A. Proof for the gradient of target class in EML

Here we provide the proof of the Eq.7 in Section 3.2 of the main paper. First, we can simplify $y_c$ based on the output probability of target class $p_{tc}$:

$$y_c = \frac{1 - p_{tc}}{C - 1}$$  (1)

we omitted the superscript $i$ in which denotes the sample here. The gradient of the EML loss with respect to the target class can be calculated by the chain rule:

$$\frac{\partial L_{eml}}{\partial p_{tc}} = \frac{\partial L_{eml}}{\partial y_c} \cdot \frac{\partial y_c}{\partial p_{tc}}$$  (2)

we can calculate the gradient of the first term from according to Eq.(7) of the main paper:

$$\frac{\partial L_{eml}}{\partial y_c} = -\frac{1}{BC} \sum_{c \in D} \left[ \log(p_c) - \log(1 - p_c) \right]$$

$$= \frac{1}{BC} \sum_{c \in D} \left[ \log(1 - p_c) - \log(p_c) \right]$$

$$= \frac{1}{BC} \sum_{c \in D} \log(1 - p_c) - \sum_{c \in D} \log(p_c)$$  (3)

$$= \frac{1}{BC} \log \prod_{c \in D} (1 - p_c) - \log \prod_{c \in D} p_c$$

$$= \frac{1}{BC} \log \prod_{c \in D} (1 - p_c) \prod_{c \in D} p_c$$

where $D = \{c|c \in [1, C] \& c \neq tc\}$. we can compute the gradient of the second term from according to Eq.(2):

$$\frac{\partial y_c}{\partial p_{tc}} = -\frac{1}{C - 1}$$  (4)

Thus, we can obtain the overall gradient according to Eq.(3) and Eq.(4):

$$\frac{\partial L_{eml}}{\partial p_{tc}} = -\frac{1}{BC(C - 1)} \log \frac{\prod_{c \in D} (1 - p_c)}{\prod_{c \in D} p_c}$$  (5)

which finishes the proof of Eq.(7) in the main paper. Furthermore, we will analysis the gradient directions of EML and cross-entropy loss are the same.
limited labeled samples. As shown in Fig. 1, when operating on CIFAR-100 with 400 label samples (i.e., 4 labels per class), the accuracy of PL (i.e., pseudo label) (yellow) is just about 65%, while NPL (red) still maintains a very high accuracy (about 95%) without reducing the selected amount (i.e., $k$ is not clearly changed). This further demonstrates the effectiveness of NPL.

B.2. Initial Noise

To illustrate that there is no noise perturbation in ANL even in the beginning of training, we visualize the accuracy of top-$k$ in the first 1K iterations, as shown in Fig. 2. In the initial iterations, our calculated $k$ is close to the class number. Hence, the amount of selected NPL is very small (e.g., only assign the last-1 prediction as NPL). The possible noise is rather minor. After few iterations (no more than 20), the probability is extremely small that the positive class is the last-ranked one, so the accuracy is near to 1 without noise issue.

C. Analysis about EML

In this section, we compare some similar methods with our proposed EML, as shown in Table 1. Seeing col.2 vs col.3, it leads to a failure training when applying Label Smooth (LS) (target class label is set to 0.90) into unsupervised loss $\mathcal{L}_{us}$, since it will decrease the score of the target class thus leading to missing abundant pseudo-label samples. When using LS (0.98), the issue can be significantly alleviated. The last 3 columns show Entropy Regularization (ER) can benefit FixMatch and can be used with EML in a complementary fashion.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>CIFAR-10</th>
<th>CIFAR-100</th>
<th>SVHN</th>
<th>STL-10</th>
<th>ImageNet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weight Decay</td>
<td>0.0005</td>
<td>0.001</td>
<td>0.0005</td>
<td>0.0005</td>
<td>0.0001</td>
</tr>
<tr>
<td>Threshold $\tau$</td>
<td>0.95</td>
<td>0.95</td>
<td>0.95</td>
<td>0.95</td>
<td>0.7</td>
</tr>
<tr>
<td>Labeled Data Batch Size</td>
<td>64</td>
<td>64</td>
<td>64</td>
<td>64</td>
<td>128</td>
</tr>
<tr>
<td>Unlabeled Data Ratio $\mu$</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>Learning Rate</td>
<td>0.03</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SGD Momentum</td>
<td>0.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EMA Momentum</td>
<td>0.999</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2. Hyperparameter settings of FullMatch/FullFlex for CIFAR-10, CIFAR-100, SVHN and STL-10.

D. Experiment and Algorithm

D.1. FullMatch Algorithm

We present the complete algorithm for FullMatch in Algorithm 1. Please refer to the main paper for all symbols and equations.

D.2. Implementation Details

For reproduction, we present the complete list of hyperparameters for FullMatch and FullFlex when operating in different benchmarks, as shown in Table 2, which are mainly consistent with TorchSSL settings.

Algorithm 1 FullMatch algorithm.
1: **Input:** $X = \{(x_m, y_m) : m \in (1, ..., M)\}$, $U = \{\mu_n : n \in (1, ..., N)\}$, $\tau$ is the confidence threshold. \{M labeled data and N unlabeled data\};
2: while not reach the maximum iteration do
3: Generate $P^{(i)}$ and $Q^{(i)}$ \{Predictions of strongly-augmented and weakly-augmented version, respectively\};
4: Calculate $k$ using Eq. (8) \{Generate negative pseudo-labels for all unlabeled examples\};
5: Calculate ANL loss $\mathcal{L}_{anl}$ using Eq. (10) for all unlabeled examples;
6: if $\max(Q^{(i)}) \geq \tau$ then
7: Calculate unsupervised loss $\mathcal{L}_{us}$ using Eq. (2);
8: Calculate $y_c$ using Eq. (5) \{Determine the label of non-target categories\};
9: Calculate EML $\mathcal{L}_{eml}$ using Eq. (6);
10: else
11: $\mathcal{L}_{eml} = 0$, $\mathcal{L}_{us} = 0$. \{Ignore the examples without pseudo-labels\};
12: end if
13: Calculate supervised loss $\mathcal{L}_s$ using Eq. (12);
14: Update model via $\mathcal{L}_{sum} = \mathcal{L}_s + \mathcal{L}_{us} + \mathcal{L}_{anl} + \mathcal{L}_{eml}$;
15: end while
16: return Model parameters