RandAug [7] was used for generating the positive set ($\mathbf{x}^+$) in Subsection 3.1 consisting of Contrast, Brightness, Color, Posterization, Rotation, Sharpening, Shearing, Solarization, Translation, and Cutout functions. We randomly chose the three augmentations among them to generate positive samples during training.

C.2. Optimiaztion

We used the learning rate of 0.03, cosine decay as the learning rate scheduler, weight decay of 0.0005, and the total number of training steps of 2^{20} in the proposed method.

D. GradCut depending on Percentile (p)

Figure C shows the simple illustration of GradCut depending on the p. As shown in this figure, the smaller p makes the smaller ranges of GradCut. Table B shows the Top 1 classification error rates (%) depending on the p in Subsection 3.6, Eq. (12). As shown in this table, experiments with small p (70, 60) provided a slightly degraded classification accuracy compared to the optimal set 80. We generated the two different views by using weak and strong augmentations. This showed that experiments with small p (70, 60) provided a slightly degraded classification accuracy compared to the optimal set 80. This means that the proposed GradCut significantly affects generalization performance by inducing high-entropy predictions from GP classifiers. When the p was set to 90, the classification accuracy slightly decreased but still provided the pleasing classification accuracy thus representing the stability of the proposed GradCut.

E. Sampling Rate

Figure D showed the sampling rates depending on the benchmark methods and proposed method. The "sampling rate" refers to the proportion or percentage of samples in the mini-batch that are considered pseudo-labels. For example, the sampling rate of 0.5 means that half of the samples in a mini-batch, those with confidence greater than the threshold ($\mathcal{T}$), will be used for model updates. As shown in this figure, the green line, labeled as "Actual", showed the observed sampling rate at each iteration. A red dashed line helps to compare the slope of the global threshold variations. FixMatch [32], CRMatch [9], and SimMatch [46] rapidly reached a high utilization of confident samples compared to the proposed method, which means the rapid convergence of the model's prediction confidence. Dash [39] showed an initial sharp drop in the sampling rate, followed by a gradual and steady increase over subsequent iterations. This behavior can be caused by overconfidence in the model's prediction. This is because the pseudo-labeling loss of PseudoLabel [18] in Section 2 that can accelerate the overconfidence is adopted as the threshold value for selecting confident positive samples (strongly augmented).

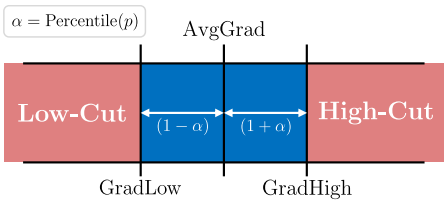

Figure C. Visualization of GradCut operation depending on percentile p and its quantile α.

F. t-SNE Visualization

Figure E showed the t-SNE [34] visualizations of embeddings from the benchmark methods and proposed method depending on the datasets. We followed the experimental setting on a unified semi-supervised learning codebase (USB) [36] for the benchmark experiments. As shown in this figure, Dash in Section 2 significantly showed the high error rate (>0.8) for the 5th class which means the confirmation bias. FixMatch [32] and CRMatch [9] provided high error rates for certain classes (e.g. 2, 3, and 5), which also caused confirmation bias. In conclusion, the proposed method provided the most similar error rate with fully supervised training in terms of generalization ability.

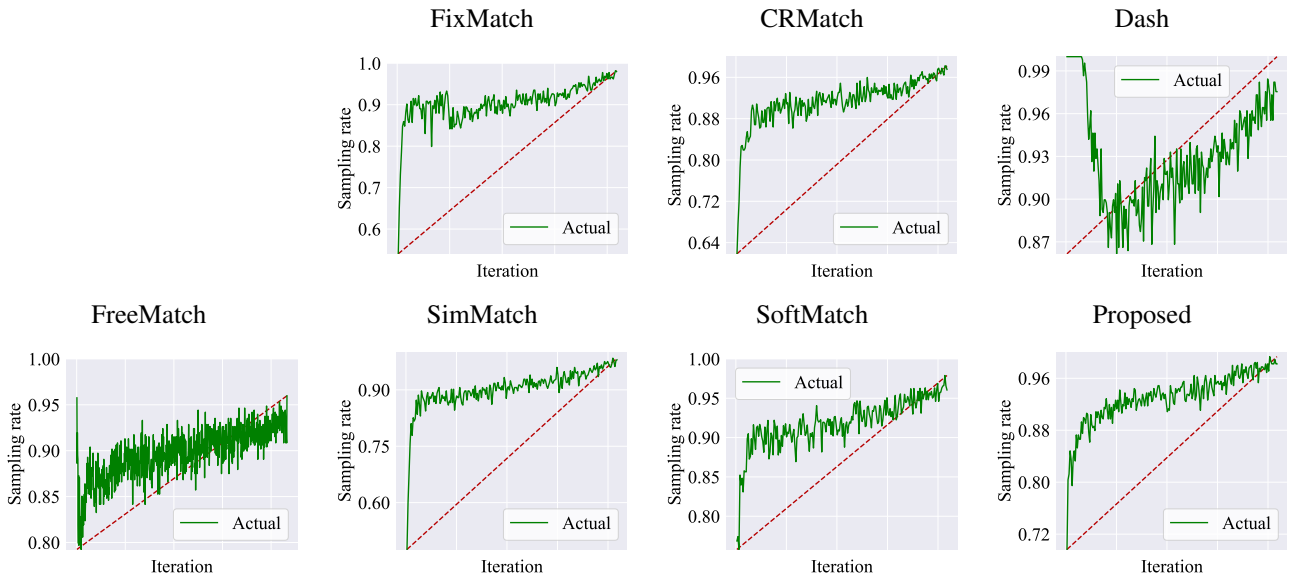

Figure D. The sampling rates depending on the benchmark methods and proposed method in CIFAR-10 (40) scenario.

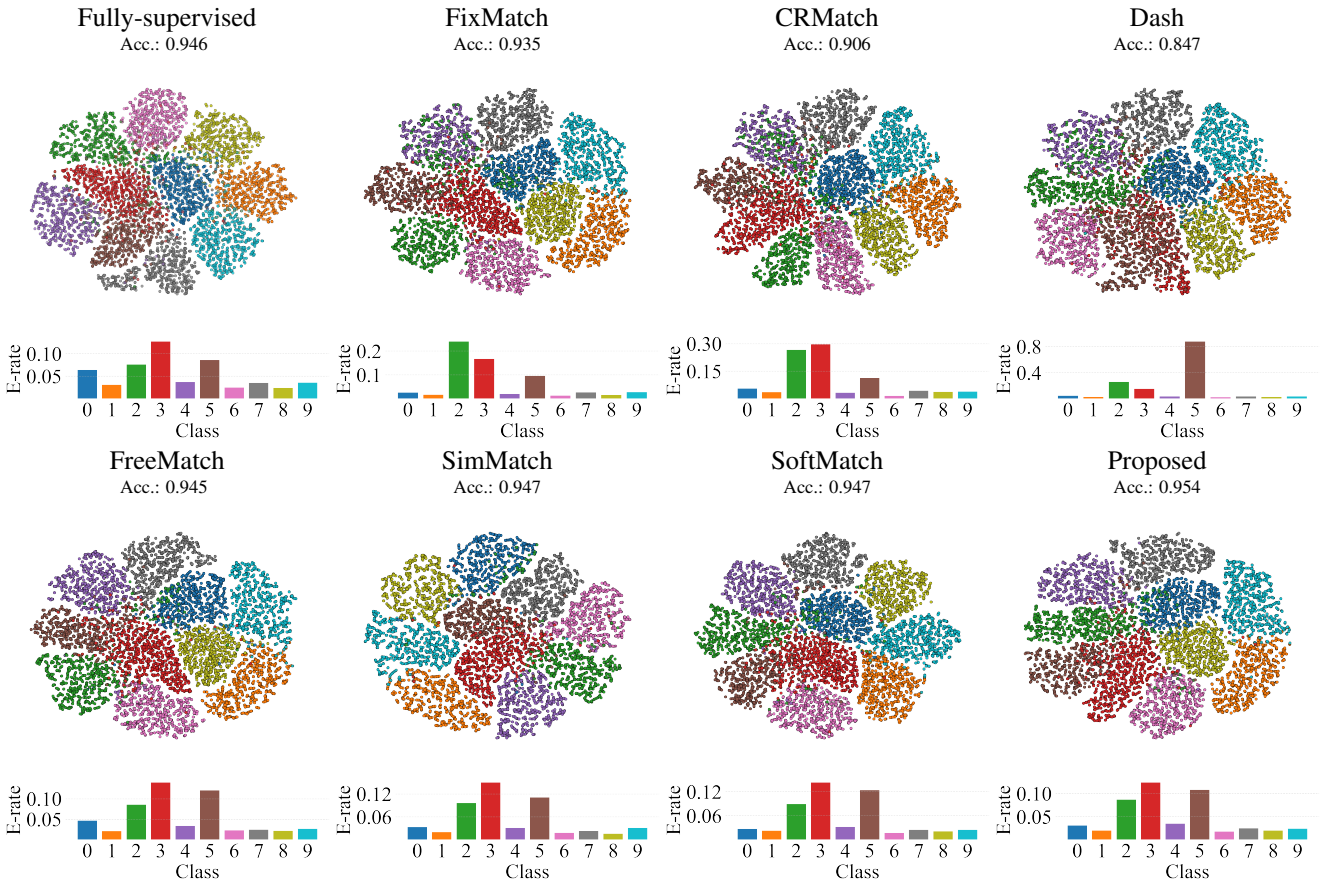

Figure E. t-SNE visualizations of embeddings from testing data depending on the benchmark methods and proposed method in CIFAR-10 (40) scenario. The differently colored circles indicate each class drawn by ground-truth labels. The bar graph provides the corresponding color class's error rate (E-rate).