CLASS-IMBALANCED SEMI-SUPERVISED LEARNING

Minsung Hyun, Jisoo Jeong, Nojun Kwak *
Seoul National University, Seoul, Korea
{minsung.hyun,soo3553,nojunk}@snu.ac.kr

ABSTRACT

Semi-Supervised Learning has achieved great success in overcoming the difficulties of labeling and making full use of unlabeled data. However, Semi-Supervised Learning has a limited assumption that the numbers of samples in different classes are balanced, and many Semi-Supervised Learning algorithms show lower performance for the datasets with the imbalanced class distribution. In this paper, we introduce a task of Class-Imbalanced Semi-Supervised Learning, which refers to semi-supervised learning with class-imbalanced data. In doing so, we consider class imbalance in both labeled and unlabeled sets. We propose Suppressed Consistency Loss, a regularization method robust to class imbalance. Our method shows better performance than the conventional methods in the Class-Imbalanced Semi-Supervised Learning task. In particular, the more severe the class imbalance and the smaller the size of the labeled data, the better our method performs.

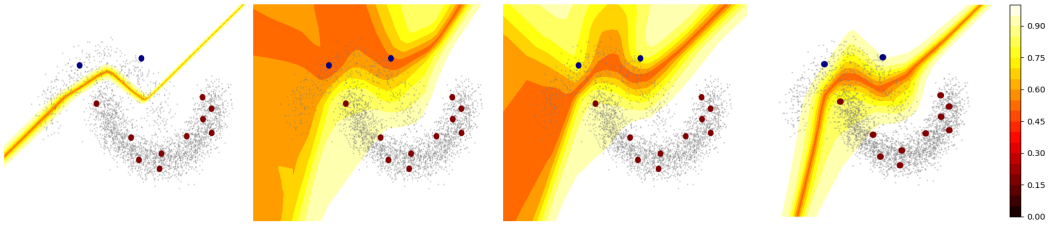
1 INTRODUCTION

A large dataset with well-refined annotations is essential to the success of deep learning and every time we encounter a new problem, we need to annotate the whole dataset, which costs a lot of time and effort (Russakovsky et al., 2015; Bearman et al., 2016). To alleviate this annotation burden, many researchers have studied semi-supervised learning (SSL) that improves the performance of models by utilizing the information contained in unlabeled data (Chapelle et al., 2009; Verma et al., 2019; Berthelot et al., 2019b). However, SSL has a couple of main assumptions and shows excellent performance only in these limited settings. The first assumption is that unlabeled data is in-distribution (Oliver et al., 2018) and the second is the assumption of balanced class distribution (Li et al., 2011; Stanescu & Caragea, 2014). In this paper, we performed a study dealing with the second assumption.

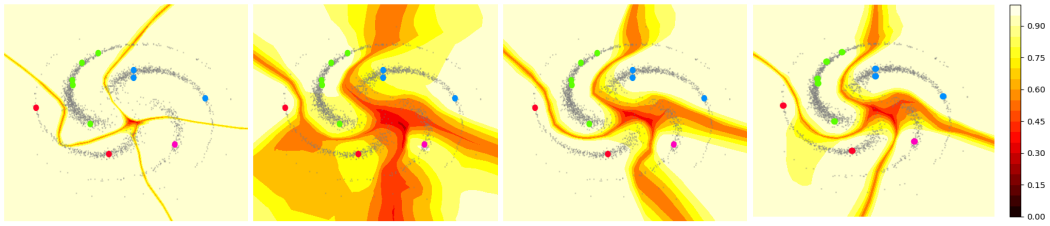
The class distribution of data, in reality, is not refined and is known to have long tails (Kendall et al., 1946). However, many researches have developed models based on well-refined balanced data (Krizhevsky et al., 2009; Netzer et al., 2011; Deng et al., 2009). Training the model with imbalanced datasets causes performance degradation. Class imbalanced learning (CIL) is a way to solve such class imbalance and proposes various methods in the level of data, algorithm, and their hybrids (Krawczyk, 2016; Johnson & Khoshgoftaar, 2019). However, to our best knowledge, most of the studies on CIL have relied on labeled datasets for training and have not considered the use of unlabeled data. In this paper, we define a task, *class-imbalanced semi-supervised learning* (CISSL), and propose a suitable algorithm for it. By assuming class imbalance in both labeled and unlabeled data, CISSL relaxes the assumption of balanced class distribution in SSL. Also, it can be considered as a task of adding unlabeled data to CIL. We propose a regularization method using *'suppressed consistency loss'* (SCL), for better performance in the CISSL settings. SCL prohibits the decision boundary in a minor class region from being smoothed too much in the wrong direction as shown in Fig.1d, 1h.

Our main contributions can be summarized as follows:
• We defined a task of imbalanced semi-supervised learning, reflecting a more realistic situation, and suggested standard experimental settings.
• We analyzed how the existing SSL methods work in CISSL settings.
• We proposed Suppressed Consistency Loss that works robustly for problems with class imbalance, and experimentally show that our method improves performance.

*corresponding author



(a) Two moons Supervised (b) Two moons Π model (c) Two moons Mean Teacher (d) Two moons SCL (ours)



(e) Four spins Supervised (f) Four spins Π model (g) Four spins Mean Teacher (h) Four spins SCL (ours)

Figure 1: Toy examples: We experimented on *Two moons* and *Four spins* datasets in CISSL settings for four algorithms (Supervised learning, Π model (Laine & Aila, 2016), Mean Teacher (Tarvainen & Valpola, 2017) and SCL (ours)). The color represents the probability of of the class with the highest confidence.

2 Analysis of SSL under Class Imbalance

In this section, we look into the topography of the decision boundary to see how the SSL algorithms work in the class-imbalanced task. We compare supervised learning with SSL's representative algorithms, Π model (Laine & Aila, 2016) and Mean Teacher (Tarvainen & Valpola, 2017) via toy examples. And we analyze why MT performs better in CISSL through a mathematical approach in Appendix B.

Table 1: Mean and standard deviation of validation error rates (%) for all, major, and minor classes in toy examples. We conducted 5 runs with different random seeds for class imbalance distribution.

(%)	Class Type	Supervised	Π Model	Mean Teacher	MT+SCL (Ours)
Two moons	All	25.06 ± 12.43	41.57 ± 8.82	34.99 ± 9.98	**24.39 ± 15.14**
	Major	0.95 ± 1.24	**0.00 ± 0.00**	0.01 ± 0.03	0.06 ± 0.07
	Minor	49.17 ± 24.74	83.14 ± 17.64	69.96 ± 19.98	**48.01 ± 31.04**
Four spins	All	19.70 ± 6.70	17.79 ± 8.39	14.99 ± 8.46	**10.91 ± 8.94**
	Major	7.83 ± 5.43	**4.75 ± 3.74**	4.76 ± 3.30	6.28 ± 3.26
	Minor	49.39 ± 25.61	52.68 ± 31.17	43.29 ± 31.53	**27.68 ± 36.48**

We trained each algorithm by 5,000 iterations on *two moons* and *four spins* datasets with an imbalance factor of 5 for each labeled and unlabeled data. [1] We described experimental details in Appendix C.1. Fig.1 represents the highest confidence at each location. The region with relatively low probability, closer to the dark red color, is the decision boundary in the figure. In Fig.1a, 1e, the decision boundary of the supervised learning is very steep. And there are very high confidence areas far away from the decision boundary. With the SSL methods, unlabeled data smooth the decision boundary through consistency regularization (Chapelle et al., 2009). In particular, the decision boundary smoothing is larger in the minor class area. Also, we found that the learning patterns of the Π model and MT are different. Table.1 shows the validation error rates where performance degradation is evident in the minor class. MT shows relatively better performance than Π model, although it shows inferior performance than the supervised learning in *two moons*. We described more details on this approach in Appendix B. Our method which applies SCL to MT achieves the best performance in both *two moons* and *four spins* datasets.

[1] The number of samples of class c is set to $N_c = N_{max} \times \rho^{-\frac{R_c - 1}{C-1}}$. The *Rank*, R_c, of the major class is 1. ρ and C are the imbalance ratio and the number of classes.

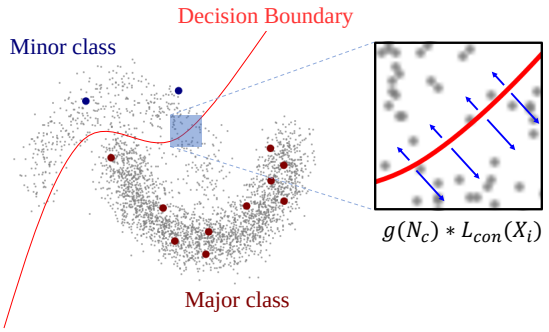


Figure 2: Suppressed Consistency Loss (SCL). Due to the imbalance in data, decision boundary tends to skew into the areas of minor class with consistency regularization. SCL inversely weights consistency loss to the number of class samples and pushes the decision boundary against low-density areas.

3 Suppressed Consistency Loss

In Section 2, we found that performance degradation of SSL models in CISSL is due to consistency regularization in minor classes. With the intuition that we should suppress the consistency regularization of minor classes in CISSL, we propose a new loss term, *suppressed consistency loss* (SCL), as follows:

$$\begin{aligned} L_{SCL}(X_i) &= g(N_c) * L_{con}(X_i), \\ \text{where} \quad c &= \text{argmax}(f_\theta(X_i)). \end{aligned} \tag{1}$$

Here, $g(z)$ can be any function proportional to z and we set it as

$$g(z) = \beta^{1-\frac{z}{N_{max}}}, \tag{2}$$

where $\beta \in (0, 1]$. N_c is the number of training samples of the class predicted by the model, N_{max} is the number of samples of the class with the most frequency. SCL weights the consistency loss in an exponentially proportional to the number of samples in a class. In (1), $g(N_c)$ is 1 for the most frequent class, where it works the same as the conventional consistency loss. For the least frequent class, the influence of the consistency loss is suppressed.

Benefits of using SCL instead of conventional consistency loss is as follows: We can consider the consistency loss as imposing a small ball $B(X)$ centered at X, making $f_\theta(B(X))$ as constant as possible. If we apply consistency loss to the sampled points, due to the sparsity of labeled data in the minor class, $f_\theta(X)$ will be more influenced by this smoothing effect in the minor class areas while the effect is smaller for the major class which has densely labeled data. Because of this, near the decision boundary, $f(X)$ of the major class propagates to the minor class as shown in Fig.1b,1c,1f,1g. Instead, if we apply SCL, as consistency loss is suppressed in the negative area, smoothing does not propagate to the minor class area much. Fig.2 illustrates the effect of consistency regularization by SCL. When the model predicts actual minor class samples as minor one, the SCL limits the smoothing of the decision boundary towards the minor class cluster. On the other hand, when the model predicts actual minor class samples as a major class in the high-density area of the minor class, the decision boundary is smoothed with higher weight. Consequentially, SCL pushes the decision boundary to low-density areas of the minor class and prevents performance degradation, as shown in the square box of Fig.2.

4 Experiments

In our main experiments, we split the training set into the labeled set and the unlabeled set. The size of the unlabeled set changes depending on unlabeled data imbalance types because of the limitation of the training dataset. We divided the training dataset into three parts: labeled dataset, unlabeled dataset, and validation dataset. Labeled data is configured to have an imbalance for each class according to the CIL task. We have experimented with various numbers of imbalance factors (ρ), the

Table 2: Test error rates (%) and standard deviation from experiments with 4k number of labeled data and imbalance factor {10, 20, 50, 100 } under 3 different unlabeled imbalance types in CIFAR10 and SVHN. VAT+EM refers to Virtual Adversarial Training with Entropy Minimization. (**Bold/Red/Blue**: **supervised**, **best** and **second best** results for each column.)

Dataset		CIFAR 10				SVHN			
Imbalance factor(ρ_l)		10	20	50	100	10	20	50	100
Supervised	Uniform ($\rho_u = 1$)	**23.03 ± 1.65**	**27.49 ± 1.87**	**33.15 ± 2.83**	**36.71 ± 2.79**	**18.49 ± 1.90**	**21.92 ± 2.28**	**30.03 ± 3.83**	**35.89 ± 6.39**
Π-Model (Laine & Aila, 2016)		21.10 ± 1.93	25.74 ± 3.82	33.91 ± 3.49	39.36 ± 4.47	11.74 ± 1.80	13.42 ± 2.14	21.63 ± 4.58	28.59 ± 7.90
MT (Tarvainen & Valpola, 2017)		**16.45 ± 1.24**	**19.25 ± 1.99**	**23.45 ± 3.30**	**29.06 ± 5.13**	**6.52 ± 0.55**	**6.75 ± 0.49**	**7.60 ± 1.85**	**8.94 ± 2.12**
VAT + em (Miyato et al., 2018)		17.93 ± 2.12	20.18 ± 3.18	30.43 ± 6.18	36.57 ± 7.20	6.81 ± 0.30	7.70 ± 0.87	13.84 ± 6.17	29.15 ± 4.80
VAT + em + SNTG (Luo et al., 2018)		18.15 ± 2.25	20.39 ± 2.46	29.77 ± 6.71	36.34 ± 6.54	93.30 ± 0.00	93.30 ± 0.00	14.88 ± 5.38	93.30 ± 0.00
Pseudo-Label (Lee, 2013)		19.33 ± 1.36	24.34 ± 4.06	34.18 ± 4.23	39.59 ± 5.70	10.15 ± 0.87	9.97 ± 1.45	16.00 ± 4.34	32.79 ± 7.62
ICT (Verma et al., 2019)		18.01 ± 1.28	20.52 ± 1.91	30.18 ± 2.63	38.33 ± 4.72	27.82 ± 5.12	37.75 ± 7.50	58.20 ± 9.38	67.02 ± 12.66
MT+SCL (ours)		**15.65 ± 0.69**	**16.99 ± 1.31**	**19.95 ± 2.36**	**22.62 ± 3.54**	**6.52 ± 0.53**	**7.11 ± 0.30**	**7.70 ± 0.73**	**8.56 ± 0.86**
Supervised	Half ($\rho_u = \rho_l/2$)	**23.03 ± 1.65**	**27.49 ± 1.87**	**33.15 ± 2.83**	**36.71 ± 2.79**	**18.49 ± 1.90**	**21.92 ± 2.28**	**30.03 ± 3.83**	**35.89 ± 6.39**
Π-Model (Laine & Aila, 2016)		22.69 ± 1.99	27.72 ± 4.17	33.96 ± 3.19	38.84 ± 4.17	12.96 ± 1.26	16.70 ± 4.01	24.02 ± 3.97	33.73 ± 7.52
MT (Tarvainen & Valpola, 2017)		**19.48 ± 1.96**	**23.30 ± 2.85**	**30.06 ± 3.92**	**35.37 ± 3.52**	**7.25 ± 0.38**	**8.85 ± 1.10**	**12.19 ± 1.68**	**17.23 ± 2.44**
VAT + em (Miyato et al., 2018)		20.17 ± 2.49	24.50 ± 2.88	32.54 ± 4.61	36.77 ± 3.75	8.99 ± 1.21	11.59 ± 1.85	18.95 ± 4.49	30.44 ± 6.95
VAT + em + SNTG (Luo et al., 2018)		20.41 ± 2.47	24.64 ± 2.79	32.56 ± 4.05	38.48 ± 3.87	93.30 ± 0.00	93.30 ± 0.00	20.60 ± 5.73	93.30 ± 0.00
Pseudo-Label (Lee, 2013)		21.23 ± 2.52	26.78 ± 3.41	34.12 ± 4.51	39.72 ± 4.20	11.59 ± 1.96	13.97 ± 2.11	24.40 ± 4.46	33.70 ± 6.89
ICT (Verma et al., 2019)		19.53 ± 1.41	23.90 ± 2.07	31.09 ± 3.35	37.36 ± 2.02	22.38 ± 7.89	38.12 ± 6.57	48.88 ± 8.33	58.99 ± 7.35
MT+SCL (ours)		**17.36 ± 1.17**	**21.74 ± 2.15**	**28.20 ± 3.09**	**33.09 ± 3.63**	**7.54 ± 0.50**	**9.29 ± 1.48**	**11.46 ± 1.21**	**18.63 ± 3.97**
Supervised	Same ($\rho_u = \rho_l$)	**23.03 ± 1.65**	**27.49 ± 1.87**	**33.15 ± 2.83**	**36.71 ± 2.79**	**18.49 ± 1.90**	**21.92 ± 2.28**	**30.03 ± 3.83**	**35.89 ± 6.39**
Π-Model (Laine & Aila, 2016)		23.49 ± 2.69	28.18 ± 3.31	34.22 ± 3.19	38.05 ± 3.19	13.46 ± 2.13	17.13 ± 2.61	26.53 ± 3.43	33.71 ± 8.17
MT (Tarvainen & Valpola, 2017)		20.50 ± 2.58	**24.67 ± 2.60**	**31.77 ± 3.79**	**35.91 ± 3.70**	**8.62 ± 1.29**	**9.29 ± 1.41**	**15.16 ± 3.54**	**21.01 ± 4.14**
VAT + em (Miyato et al., 2018)		21.45 ± 1.88	25.83 ± 3.21	33.13 ± 3.67	37.67 ± 2.20	10.39 ± 0.96	13.62 ± 2.00	21.49 ± 5.27	32.39 ± 8.25
VAT + em + SNTG (Luo et al., 2018)		21.87 ± 2.65	26.49 ± 3.07	33.36 ± 3.86	38.48 ± 2.96	93.30 ± 0.00	93.30 ± 0.00	23.52 ± 7.34	93.30 ± 0.00
Pseudo-Label (Lee, 2013)		22.73 ± 2.74	27.50 ± 3.39	34.91 ± 2.57	38.69 ± 4.28	12.34 ± 1.79	15.93 ± 2.43	25.66 ± 5.95	33.53 ± 8.08
ICT (Verma et al., 2019)		**19.96 ± 1.05**	25.63 ± 1.91	33.56 ± 3.14	36.85 ± 3.44	24.53 ± 12.62	37.25 ± 8.22	49.85 ± 7.74	56.97 ± 10.28
MT+SCL (ours)		**18.69 ± 2.09**	**22.98 ± 2.33**	**29.76 ± 2.40**	**34.22 ± 3.50**	**8.22 ± 0.89**	**10.04 ± 0.82**	**15.48 ± 2.29**	**20.39 ± 4.10**

ratio between the numbers of samples of the most frequent and the least frequent classes. We considered three types of class imbalance in unlabeled data: *Same* ($\rho_u = \rho_l$, where ρ_l and ρ_u are the imbalance factors for labeled and unlabeled dataset.), *Uniform* (uniform distribution, $\rho_u = 1$), and *Half* ($\rho_u = \rho_l/2$). The size of the unlabeled dataset changes depending on unlabeled data imbalance types because of the limitation of the dataset used. For fair experiments, we set the size of the unlabeled set based on the *Same* case, which uses the lowest number of unlabeled samples. Validation data is made up as in (Oliver et al., 2018).

We experimented with changing the imbalance factor while keeping the number of labeled samples. We experimented on CIFAR-10 and SVHN with imbalance factor $\rho_l \in \{10, 20, 50, 100\}$. The results are shown in Table 2. In CIFAR-10, the left part of Table.2, the higher the imbalance factor, the lower the performance. Supervised learning on imbalance factor 100 achieves 36.71% error, which $13\%p$ higher than supervised learning on imbalance factor 10. With the small imbalance factor, SSL algorithms generally improve performance although unlabeled data has same imbalance with labeled data. As the imbalance factor increases, on the other hand, some SSL algorithms show lower performance than supervised learning. Mean Teacher is the only SSL algorithm that improves the performance with imbalance factor 100 in *Same* case. This means that general SSL algorithms are vulnerable to the class imbalance. However, the proposed SCL has robustly improved the performance in various imbalance settings. Notably, it shows remarkable improvement in the *Uniform* case compared to SSL algorithms. SVHN, the right part of Table.2, shows similar results. However, there is no big performance difference between MT and our method. This is because SVHN is easier to classify than CIFAR10. For SVHN, SNTG and ICT show lower performance than the supervised learning. It seems that the model training fails.

5 Conclusion

The existing SSL methods do not perform well under the CISSL environment not considering data imbalance. This fact gives us some implications. First, for deep learning to become a practical application, we need to work on a harsher benchmark. We experimented on datasets which relaxed the equal class distribution assumption of SSL, and our method yielded meaningful results. Second, we should avoid developing domain-specific algorithms which work very well only under certain conditions. Finally, we need to focus not only on the performance improvement of a model but also on its causes. An in-depth analysis of the causes of the phenomena provides an intuition about the direction of future research.

ACKNOWLEDGMENTS

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (2021R1A2C3006659).

REFERENCES

Shin Ando and Chun Yuan Huang. Deep over-sampling framework for classifying imbalanced data. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 770–785. Springer, 2017.

Philip Bachman, Ouais Alsharif, and Doina Precup. Learning with pseudo-ensembles. In *Advances in Neural Information Processing Systems*, pp. 3365–3373, 2014.

Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. What's the point: Semantic segmentation with point supervision. In *European conference on computer vision*, pp. 549–565. Springer, 2016.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48, 2009.

David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *arXiv preprint arXiv:1911.09785*, 2019a.

David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. *arXiv preprint arXiv:1905.02249*, 2019b.

Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259, 2018.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *arXiv preprint arXiv:1906.07413*, 2019.

Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9268–9277, 2019.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Qi Dong, Shaogang Gong, and Xiatian Zhu. Imbalanced deep learning by minority class incremental rectification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(6):1367–1381, Jun 2019. ISSN 1939-3539. doi: 10.1109/tpami.2018.2832629. URL http://dx.doi.org/10.1109/TPAMI.2018.2832629.

Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5375–5384, 2016.

Jisoo Jeong, Seungeui Lee, Jeesoo Kim, and Nojun Kwak. Consistency-based semi-supervised learning for object detection. In *Advances in Neural Information Processing Systems*, pp. 10758–10767, 2019.

Justin M Johnson and Taghi M Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1):27, 2019.

Maurice George Kendall et al. The advanced theory of statistics. *The advanced theory of statistics.*, (2nd Ed), 1946.

Salman H Khan, Munawar Hayat, Mohammed Bennamoun, Ferdous A Sohel, and Roberto Togneri. Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE transactions on neural networks and learning systems*, 29(8):3573–3587, 2017.

Jaehyung Kim, Youngbum Hur, Sejun Park, Eunho Yang, Sung Ju Hwang, and Jinwoo Shin. Distribution aligning refinery of pseudo-label for imbalanced semi-supervised learning. *arXiv preprint arXiv:2007.08844*, 2020.

Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in neural information processing systems*, pp. 3581–3589, 2014.

Bartosz Krawczyk. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232, 2016.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

Abhishek Kumar, Prasanna Sattigeri, and Tom Fletcher. Semi-supervised learning with gans: Manifold invariance with improved inference. In *Advances in neural information processing systems*, pp. 5534–5544, 2017.

Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

Bruno Lecouat, Chuan-Sheng Foo, Houssam Zenati, and Vijay Chandrasekhar. Manifold regularization with gans for semi-supervised learning. *arXiv preprint arXiv:1807.04307*, 2018.

Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, ICML*, volume 3, pp. 2, 2013.

Hansang Lee, Minseok Park, and Junmo Kim. Plankton classification on imbalanced large scale database via convolutional neural networks with transfer learning. In *2016 IEEE international conference on image processing (ICIP)*, pp. 3713–3717. IEEE, 2016.

Shoushan Li, Zhongqing Wang, Guodong Zhou, and Sophia Yat Mei Lee. Semi-supervised learning for imbalanced sentiment classification. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pp. 21–37. Springer, 2016.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Yucen Luo, Jun Zhu, Mengxi Li, Yong Ren, and Bo Zhang. Smooth neighbors on teacher graphs for semi-supervised learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8896–8905, 2018.

Lars Maaløe, Casper Kaae Sønderby, Søren Kaae Sønderby, and Ole Winther. Auxiliary deep generative models. *arXiv preprint arXiv:1602.05473*, 2016.

David Masko and Paulina Hensman. The impact of imbalanced training data for convolutional neural networks, 2015.

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.

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

Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in Neural Information Processing Systems*, pp. 3235–3246, 2018.

Samira Pouyanfar, Yudong Tao, Anup Mohan, Haiman Tian, Ahmed S Kaseb, Kent Gauen, Ryan Dailey, Sarah Aghajanzadeh, Yung-Hsiang Lu, Shu-Ching Chen, et al. Dynamic sampling in convolutional neural networks for imbalanced data classification. In *2018 IEEE conference on multimedia information processing and retrieval (MIPR)*, pp. 112–117. IEEE, 2018.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Olga Russakovsky, Li-Jia Li, and Li Fei-Fei. Best of both worlds: human-machine collaboration for object annotation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2121–2131, 2015.

Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Advances in Neural Information Processing Systems*, pp. 1163–1171, 2016.

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. *arXiv preprint arXiv:2001.07685*, 2020.

Ana Stanescu and Doina Caragea. Semi-supervised self-training approaches for imbalanced splice site datasets. In *Proceedings of The Sixth International Conference on Bioinformatics and Computational Biology, BICoB 2014*, pp. 131–136, 2014.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*, pp. 1195–1204, 2017.

Vikas Verma, Alex Lamb, Juho Kannala, Yoshua Bengio, and David Lopez-Paz. Interpolation consistency training for semi-supervised learning. *arXiv preprint arXiv:1903.03825*, 2019.

Haishuai Wang, Zhicheng Cui, Yixin Chen, Michael Avidan, Arbi Ben Abdallah, and Alexander Kronzer. Predicting hospital readmission via cost-sensitive deep learning. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 15(6):1968–1978, 2018.

Shoujin Wang, Wei Liu, Jia Wu, Longbing Cao, Qinxue Meng, and Paul J Kennedy. Training deep neural networks on imbalanced data sets. In *2016 international joint conference on neural networks (IJCNN)*, pp. 4368–4374. IEEE, 2016.

Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. In *Advances in Neural Information Processing Systems*, pp. 7029–7039, 2017.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. *arXiv preprint arXiv:1905.03670*, 2019.

Chong Zhang, Kay Chen Tan, and Ruoxu Ren. Training cost-sensitive deep belief networks on imbalance data problems. In *2016 international joint conference on neural networks (IJCNN)*, pp. 4362–4367. IEEE, 2016.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

A Related Work

A.1 Semi-Supervised Learning

Semi-supervised learning is a learning method that tries to improve the performance of supervised learning, which is based only on labeled data ($\mathcal{D}_\mathcal{L}$), by additional usage of unlabeled data ($\mathcal{D}_\mathcal{U}$).

Numerous researches tried to utilize unlabeled data in diverse ways. The self-training-based SSL method gradually utilizes unlabeled data for learning, similar to curriculum learning (Bengio et al., 2009). The Pseudo Label (Lee, 2013), as the name implies, assumes we know the ground truth label of unlabeled samples, whose highest confidence probability from the output of the network exceed the given threshold. S^4L (Zhai et al., 2019) generates an arbitrary problem like image rotation and learns features by learning the problem. Another approach to solving the semi-supervised learning problem is using generative models (Goodfellow et al., 2014; Kingma et al., 2014; Radford et al., 2015; Maaløe et al., 2016; Dumoulin et al., 2016; Kumar et al., 2017; Lecouat et al., 2018). The GAN (Goodfellow et al., 2014) framework learns a discriminator classifying real and fake images and a generator that tries to create real-like images. In addition to this, in generative-model-based SSL, the discriminator classifies the label of samples (Kingma et al., 2014). While training the discriminator and the generator with unlabeled data, the model learns latent features that assist the SSL problem's model performance.

Above this, consistency regularization has shown good performance in semi-supervised learning, which pushes the decision boundary to low-density areas using unlabeled data (Bachman et al., 2014; Sajjadi et al., 2016; Laine & Aila, 2016; Verma et al., 2019). The objective function $\mathcal{J}$ is composed of supervised loss, L_{sup}, for $\mathcal{D}_\mathcal{L}$ and consistency regularization loss, L_{con}, for $\mathcal{D}_\mathcal{U}$. As a typical semi-supervised learning method (Laine & Aila, 2016; Oliver et al., 2018), ramp-up scheduling function $w(t)$ is used for stable training:

$$\mathcal{J} = L_{sup} + w(t) \cdot L_{con} \tag{3}$$

$$L_{con}(X) = d(f_\theta(X + \epsilon), f_{\theta_{tg}}(X + \epsilon')), \tag{4}$$

where d is a distance metric such as L_2 distance or KL-divergence, ϵ and ϵ' are perturbations to input data, and θ and θ_{tg} are the parameters of the model and target model, respectively. For C-class classification problem, $f_\theta(X) \in \mathbb{R}^C_+$ is the output logit (class probability) for the input X. Π model (Laine & Aila, 2016) and Mean Teacher (MT) (Tarvainen & Valpola, 2017) are the representative algorithms using consistency regularization. The Π model uses θ as θ_{tg} and MT updates θ_{tg} with EMA (exponential moving average) as follows:

$$\theta_{tg} \leftarrow \gamma\theta_{tg} + (1 - \gamma)\theta. \tag{5}$$

From (5), MT can be considered as a temporal ensemble model in the parameter space.

There are some other methods derived from consistency regularization (Berthelot et al., 2019b;a; Sohn et al., 2020). Virtual Adversarial Training (VAT) (Miyato et al., 2018) optimizes the direction of perturbation (ϵ). SNTG (Luo et al., 2018) suggests an additional regularization loss, which gathers same class samples together and pushes the samples of different classes to the outside of the margin. Interpolation Consistency Training (ICT)(Verma et al., 2019) fused consistency regularization with mixup (Zhang et al., 2017). Similar to our study, Kim et al. (2020) recently proposed a pseudo-label-based method to improve SSL classification performance under class imbalance.

In addition, the consistency-based semi-supervised learning for object detection (CSD) (Jeong et al., 2019) is an algorithm that applies consistency regularization to semi-supervised object detection problem. In a typical object detection algorithm, the model learns through classification loss and localization loss. Reflecting the characteristics of the object detection problem, CSD applies consistency regularization to each loss by perturbing images with the horizontal flip.

A.2 Class Imbalanced Learning

Class imbalanced learning is a way to alleviate the performance degradation due to class imbalance. Buda et al. (2018) defined the class imbalance factor ρ as the ratio between the numbers of samples of the most frequent and the least frequent classes. And we call each class as major class and minor class.

So far, there has been some research trying to address the class imbalance problems (Johnson & Khoshgoftaar, 2019). Data-level methods approach the problem by over-sampling minor classes or under-sampling major classes (Masko & Hensman, 2015; Lee et al., 2016; Pouyanfar et al., 2018; Buda et al., 2018). Algorithm-level methods re-weight the loss or propose a new loss without touching the sampling scheme (Wang et al., 2016; Lin et al., 2017; Wang et al., 2018; Khan et al., 2017; Zhang et al., 2016; Wang et al., 2017; Cui et al., 2019; Cao et al., 2019). Compared to data-level methods, algorithm-level methods can be easily applied without affecting training time. Because data-level methods require additional computation time in model training due to re-sampling. There are also hybrids of both methods (Huang et al., 2016; Ando & Huang, 2017; Dong et al., 2019). We applied three algorithm-level methods to the CISSL task and compared their performance to cross-entropy loss (CE).:

(i) Normalized weights, which weight a loss inversely proportional to the class frequency (IN) (Cao et al., 2019).

(ii) Focal loss which modulates by putting fewer weights on samples that the model is easy to classify (Lin et al., 2017).

(iii) Class-balanced loss which re-weights the loss in inverse proportion to the effective number of classes (CB) (Cui et al., 2019).

B Π MODEL VS. MEAN TEACHER

We analyze the results of Section 2 in this part. When the consistency regularization is applied to supervised learning in Fig.1a, 1e, compared to the samples far away from the boundary, the influence of the samples around the decision boundary is considerable. This is because the model output does not change even if small perturbation is added to the model input in the region far from the decision boundary from (4). As a result, consistency regularization smooths the decision boundary, as shown in Fig.1b, 1f.

According to the *cluster assumption* (Chapelle et al., 2009), the decision boundary lies in the low-density area and far from the high-density area. However, in a problem with severe class imbalance, the decision boundary may penetrate a globally sparse but relatively high-density area of a minor class as shown in the square in Fig.2. By consistency regularization, decision boundary smoothing occurs in this area, and many samples in the minor class are misclassified.

Therefore, conventional consistency regularization-based methods are generally expected to degrade the performance for the minor class. But we found that the severity of this phenomenon differs depending on the SSL algorithm. In Table.1, MT consistently performed better than Π model, especially for the minor class.

First, we analyzed the behavior of MT in CISSL with the simple SGD optimizer. Consider the model parameter θ, the learning rate α, and the objective function $\mathcal{J}$, then the update rule of SGD optimizer is:

$$\theta \leftarrow \theta - \alpha \nabla \mathcal{J}(\theta). \tag{6}$$

For a EMA decay factor of MT, $\gamma \in (0, 1]$, the current (θ) and the target (θ') model parameters at the t-th iteration are

$$\theta_t = \theta_0 - \alpha \sum_{k=0}^{t-1} \nabla \mathcal{J}(\theta_k), \tag{7}$$

$$\theta'_t = \theta_0 - \alpha \sum_{k=0}^{t-1} (1 - \gamma^{t-k-1}) \nabla \mathcal{J}(\theta_k). \tag{8}$$

Comparing (7) and (8), we can see that θ', the target for the consistency loss in MT, is updated slower than the model parameter θ because of the use of the EMA decay factor γ. On the other hand, in Π model, because $\theta' = \theta$, the target is updated faster than that of MT As described in the supplementary, we can get the same results of slow target update in MT for the SGD with momentum case that we used for our experiments.

Now we will check why MT performs better than Π model in CISSL task. Assume θ^{Π} and θ^{MT} be initially with the same value θ. In this case, the consistency loss of Π model and MT are

$$\begin{aligned} \Pi \text{ model} : L_{con}^{\Pi}(\theta) &= d(f_{\theta}(X+\epsilon), f_{\theta'=\theta}(X+\epsilon')) \\ \text{MT} : L_{con}^{MT}(\theta) &= d(f_{\theta}(X+\epsilon), f_{\theta'}(X+\epsilon')). \end{aligned} \tag{9}$$

If we use L_2 distance for d for simplicity, their derivatives become

$$\begin{aligned} \nabla_{\theta} L_{con}^{\Pi} &= \nabla_{\theta} \frac{1}{2} [f_{\theta}(X+\epsilon) - f_{\theta}(X+\epsilon')]^2 \\ &= [f_{\theta}(X+\epsilon) - f_{\theta}(X+\epsilon')] \nabla_{\theta} f_{\theta}(X+\epsilon), \end{aligned} \tag{10}$$

$$\begin{aligned} \nabla_{\theta} L_{con}^{MT} &= \nabla_{\theta} \frac{1}{2} [f_{\theta}(X+\epsilon) - f_{\theta'}(X+\epsilon')]^2 \\ &= [f_{\theta}(X+\epsilon) - f_{\theta'}(X+\epsilon')] \nabla_{\theta} f_{\theta}(X+\epsilon). \end{aligned} \tag{11}$$

Note the target parameters (θ') in (9) are not included in the gradient calculation. Using the Taylor series expansion $f_{\theta'}(X+\epsilon') \simeq f_{\theta}(X+\epsilon') + (\theta'-\theta)^T \nabla_{\theta} f_{\theta}(X+\epsilon')$ and subtracting (10) from (11), we obtain

$$\begin{aligned} &\nabla_{\theta} L_{con}^{MT} - \nabla_{\theta} L_{con}^{\Pi} \\ &= \nabla_{\theta} f_{\theta}(X+\epsilon')(\theta-\theta')^T \nabla_{\theta} f_{\theta}(X+\epsilon) \\ &\simeq \nabla_{\theta} f_{\theta}(X) \nabla_{\theta} f_{\theta}(X)^T (\theta-\theta'). \end{aligned} \tag{12}$$

In the last line of (12), we assumed gradients be constant in a small area around X. When the sample X is far away from the decision boundary, $\nabla_{\theta} f_{\theta}(X) \simeq 0$ and MT and Π model behave the same, but in the area near the decision boundary, it becomes $||\nabla_{\theta} f_{\theta}(X)|| \gg 0$, and in the gradient descent step, compared to the Π model, the negative gradient of MT ($\nabla_{\theta} \mathcal{J}$ in (3)) prohibits θ from being away from the target θ'. Because $f_{\theta}(X)$ is a continuous function, the distance between parameters is correlated with that between outputs. In the CISSL task, while Π model pushes the boundary towards the minor class, MT mitigates this by retaining the old target boundary like ensemble models.

In summary, the performance difference between the Π model and MT in CISSL is due to different targets of consistency regularization. The Π model uses the current model (θ) as a target. Therefore, the model smooths the decision boundary regardless of whether it passes the high-density area of the minor class. Because the target is the same as the parameter, smoothing causes model degradation as the parameter update is repeated. MT, on the other hand, targets a more conservative model (θ') than the current model. Note that since the target of MT is different from the current model, even if we reduce the learning rate of the Π model, it would work differently from MT. The conservative target has an ensemble effect with consistency regularization, so smoothing does not cause severe performance degradation.

Besides, we can explain the reason why MT performs better than the Π model in terms of batch sampling. In the mini-batch, minor class samples are sampled at a relatively low frequency. For this reason, the Π model frequently updates the model without a minor sample during the consistency regularization, which distorts the decision boundary. On the other hand, since the target of MT is calculated by EMA, even if there is no minor class sample in the mini-batch, it includes more information about the minor class samples. Thus, we can say that MT learns with a more stable target than the Π model.

C Experimental Details

C.1 Toy Examples Details

We generated *two moons* and *four spins* datasets. We split the train set into labeled data and unlabeled data with imbalance factor 5. The class distribution of unlabeled data follows that of labeled data. The size of the labeled data is 12 ($\{2, 10\}$ samples each) in two moons, 11 ($\{1,2,3,5\}$ samples each) in four spins. The size of the unlabeled data is 3000 in two moons, 2658 in four spins. Both datasets have 6,000 validation samples. We trained each algorithm by 5,000 iterations. The model is a 3-layer network; optimizer is SGD with momentum, the learning rate is 0.1 decaying at 4,000 iterations multiplied by 0.2, and momentum is 0.9.

Table 3: Number of unlabeled data in CIFAR10 and SVHN according to imbalance factor (imbalance factor = 100) and number of labeled data (CIFAR10 = 4k, SVHN = 1k).

(a) Different number of labeled data (Imbalance Factor = 100).

# of labeled data	250	500	1k	2k	4k
CIFAR10	-	-	10,160	9,160	7,160
SVHN	16,100	15,850	15,360	-	-

(b) Different imbalance factor (labeled data size: CIFAR10 = 4k and SVHN = 1k).

imbalance factor	10	20	50	100
CIFAR10	14,380	11,320	8,590	7,160
SVHN	25,940	21,440	17,450	15,360

Table 4: Hyperparameters for shared environment and each SSL algorithms and our method used in the experiments.

Shared	
Training iteration	500k
Consistency ramp-up iteration	200k
Initial learning rate	0.1
Cosine learning rate ramp-down iteration	600k
Weight decay	10^{-4}
Momentum	0.9
Π Model	
Max consistency coefficient	20
Mean Teacher	
Max consistency coefficient	8
Exponential Moving Average decay factor	0.95
VAT+em	
Max consistency coefficient	0.3
VAT ϵ (CIFAR10)	6.0
VAT ϵ (SVHN)	1.0
VAT ξ	10^{e-6}
VAT+EM+SNTG (as for VAT)	
Entropy penaly multiplier	0.06
Pseudo-Label	
Max consistency coefficient	1.0
Pseudo-label threshold	0.95
ICT	
Max consistency coefficient	100
Exponential Moving Average decay factor	0.999
ICT α	1.0
Suppressed Consitency Loss (Ours)	
Suppression Coefficient (β)	0.5

C.2 Datasets and Implementation Details

We conducted experiments using the CIFAR10 (Krizhevsky et al., 2009) and SVHN (Netzer et al., 2011) datasets in our proposed tasks and followed the common practice in SSL and CIL (Oliver et al., 2018; Johnson & Khoshgoftaar, 2019). For CIFAR10, there are 50,000 training images and 10,000 test images. We split the training set into a 45,000 train set and a 5,000 validation set for experiments. The validation set consists of the same size per class. We applied global contrast normalization and ZCA normalization. For data augmentation, we used random horizontal flipping, random cropping by padding 2 pixels each side of the image, and added Gaussian noise with standard deviation 0.15 to each pixel.

Table 5: Test error rates (%) and standard deviation from experiments with different re-weighting methods in CIFAR10 and SVHN. We compared inverse and normalization (IN), focal loss (FOCAL), and class-balanced loss (CB) to conventional cross-entropy loss (CE). (**Red**: best results for each row with same unlabeled data imbalance.)

Dataset		CIFAR 10				SVHN			
Re-weighting Method		CE	IN	Focal	CB	CE	IN	Focal	CB
Supervised	Uniform ($\rho_u = 1$)	36.71 ± 2.79	**35.73 ± 2.39**	36.80 ± 2.41	37.19 ± 2.88	35.89 ± 6.39	**34.60 ± 6.51**	35.45 ± 6.20	35.30 ± 7.08
Π-Model (Laine & Aila, 2016)		39.36 ± 4.47	**36.90 ± 3.56**	39.89 ± 4.23	39.20 ± 4.38	28.59 ± 7.90	**26.72 ± 6.12**	30.15 ± 8.13	27.99 ± 5.95
MT (Tarvainen & Valpola, 2017)		29.06 ± 5.13	**24.00 ± 3.17**	30.73 ± 6.20	29.50 ± 5.69	8.94 ± 2.12	**6.82 ± 0.34**	8.66 ± 2.11	7.86 ± 1.82
VAT + em (Miyato et al., 2018)		36.57 ± 7.20	**31.34 ± 5.01**	37.51 ± 7.56	36.78 ± 8.40	29.15 ± 4.80	**20.26 ± 7.97**	28.09 ± 5.46	29.37 ± 4.76
VAT + em + SNTG (Luo et al., 2018)		36.34 ± 6.54	**33.03 ± 4.78**	37.78 ± 6.94	36.26 ± 6.78	93.30 ± 0.00	93.30 ± 0.00	93.30 ± 0.00	93.30 ± 0.00
Pseudo-Label (Lee, 2013)		39.59 ± 5.70	**30.62 ± 3.62**	37.90 ± 6.87	39.38 ± 4.79	32.79 ± 7.62	**13.48 ± 2.45**	35.07 ± 10.85	34.38 ± 9.48
MT+SCL (ours)		22.62 ± 3.54	**21.59 ± 3.05**	23.44 ± 3.24	22.93 ± 3.53	8.56 ± 0.86	8.48 ± 1.47	**7.74 ± 0.58**	9.02 ± 1.34
Supervised	Half ($\rho_u = \rho/2$)	36.71 ± 2.79	**35.73 ± 2.39**	36.80 ± 2.41	37.19 ± 2.88	35.89 ± 6.39	34.60 ± 6.51	**35.45 ± 6.20**	35.30 ± 7.08
Π-Model (Laine & Aila, 2016)		38.84 ± 4.17	37.82 ± 1.55	38.28 ± 3.37	**37.51 ± 1.43**	33.73 ± 7.52	**29.60 ± 7.33**	31.67 ± 6.43	31.12 ± 7.72
MT (Tarvainen & Valpola, 2017)		35.37 ± 3.52	**33.08 ± 2.78**	34.45 ± 3.89	35.04 ± 3.42	17.23 ± 2.44	17.02 ± 3.93	**16.20 ± 3.70**	16.68 ± 2.24
VAT + em (Miyato et al., 2018)		36.77 ± 3.75	**36.20 ± 1.93**	37.62 ± 3.94	38.13 ± 4.63	30.44 ± 6.95	**27.44 ± 7.63**	28.62 ± 8.11	29.65 ± 6.90
VAT + em + SNTG (Luo et al., 2018)		38.48 ± 3.87	**35.90 ± 2.78**	38.01 ± 4.85	37.44 ± 4.00	93.30 ± 0.00	93.30 ± 0.00	93.30 ± 0.00	93.30 ± 0.00
Pseudo-Label (Lee, 2013)		39.72 ± 4.20	**37.36 ± 3.14**	38.77 ± 4.23	39.12 ± 3.19	33.70 ± 6.89	**31.83 ± 4.98**	32.79 ± 6.72	32.83 ± 8.27
MT+SCL (ours)		33.09 ± 3.63	**31.63 ± 2.31**	34.09 ± 3.22	33.14 ± 3.43	18.63 ± 3.97	18.59 ± 3.71	**16.34 ± 2.62**	16.44 ± 2.47
Supervised	Same ($\rho_u = \rho$)	36.71 ± 2.79	**35.73 ± 2.39**	36.80 ± 2.41	37.19 ± 2.88	35.89 ± 6.39	**34.60 ± 6.51**	35.45 ± 6.20	35.30 ± 7.08
Π-Model (Laine & Aila, 2016)		38.05 ± 3.19	**37.18 ± 2.12**	38.10 ± 3.37	37.34 ± 2.48	33.71 ± 8.17	**31.70 ± 6.94**	**31.70 ± 5.08**	33.17 ± 5.35
MT (Tarvainen & Valpola, 2017)		35.91 ± 3.70	**34.01 ± 2.85**	35.65 ± 2.64	35.17 ± 3.77	21.01 ± 4.14	20.80 ± 5.43	**20.01 ± 4.41**	21.77 ± 4.45
VAT + em (Miyato et al., 2018)		37.67 ± 2.20	**36.91 ± 2.33**	37.88 ± 3.66	37.64 ± 2.74	32.39 ± 8.25	**29.18 ± 7.45**	30.62 ± 8.23	30.93 ± 6.51
VAT + em + SNTG (Luo et al., 2018)		38.48 ± 2.96	**36.99 ± 2.77**	37.71 ± 4.32	37.53 ± 3.52	93.30 ± 0.00	93.30 ± 0.00	93.30 ± 0.00	93.30 ± 0.00
Pseudo-Label (Lee, 2013)		38.69 ± 4.28	**36.84 ± 3.31**	38.92 ± 3.31	38.52 ± 3.13	33.53 ± 8.08	**31.62 ± 6.06**	33.63 ± 6.24	34.55 ± 6.58
MT+SCL (ours)		34.22 ± 3.50	**32.09 ± 2.16**	33.93 ± 3.27	34.66 ± 4.37	**20.39 ± 4.10**	20.51 ± 5.43	20.95 ± 4.48	21.06 ± 4.78

For SVHN, there are 73,257 training images and 26,032 test images. We split the training set into a 65,931 train set and a 7,326 validation set for experiments. The validation set consists of the same size per class. We applied global contrast normalization and ZCA normalization. For data augmentation, we used random cropping by padding 2 pixels on each side of the image only.

In all experiments, we used the Wide-Resnet-28-2 model (Zagoruyko & Komodakis, 2016). It has enough capacity to show the performance improvement of SSL objectively (Oliver et al., 2018), and it is used in the new SSL methods (Berthelot et al., 2019b; Verma et al., 2019). Following the settings from (Verma et al., 2019), we set SGD with Nesterov momentum as our optimzer and adopted the cosine annealing technique (Loshchilov & Hutter, 2016). Detailed hyperparameters are set under a similar setting with (Oliver et al., 2018) and described in Table.4[2]. In our experiments, we used third-party implementation [3]. All the scores of test error rates are from five independent runs with different random seeds. Experiments with different random seeds shuffle the frequency ranking of each class when the imbalance factor is constant, and cover a variety of cases.

D Additional Experiments

D.1 Baselines to CISSL

We conducted experiments on how existing methods in the field of SSL and CIL perform in our defined CISSL task and used them as the baseline for our research. We experimented in the case of 4k and 1k labeled samples for CIFAR10 and SVHN each, both with imbalance factor 100.

D.1.1 Comparison of Semi-supervised Learning Methods

The column with imbalance factor 100 in the left part of Table.2 is the results of applying the SSL methods to the CISSL problem in CIFAR10. Except for MT, almost all SSL methods are inferior to supervised learning. Even if the unlabeled data imbalance is mitigated to *Uniform* case, there is no improvement in the performance of SSL methods except MT.

The column with imbalance factor 100 in the right part of Table.2 is the same experiment for SVHN. Most SSL methods perform better when the unlabeled data imbalance is lower, i.e. in *Uniform* case than in *Same* case. Notably, ICT showed a performance degradation of over $21\%p$ compared to the supervised learning, and SNTG even failed to train a model.

[2] https://github.com/brain-research/realistic-ssl-evaluation

[3] https://github.com/perrying/realistic-ssl-evaluation-pytorch

Table 6: Test error rates (%) and standard deviation from experiments with imbalance factor 100 and the number of labeled data {1k, 2k, 4k} in CIFAR10, and the number of labeled data {250, 500, 1k} in SVHN under 3 different unlabeled imbalance types. (**Bold/Red/Blue**: **supervised**, **best** and **second best** results for each column.)

Dataset		CIFAR 10			SVHN		
Number of labeled data		1000	2000	4000	250	500	1000
Supervised	Uniform ($\rho_u = 1$)	**54.24 ± 2.08**	**45.81 ± 3.00**	**36.71 ± 2.79**	**61.31 ± 8.05**	**47.98 ± 7.08**	**35.89 ± 6.39**
Π-Model (Laine & Aila, 2016)		56.82 ± 3.63	48.55 ± 4.26	39.36 ± 4.47	54.51 ± 8.43	39.49 ± 9.10	28.59 ± 7.90
MT (Tarvainen & Valpola, 2017)		**51.74 ± 5.33**	**38.94 ± 7.67**	**29.06 ± 5.13**	**38.32 ± 11.69**	**18.14 ± 10.47**	**8.94 ± 2.12**
VAT + em (Miyato et al., 2018)		53.68 ± 4.21	48.47 ± 3.66	36.57 ± 7.20	64.67 ± 6.41	44.04 ± 8.88	29.15 ± 4.80
VAT + em + SNTG (Luo et al., 2018)		54.53 ± 3.09	48.23 ± 3.50	36.34 ± 6.54	65.02 ± 5.23	93.30 ± 0.00	93.30 ± 0.00
Pseudo-Label (Lee, 2013)		58.19 ± 1.73	50.01 ± 2.78	39.59 ± 5.70	63.16 ± 6.60	49.78 ± 7.92	32.79 ± 7.62
ICT (Verma et al., 2019)		57.10 ± 4.56	48.25 ± 1.53	38.33 ± 4.72	86.54 ± 5.27	77.64 ± 1.94	67.02 ± 12.66
MT+SCL (ours)		**42.84 ± 2.88**	**28.69 ± 4.55**	**22.62 ± 3.54**	**26.25 ± 12.84**	**15.31 ± 6.81**	**8.56 ± 0.86**
Supervised	Half ($\rho_u = \rho_l/2$)	**54.24 ± 2.08**	**45.81 ± 3.00**	**36.71 ± 2.79**	**61.31 ± 8.05**	**47.98 ± 7.08**	**35.89 ± 6.39**
Π-Model (Laine & Aila, 2016)		55.99 ± 2.79	47.74 ± 3.82	38.84 ± 4.17	54.14 ± 9.11	42.20 ± 7.73	33.73 ± 7.52
MT (Tarvainen & Valpola, 2017)		**51.61 ± 4.58**	**42.47 ± 5.66**	**35.37 ± 3.52**	**41.72 ± 9.34**	**23.33 ± 10.78**	**17.23 ± 2.44**
VAT + em (Miyato et al., 2018)		53.60 ± 3.18	45.20 ± 4.84	36.77 ± 3.75	58.01 ± 10.44	41.15 ± 10.23	30.44 ± 6.95
VAT + em + SNTG (Luo et al., 2018)		55.59 ± 3.54	45.37 ± 3.25	38.48 ± 3.87	57.94 ± 8.87	93.30 ± 0.00	93.30 ± 0.00
Pseudo-Label (Lee, 2013)		57.05 ± 2.86	49.42 ± 2.42	39.72 ± 4.20	54.79 ± 10.42	44.32 ± 7.29	33.70 ± 6.89
ICT (Verma et al., 2019)		56.02 ± 3.37	47.60 ± 2.39	37.36 ± 2.02	84.22 ± 7.46	72.21 ± 9.43	58.99 ± 7.35
MT+SCL (ours)		**45.72 ± 2.62**	**39.97 ± 2.58**	**33.09 ± 3.63**	**33.44 ± 10.81**	**22.26 ± 6.22**	**18.63 ± 3.97**
Supervised	Same ($\rho_u = \rho_l$)	**54.24 ± 2.08**	**45.81 ± 3.00**	**36.71 ± 2.79**	**61.31 ± 8.05**	**47.98 ± 7.08**	**35.89 ± 6.39**
Π-Model (Laine & Aila, 2016)		55.42 ± 1.47	46.83 ± 3.29	38.05 ± 3.19	54.10 ± 10.07	43.89 ± 9.68	33.71 ± 8.17
MT (Tarvainen & Valpola, 2017)		**52.58 ± 3.23**	**44.11 ± 4.16**	**35.91 ± 3.70**	**42.42 ± 9.74**	**28.86 ± 10.57**	**21.01 ± 4.14**
VAT + em (Miyato et al., 2018)		53.62 ± 3.14	44.77 ± 2.82	37.67 ± 2.20	55.03 ± 8.85	42.44 ± 8.04	32.39 ± 8.25
VAT + em + SNTG (Luo et al., 2018)		55.55 ± 2.47	45.99 ± 4.29	38.48 ± 2.96	54.19 ± 9.43	93.30 ± 0.00	93.30 ± 0.00
Pseudo-Label (Lee, 2013)		56.68 ± 3.00	48.45 ± 3.00	38.69 ± 4.28	56.83 ± 8.81	43.71 ± 5.76	33.53 ± 8.08
ICT (Verma et al., 2019)		55.10 ± 2.68	47.19 ± 1.57	36.85 ± 3.44	85.15 ± 5.89	71.19 ± 7.68	56.97 ± 10.28
MT+SCL (ours)		**48.00 ± 3.41**	**40.69 ± 3.41**	**34.22 ± 3.50**	**35.32 ± 10.59**	**27.13 ± 10.58**	**20.39 ± 4.10**

From this experimental results and the analysis in Section 2, we used MT as our baseline, which performed best in all experiments.

D.1.2 Comparison of Class Imbalanced Learning Methods

We carried out the ablation experiments to cross-entropy loss (CE) as three types of CIL: Inverse and Normalization (IN), Focal loss, and Class-Balanced (CB) loss. We applied these CIL methods only to the supervised loss, L_{sup} in (3), and did not apply them to unlabeled data because we do not know the class label of the unlabeled data. In this experiment, we ignored ICT because CIL methods cannot be applied to ICT which uses mixup supervised loss.

The left part of Table.5 is the result of CIFAR10 experimented with imbalance factor 100, 4k labeled dataset. First of all, it seems that not all CIL methods always improve performance over CE. As unlabeled data imbalance and SSL methods change, their relative performance with CE differs. In this table, IN shows the best performance in all cases except the *Half* case of the Π model.

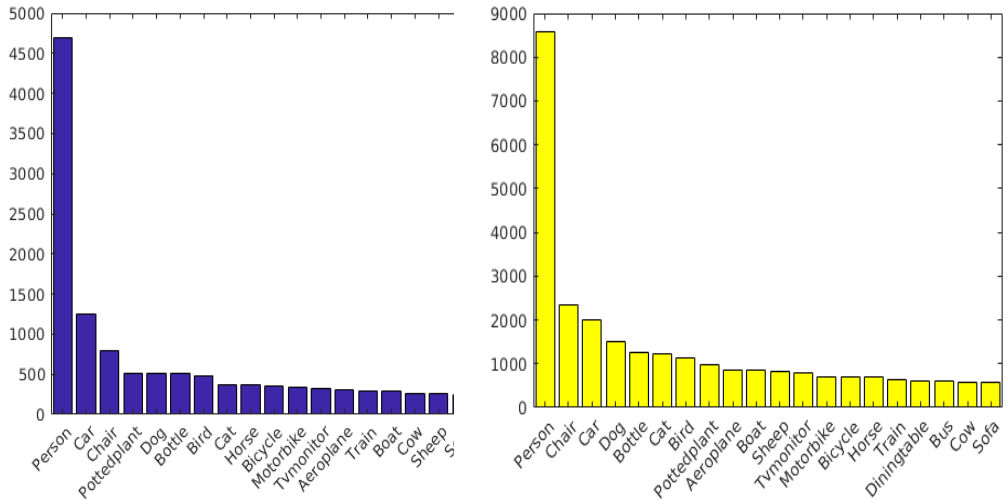
The right part of Table.5 is the result of SVHN experiments with imbalance factor 100, 1k labeled dataset. Unlike the previous CIFAR10 results, IN does not always dominate. The best algorithm differs according to the unlabeled data imbalance type in MT and our method. Since we do not know the unlabeled data imbalance beforehand, choosing a specific CIL algorithm does not guarantee a performance boost. So we used the most common cross-entropy as our baseline. In addition, SNTG failed to learn, as in the right part of Table.2.

D.2 Comparison of The Number of Labeled Samples

We experimented with keeping the imbalance factor while changing the number of labeled samples. We set the number of labeled data to {1k, 2k, 4k} in CIFAR10, and {250, 500, 1k} in SVHN. The results of CIFAR10 and SVHN are shown in Table 6, respectively.

In CIFAR-10, the left part of Table.6, the smaller the size of the labeled set, the lower the performance. In particular, when the size of the labeled data is 1k, most of the algorithms are weaker than supervised learning , while our method improves performance. This result indicates that consistency regularization is not valid when the baseline classifier is not performing well.

SVHN also shows similar tendency between the size of labeled data and performance in the right part of Table.6 . For SNTG and ICT, same as Section **??**, they have lower performance than supervised learning, either.



(a) Labeled dataset (VOC2007) (b) Unlabeled dataset (VOC2012)

Figure 3: Distributions for the labeled dataset (VOC2007) and the unlabeled dataset (VOC2012).

Table 7: Detection results for PASCAL VOC2007 testset. cls and loc are the consistency loss for classification and localization, respectively. We trained SSD300 on VOC07(L)+VOC12(U). Our result is from three independent trials.

Algorithm	Supervised	CSD (Jeong et al., 2019)		CSD + SCL(Ours)	
cls		o	o	o	o
loc			o		o
mAP (%)	70.2	71.7	72.3	72.07 ± 0.15	$\mathbf{72.60 \pm 0.10}$

D.3 Object Detection

We experimented with the object detection task to show the generality of our method, not limited to classification problems. We followed the CSD (Jeong et al., 2019) experiment settings and used the SSD300 model (Liu et al., 2016). We used PASCAL VOC2007 trainval dataset as the labeled data and PASCAL VOC2012 trainval dataset as the unlabeled data. We evaluated with PASCAL VOC2007 test dataset. Fig. 3 shows the distributions of PASCAL VOC data. The imbalance factor of labeled data is 22, and the imbalance factor of unlabeled data is 15. The order of the number of classes is also different. It means that the object detection task is more difficult and real settings. We applied our algorithm only to the classification consistency loss of CSD. In Table 7, supervised learning using VOC2007 shows 70.2 mAP. CSD with only classification consistency loss is 1.5%p higher than the supervised and CSD shows 2.1%p of enhancement. When SCL is applied to the CSD, our method shows additional improvement.