Towards Semi-supervised Learning with Non-random Missing Labels
- Supplementary Material -

A. Algorithm of PRG$^{\text{Last}}$

Algorithm 2: PRG$^{\text{Last}}$: PRG Using Class Predictions from the Last Epoch

```
Input: class tracking matrices $\mathcal{C} = \{\mathbf{C}^{(i)}; i \in (1, ..., N_B)\}$, labeled training dataset $D_L$, unlabeled training dataset $D_U$, model $\theta$,
       label bank $\{l^{(i)}; i \in (1, ..., n_T - n_L)\}$
for $n = 1$ to $\text{MaxIteration}$ do
    From $D_L$, draw a mini-batch $\mathcal{B}_L = \{(x_L^{(b)}, y_L^{(b)}); b \in (1, ..., B)\}$
    From $D_U$, draw a mini-batch $\mathcal{B}_U = \{(x_U^{(b)}); b \in (1, ..., B_U)\}$
    $\mathbf{H} = \text{RowWiseNormalize}(\text{Average}(\mathcal{C}))$          // Construct transition matrix
    $H'_{ij} = \frac{H_{ij}}{\frac{L_j}{\sum_{d=1}^{k} L_d}}$          // Rescale $\mathbf{H}$ at class-level
    for $b = 1$ to $B_U$ do
        $p^{(b)} = f_\theta(x_U^{(b)})$          // Compute model prediction
        $\text{idx} = \text{Index}(x_U^{(b)})$          // Obtain the index of $x_U^{(b)}$ in $D_U$
        $\tilde{p}^{(b)} = \text{Normalize}(H'_{l^{(\text{idx})}} \circ p^{(b)})$          // Perform pseudo-rectifying guidance
        $\hat{p}^{(b)} = \arg\max(p^{(b)})$          // Compute class prediction
        if $l^{(\text{idx})} \neq \hat{p}^{(b)}$ then
            $C^{(n)}_{l^{(\text{idx})} \hat{p}^{(b)}} = C^{(n)}_{l^{(\text{idx})} \hat{p}^{(b)}} + 1$          // Perform class transition tracking
            $l^{(\text{idx})} = \hat{p}^{(b)}$
        end
    end
    $\mathcal{L}_L, \mathcal{L}_U = \text{FixMatch}\left(\mathcal{B}_L, \mathcal{B}_U, \{\tilde{p}^{(b)}; b \in (1, ..., B_U)\}\right)$          // Run an applicable SSL learner
    $\theta = \text{SGD}(\mathcal{L}_L + \mathcal{L}_U, \theta)$          // Update model parameters $\theta$
end
```

B. Discussion on Re-Weighting Scheme of H

In this section, we give insights into re-weighting scheme of $\mathbf{H}$ in Eq. (6) based on the following theoretical justification. Overall, we give an explanation from the perspective of gradient. Our re-weighting scheme potentially scale the gradient magnitude on the learning of the unlabeled data to mitigate adverse effects of biased labeled data. Letting p be the naive soft label vector, by Eq. (6), we re-weight $\mathbf{H}$ by $H'_{ij} = \times \frac{H_{ij}}{\frac{L_j}{\sum_{d=1}^{k} L_d}}$ and obtain the rescaled pseudo-label vector $\tilde{p} = \text{Normalize}(\mathbf{H}' \circ p)$. Hence, the cross-entropy between prediction p and $\tilde{p}$ can be formalized as

$$
\begin{aligned}
\mathcal{L}_U = -\sum_{c}^{k} \tilde{p} \log p_c &= -\sum_{c}^{k} \left(\frac{H'_{ij} \times p_c}{\mathcal{Z}} \right) \log p_c \\
&= -\sum_{c}^{k} \left(\frac{H_{\hat{p}c} \times p_c}{\mathcal{Z} \frac{L_c}{\sum_{d=1}^{k} L_d}} \right) \log p_c, \qquad (8)
\end{aligned}
$$

where $\mathcal{Z}$ is the normalize factor. $\frac{L_c}{\sum_{d=1}^{k} L_d}$ can be regarded as the ratio of pseudo-labels belonging to class c to all labels. Denoting the logit outputted from the model as o (implying $p = \mathrm{Softmax}(o)$), with no gradient on pseudo-label $\tilde{p}$, we obtain $\frac{\partial \mathcal{L}_U}{\partial o_c} = -\sum_c^k \frac{\tilde{p}_c}{p_c} \frac{\partial p_c}{\partial o_c}$, *i.e.*,

$$\frac{\partial \mathcal{L}_U}{\partial o_c} = -(\tilde{p}_c - \tilde{p}_c p_c - \sum_{i \neq c}^{k} \tilde{p}_i p_c) \tag{9}$$

$$= \left(1 - \frac{H_{\hat{p}c}}{\mathcal{Z} \frac{L_c}{\sum_{d=1}^{k} L_d}} \right) p_c. \tag{10}$$

The larger the difference between $H_{\hat{p}c}$ and $\frac{L_c}{\sum_{d=1}^{k} L_d}$, the larger the gradient; and the smaller the difference between $H_{\hat{p}c}$ and $\frac{L_c}{\sum_{d=1}^{k} L_d}$, the smaller the gradient ($\frac{\partial \mathcal{L}_U}{\partial o_c} = 0$ when $\frac{H_{\hat{p}c}}{\mathcal{Z} \frac{L_c}{\sum_{d=1}^{k} L_d}} = 1$). This means that we intend to provide unbiased guidance (because this is derived from the unlabeled data) for the learning of unlabeled samples from the class level, so as to resist the influence of biased labeled samples. In addition, the idea behind this re-weighting scheme is that the model should increase the learning effort for rare classes (the less labels a class is assigned, the smaller the $\frac{L_c}{\sum_{d=1}^{k} L_d}$, the larger the gradient) rather than overlearn popular classes. This will implicitly lead to the model not carrying out too many pseudo-rectifying processes resulting in more labels transition to classes with too many labels assigned, but trying to assign labels to rare classes.

C. Implementation Details

In this section, we show the complete hyper-parameters in Tab. 7. As mentioned in Sec. 4, our method is implemented as a plugin to FixMatch [6]. Thus, we keep the original hyper-parameters in FixMatch and alert additional hyper-parameters in our method. Note that FixMatch sets different values of weight decay w for CIFAR-10 and CIFAR-100, which are 0.0005 and 0.001 respectively. For simplicity, we set $w = 0.0005$ for all experiments in our work. Additionally, the models in this paper are trained on GeForce RTX 3090/2080 Ti and Tesla V100. We observe that since no additional network components are introduced, the average running time of single iteration hardly increased, which means our method does not introduce excessive computational overhead.

Table 7: Complete list of hyper-parameters of PRG plugged in FixMatch [6]. N_B and α are additional hyper-parameters in our method whereas other hyper-parameters follow the setting of original FixMatch. Note that unlabeled data batch size B_U can be calculated by $B_U = \mu B$.

Hyper-parameter	Description	CIFAR-10	CIFAR-100	mini-ImageNet
μ	The ratio of unlabeled data to labeled data in a mini-batch	7		
B	Batch size for labeled data and class transition tracking	64		
B_U	Batch size for unlabeled data	448		
λ_U	Unlabeled loss weight	1		
τ	Confidence threshold	0.95		
lr	Start learning rate	0.03		
β	Momentum	0.9		
w	Weight decay	0.0005		
N_B	Tracked batch number	128		
α	Class invariance coefficient	1		

D. Additional Experimental Results

D.1. Using Distribution Alignment in MNAR

As discussed in Sec. 3.2, *distribution alignment* (DA) aims to perform strong regularization on pseudo-labels by aligning the class distribution of predictions on unlabeled data to that of labeled data. DA boosts the performance of SSL models tangibly [1, 2, 4, 6]. However, DA works on a strong assumption that the distribution of unlabeled data matches that of labeled data. In MNAR, this assumption does not hold obviously. Therefore, SSL methods that incorporate DA will face predicaments in MNAR. As shown in Tab. 8, rather than improving performance, integrating DA into SSL models is counterproductive, *e.g.*,

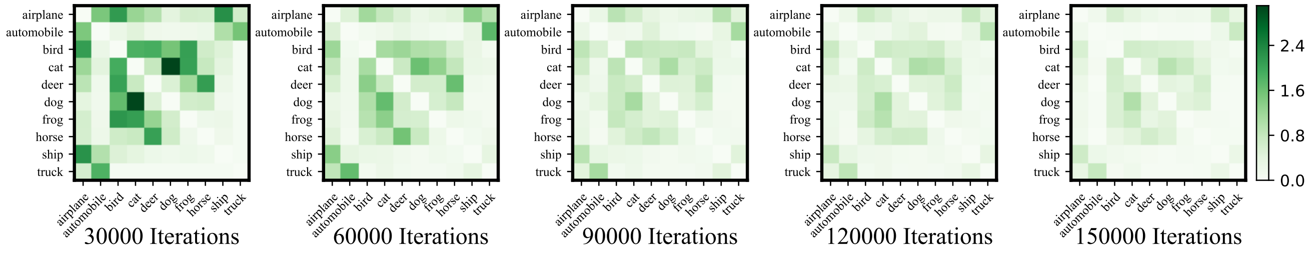

Figure 8: Visualization of class tracking matrix $\mathbf{C}$ obtained in training process of FixMatch [6] combining PRG. Experiments are conducted on CIFAR-10 with the same setting as in Fig. 4.

original FixMatch outperforms FixMatch with DA by up to 28.68% on CIFAR-10. Another example is SimMatch in Tab. 1. Despite SimMatch being a considerably more advanced method compared to FixMatch, its performance with PRG is weaker than that of FixMatch when a small value of γ is used, implying a small n_L. This underperformance can be attributed to its adoption of DA. As γ (implying n_L) increases, more supervisory information allows SimMatch's inherent strong performance begins to overshadow the negative impact of DA. Conversely, our method is not restricted by the mismatched distributions and achieves superior performance across the board, because PRG helps the model to better handle MNAR scenarios without any prior information (distribution prior estimated from labeled data is used in DA).

Table 8: Accuracy (%) in MNAR under our protocol compared with more baseline methods using distribution alignment (DA) [1]. Note that CoMatch [4] (a recently-proposed graph-based SSL method integrating contrastive learning) also combines DA to improve the quality of pseudo-labels in the conventional SSL setting.

Method	CIFAR-10 (n_L = 40)		CIFAR-10 (n_L = 250)		CIFAR-100 (n_L = 2500)		mini-ImageNet (n_L = 1000)	
	N_1 = 10	20	100	200	100	200	40	80
CoMatch	60.27	39.48	57.87	26.77	48.02	30.08	30.24	21.47
FixMatch	85.72	76.53	69.76	46.53	61.31	41.38	36.20	28.33
+ DA	$71.23^{\downarrow 14.49}$	$47.85^{\downarrow 28.68}$	$61.8^{\downarrow 7.96}$	$27.61^{\downarrow 18.92}$	$50.94^{\downarrow 10.37}$	$31.82^{\downarrow 9.56}$	$33.87^{\downarrow 2.33}$	$23.53^{\downarrow 4.78}$
+ PRG (Ours)	$\mathbf{91.87}^{\uparrow 6.15}$	$77.44^{\uparrow 0.91}$	$\mathbf{93.93}^{\uparrow 24.17}$	$\mathbf{67.86}^{\uparrow 21.33}$	$\mathbf{61.49}^{\uparrow 0.18}$	$\mathbf{49.84}^{\uparrow 8.46}$	$\mathbf{39.99}^{\uparrow 3.79}$	$\mathbf{35.39}^{\uparrow 7.069}$
+ PRG$^{\text{Last}}$ (Ours)	$85.66^{\downarrow 0.06}$	$\mathbf{77.85}^{\uparrow 1.32}$	$92.80^{\uparrow 23.04}$	$64.00^{\uparrow 17.47}$	$60.41^{\downarrow 0.90}$	$43.80^{\uparrow 2.42}$	$39.84^{\uparrow 3.64}$	$33.10^{\uparrow 4.77}$

D.2. Empirical Analysis on PRG

Different from Fig. 4, the color blocks in the heatmap in Fig. 8 almost cover the entire diagram, and some color blocks are not missing as the training progresses, *i.e.*, with the help of PRG, the information exchange between classes remains frequent during the learning process, and the model maintains the pseudo-rectifying ability for almost all classes.

D.3. More Evaluations on PRG

D.3.1 More MNAR Scenarios

We also provide more experiments on the setting of balanced labeled data with imbalanced unlabeled data, which is summarized in Tab. 9. For specific, we set $n_L = 40$ with balanced distribution and set $\gamma_u = 50, 100, 200$ for imbalanced unlabeled data, *i.e.*, the class-wise number of unlabeled data $M_i = M_1 \times \gamma_u^{-\frac{k-i}{k-1}}$, where $M_1 = 5000$ in CIFAR-10. As shown in Tab. 9, PRG outperforms all baseline methods by a large margin (the performance of CADR is even weaker than original FixMatch), proving the robustness of PRG in this MNAR scenario due to the unbiased guidance derived from the class transition history.

Table 9: Accuracy (%) on CIFAR-10 with $n_L = 40$ and various γ_u under our protocol.

Method	$\gamma_u = 20$	$\gamma_u = 50$	$\gamma_u = 100$
CoMatch	52.73	46.20	38.85
FixMatch	57.54	54.82	50.67
+ DA	$54.08^{\downarrow 3.46}$	$46.71^{\downarrow 8.11}$	$41.37^{\downarrow 9.30}$
+ CADR	$49.38^{\downarrow 8.16}$	$45.27^{\downarrow 9.55}$	$42.30^{\downarrow 8.37}$
+ PRG (Ours)	$\mathbf{62.43}^{\uparrow 4.90}$	$\mathbf{62.44}^{\uparrow 7.62}$	$\mathbf{58.23}^{\uparrow 7.56}$

Table 10: Class-wise precision and recall on CIFAR-10 during the training under CADR's protocol with $\gamma = 50$.

Method	Class Index	30000 Iterations		90000 Iterations		150000 Iterations	
		Precision	Recall	Precision	Recall	Precision	Recall
FixMatch	1	45.21	95.22	46.89	96.72	47.93	97.80
	2	49.12	99.01	49.59	99.27	50.27	98.72
	3	38.49	88.73	39.74	88.43	70.26	89.47
	4	75.02	68.13	75.63	72.19	82.04	75.93
	5	86.14	88.43	86.88	90.21	88.42	94.38
	6	89.45	62.93	91.03	64.4	89.31	75.98
	7	86.47	90.03	90.23	8.89	91.37	94.80
	8	89.09	75.94	90.48	75.21	95.32	75.37
	9	99.02	0.00	97.95	1.00	97.21	2.00
	10	0.00	0.00	99.60	0.33	98.60	0.67
+ PRG (Ours)	1	70.52	93.52	87.34	95.50	88.25	95.34
	2	82.53	98.21	96.03	98.32	96.78	98.56
	3	73.52	76.54	90.92	89.85	92.37	90.57
	4	70.21	73.77	85.36	80.51	87.89	81.37
	5	79.03	86.57	90.31	96.31	92.74	96.19
	6	74.55	61.03	90.58	79.88	90.97	82.43
	7	89.12	91.40	93.09	97.02	93.79	98.03
	8	92.58	80.14	95.01	96.21	96.32	97.50
	9	96.31	76.50	95.22	92.12	95.63	93.55
	10	96.56	62.52	96.95	96.01	97.15	96.81

Table 11: Accuracy (%) in MNAR under our protocol with more SSL learners.

Method	CIFAR-10 (n_L = 40)		CIFAR-10 (n_L = 250)		CIFAR-100 (n_L = 2500)		mini-ImageNet (n_L = 1000)	
	N_1 = 10	20	100	200	100	200	40	80
FlexMatch	90.86	84.53	79.13	55.40	61.49	45.26	39.45	34.18
+ PRG (Ours)	**92.17**$^{\uparrow 1.31}$	88.46$^{\uparrow 3.93}$	**93.95**$^{\uparrow 14.82}$	**69.88**$^{\uparrow 14.48}$	**65.29**$^{\uparrow 3.80}$	**50.31**$^{\uparrow 5.05}$	41.02$^{\uparrow 1.57}$	**36.59**$^{\uparrow 2.41}$
+ PRG$^{\text{Last}}$ (Ours)	91.03$^{\uparrow 0.17}$	**89.42**$^{\uparrow 4.89}$	92.94$^{\uparrow 13.81}$	67.07$^{\uparrow 11.67}$	64.66$^{\downarrow 3.17}$	48.82$^{\uparrow 3.56}$	**41.25**$^{\uparrow 1.80}$	35.16$^{\uparrow 0.98}$

D.3.2 More Metrics

To comprehensively explore the improvement of PRG in MNAR, we report the difference in class-wise precision and recall with/without PRG. The experimental results are shown in Tab. 10. Compared to original FixMatch, we witness FixMatch with PRG achieves higer precision/recall by and large, especially on rare classes (*i.e.*, class with larger index), which demonstrates that the bias removal capability of PRG effectively mitigates the effect of MNAR on the model. We also observe that both PRG and FixMatch achieve high precision as well as recall on popular classes and high precision but low recall on rare classes (especially FixMatch) in the early training period. The improvement of recall by PRG is due to the activated class transitions, which gives the model a certain probability to assign pseudo-labels to rare classes.

D.3.3 More SSL Learners

Moreover, to further evaluate PRG's performance, we consider building PRG on the top of more SSL frameworks. Thus, we firstly conduct experiments on CIFAR-10 under CADR's protocol with UPS [5] combining PRG. UPS is a recently-proposed uncertainty-aware pseudo-label selection framework for SSL, which is the SOTA method among pseudo-labeling based methods. We keep all training settings the same as the original UPS. With $\gamma = 20$, UPS achieves an accuracy of **30.46%** whereas UPS with PRG achieves an accuracy of **32.22%**. We note that UPS performs poorly in the MNAR scenarios because it is a more pure pseudo-labeling approach that does not introduce consistency regularization to improve model performance. Also we observe that PRG improves UPS marginally, much less than FixMatch. This is understandable because the negative learning that UPS prides itself on can be potentially negatively affected by the probability distribution of pseudo-label being adjusted by PRG, *e.g.*, uncertainty being altered. Next, we adopt a more advanced SSL learner FlexMatch [8] to evaluate PRG, which is shown in Tab. 11. PRG still complements the unrobustness of this strong SSL method in MNAR.

Table 12: Accuracy (%) in the conventional setting with various n_L. Results of baselines are reported in CADR [3] while results of * are based on our reimplementation.

Method	CIFAR-10			CIFAR-100			mini-ImageNet
	n_L = 40	250	4000	400	2500	10000	1000
FixMatch	$88.61_{\pm 3.35}$	$\mathbf{94.93}_{\pm 0.33}$	$95.69_{\pm 0.15}$	$50.05_{\pm 3.01}$	$\mathbf{71.36}_{\pm 0.24}$	$76.82_{\pm 0.11}$	$39.03_{\pm 0.66}{}^{*}$
+ CADR	$94.41^{\uparrow 5.80}$	$94.35^{\downarrow 0.58}$	$95.59^{\downarrow 0.10}$	$\mathbf{52.90}^{\uparrow 2.85}$	$70.61^{\downarrow 0.75}$	$76.93^{\uparrow 0.11}$	-
+ PRG (Ours)	$\mathbf{94.44}^{\uparrow 5.83}_{\pm 0.16}$	$94.42^{\downarrow 0.51}_{\pm 0.06}$	$95.38^{\downarrow 0.31}_{\pm 0.10}$	$52.45^{\uparrow 2.40}_{\pm 3.75}$	$70.12^{\downarrow 1.24}_{\pm 0.21}$	$76.49^{\downarrow 0.33}_{\pm 0.42}$	$47.34^{\uparrow 8.31}_{\pm 1.60}$
+ PRG$^{\text{Last}}$ (Ours)	$93.00^{\uparrow 4.39}_{\pm 0.79}$	$94.43^{\downarrow 0.50}_{\pm 0.33}$	$\mathbf{95.75}^{\uparrow 0.06}_{\pm 0.11}$	$48.81^{\downarrow 1.24}_{\pm 0.15}$	$70.01^{\downarrow 1.35}_{\pm 0.02}$	$\mathbf{77.12}^{\uparrow 0.30}_{\pm 0.13}$	$\mathbf{48.23}^{\uparrow 9.20}_{\pm 0.59}$

D.3.4 More Data Types

The results of VIME combined with PRG on tabular data are shown in Tab. 4. VIME [7] is a prevailing self- and semi-supervised learning frameworks for tabular data with pretext task of estimating mask vectors from corrupted tabular data. We implement PRG above the semi-supervised learning component of VIME. PRG provide pseudo-rectifying guidance to rescale the pseudo-labels for the original unlabeled sample in VIME. Specially, we replace the consistency loss used in VIME (*i.e.*, mean squared error in Eq. (9) in [7]) with standard cross-entropy loss to makes PRG applicable to VIME.

D.3.5 Coventional SSL Setting

As shown in Tab. 12, our method still works well in the conventional SSL setting, *i.e.*, both the labeled data and the unlabeled data are *balanced*. The class-level guidance offered by our method is also valid in the conventional setting while maintaining the vitality of class transition, even though there is not too much need to remove bias on label imputation.

References

[1] 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*, 2020. 2, 3

[2] Chengyue Gong, Dilin Wang, and Qiang Liu. Alphamatch: Improving consistency for semi-supervised learning with alpha-divergence. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 2

[3] Xinting Hu, Yulei Niu, Chunyan Miao, Xian-Sheng Hua, and Hanwang Zhang. On non-random missing labels in semi-supervised learning. In *International Conference on Learning Representations*, 2022. 5

[4] Junnan Li, Caiming Xiong, and Steven CH Hoi. Comatch: Semi-supervised learning with contrastive graph regularization. In *IEEE/CVF International Conference on Computer Vision*, 2021. 2, 3

[5] Mamshad Nayeem Rizve, Kevin Duarte, Yogesh S Rawat, and Mubarak Shah. In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. In *International Conference on Learning Representations*, 2021. 4

[6] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems*, 2020. 2, 3

[7] Jinsung Yoon, Yao Zhang, James Jordon, and Mihaela van der Schaar. Vime: Extending the success of self-and semi-supervised learning to tabular domain. In *Advances in Neural Information Processing Systems*, 2020. 5

[8] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. In *Advances in Neural Information Processing Systems*, 2021. 4