Towards Realistic Semi-Supervised Learning
Supplementary Materials

Mamshad Nayeem Rizve, Navid Kardan, and Mubarak Shah

Center for Research in Computer Vision, UCF, USA
{nayeemrizve, kardan}@knights.ucf.edu, shah@crcv.ucf.edu

1 Outline

This supplementary material includes the following sections. First, our training algorithm is introduced in section 2. Then, we provide implementation details in section 3. The details of the datasets that we use in our experiments are available in section 4. Next, in sections 5 and 6, we provide additional results with varying the amount of labeled data. After that, we provide additional details about estimating the number of novel classes in section 7. We discuss the effect of varying the number of novel classes for the novel class discovery (NCD) task in section 8. Next, we analyse the effect of changing temperature in section 9. Finally, in section 10, we demonstrate that our proposed method is able to recognise novel classes without confusing them with seen classes.

2 Training Algorithm

We provide our training algorithm in Alg. 1. For training, we require access to a labeled dataset, $\mathbb{D}_L$, and a set of unlabeled data, $\mathbb{D}_U$. We also require the prior class distribution, $\rho = \langle N_U^{C_1}/N_U, ..., N_U^{C_{|\mathbb{C}_L|+|\mathbb{C}_N|}}/N_U \rangle$, iterations per epoch, E, the maximum iterations, K, and the temperature, T. First, we initialize the neural network, f_w, and assign T to the sample uncertainties, $\mathbb{U}_L$ and $\mathbb{U}_U$. Next, we sample a batch of labeled and unlabeled data, as well as their corresponding sample uncertainties. After that, we obtain the class-distribution-aware pseudo-labels using Sinkhorn-Knopp algorithm[15] while minimizing the optimization problem in Eq. 1.

$$\min_{\mathbf{A} \in \mathcal{A}_\rho} -Tr((\mathbf{A}\mathbf{P}_\pi)^T \log(\hat{\mathbf{Y}}_U/N_U)), \tag{1}$$

where $\mathcal{A}_\rho$ is the transportation polytope defined in Eq. 3 of the main text which satisfies the prior class distribution ρ, P_π is the permutation matrix that reorders the columns of the assignment matrix, $\mathbf{A}$, according to the order of marginals of the output probabilities, $\hat{\mathbf{Y}}$.

We also perform cross pseudo-labeling to encourage perturbation invariant feature learning. Next, we generate the hard pseudo-labels for confident novel

classes from the generated pseudo-labels. As a result we will have a mixture of soft and hard pseudo-labels. Then, we concatenate the input images from the labeled and unlabeled batch. We also concatenate the ground-truth labels and generated coherent pseudo-labels, and the sample uncertainty scores. In the next step, we perform Mixup augmentation[18] on the concatenated inputs, labels, and uncertainty scores. Once we obtain the mixed inputs, labels, and uncertainty values, we update the network parameters using cross entropy loss. We update the uncertainty values for the unlabeled samples at the end of each epoch (E iterations). Finally, the algorithm ends after completing K iterations and returns the trained neural network, f_w.

Algorithm 1 Training algorithm

Input: Labeled data, $\mathbb{D}_L$, a set of unlabeled data, prior class distribution ρ, $\mathbb{D}_U$, iterations per epoch E, maximum iterations K, temperature T.
Output: Trained neural network, f_w.

Initialize neural network, f_w.
$\mathbb{U}_L \leftarrow T$, $\mathbb{U}_U \leftarrow T$ ▷ $\mathbb{U}_L$, and $\mathbb{U}_U$ are sample uncertainties.
for $k = 1...K$ **do**
  $(\mathbf{X}_l, \mathbf{Y}_l, \mathbf{u}_l) \leftarrow \text{SampleBatch}(\mathbb{D}_L, \mathbb{U}_L)$
  $(\mathbf{X}_u, \mathbf{X'}_u, \mathbf{u}_u) \leftarrow \text{SampleBatch}(\mathbb{D}_U, \mathbb{U}_U)$
  $\tilde{\mathbf{Y}}_u, \tilde{\mathbf{Y}}'_u \leftarrow \text{Sinkhorn}(f_w(\mathbf{X'}_u), \rho), \text{Sinkhorn}(f_w(\mathbf{X}_u), \rho)$ ▷ Eq. 1 and cross PL.
  $\bar{\mathbf{Y}}_u, \bar{\mathbf{Y}}'_u \leftarrow \text{MixedPL}(\tilde{\mathbf{Y}}_u), \text{MixedPL}(\tilde{\mathbf{Y}}'_u)$
  $\mathbf{X}, \mathbf{Y}, \mathbf{u} \leftarrow \text{Concat}(\mathbf{X}_l, \mathbf{X}_u, \mathbf{X'}_u), \text{Concat}(\mathbf{Y}_l, \bar{\mathbf{Y}}_u, \bar{\mathbf{Y}}'_u), \text{Concat}(\mathbf{u}_l, \mathbf{u}_u, \mathbf{u}_u)$
  $\mathbf{X}_m, \mathbf{Y}_m, \mathbf{u}_m \leftarrow \text{Mixup}(\mathbf{X}, \mathbf{Y}, \mathbf{u})$
  $w^{(k+1)} \leftarrow w^{(k)} - \alpha \nabla_w \mathcal{L}_{ce}(w^{(k)}, \mathbf{X}_m, \mathbf{Y}_m, \mathbf{u}_m)$
  if $k\%E = 0$ **then**
    $\mathbb{U}_U \leftarrow \text{Uncertainty}(\mathbb{D}_U)$ ▷ Eq. 6
return f_w

3 Implementation Details

To effectively process the lower resolution images from CIFAR-10 and CIFAR-100 datasets, similar to previous works [1,5,3], we modify the first convolutional layer and set the kernel size to 3×3 and apply a stride of 1. In addition, we remove the first max-pooling layer. We make a similar change for experiments in Tiny ImageNet dataset. However, we do not remove the first max-pooling layer since the images are of higher resolution. For ImageNet-100 and the fine-grained dataset experiments we do not make any changes to the network. Besides, for comparison, we use the same network for all the methods.

For data augmentation, we primarily use SimCLR[2] augmentations, which include: random resized crop, horizontal flip, color jittering, random grayscale, and Gaussian blur. For CIFAR-10 and CIFAR-100 we use solarize and equalize transformations instead of random grayscale and Gaussian blur. We also use Mixup augmentation in our training. For Mixup [18] augmentation, γ is set to

0.75. For all of our experiments, we use a threshold of 0.5 for generating hard pseudo-labels for the confident novel class samples.

For all of our experiments, except CIFAR-10 and CIFAR-100 experiments with 50% labeled data (Sec. 5), we apply a temperature value of 0.1. In our experiments, we observe that higher number of labeled examples per class (CIFAR-10 and CIFAR-100) creates a relatively stronger bias towards known classes when the temperature value is low. To address this issue, we use a temperature of 0.2 for the 50% labeled data experiments on CIFAR-10 and CIFAR-100 datasts. For performing uncertainty-guided temperature scaling we normalize the uncertainty values of the entire dataset to make the maximum uncertainty value 1. Finally, we clip the uncertainty values between 0.1 and 1.0 so that very low uncertainty values do not lead to overconfident predictions.

Table 1: Details of the datasets used in our experiments.

Dataset	**No Class**	**Train Samples**	**Test Samples**
CIFAR-10 [8]	10	50,000	10,000
CIFAR-100 [9]	100	50,000	10,000
ImageNet-100 [14]	100	128,545	5,000
Tiny ImageNet [11]	200	100,000	10,000
Oxford-IIIT Pet [13]	37	3,680	3,669
FGVC-Aircraft [12]	100	6,667	3,333
Stanford-Cars [7]	196	8,144	8,041

4 Datasets

We provide the details of the datasets used in our experiments in Tab. 1, which shows the number of classes in each dataset alongside the number of train and test samples. For FGVC-Aricraft[12] dataset, we train our model on the joint set of training and validation samples. Besides, since Oxford-IIIT Pet dataset contains odd number of classes, in 50% novel class experiment, we treat the first 19 classes of this dataset as seen and the remaining 18 classes as novel.

The input resolution of CIFAR-10 and CIFAR-100 images is 32×32; Tiny ImageNet images are slightly larger, i.e., 64×64. For the fine-grained datasets the images vary in size and aspect ratio. Therefore, for computational efficiency, we pre-process the images for fine-grained datasets and resize them to 256×256 resolution; this pre-processing operation is performed for both train and test images in all of our experiments.

5 Experiments with More Labeled Data

In this work we propose a solution for the realistic open-world SSL problem. Therefore, in the main text, we include experiments with only limited number

Table 2: Accuracy on **CIFAR-10** and **CIFAR-100** datasets with 50% classes as seen and 50% classes as novel.

Method	CIFAR-10			CIFAR-100		
	Seen	Novel	All	Seen	Novel	All
FixMatch[16]	71.5	50.4	49.5	39.6	23.5	20.3
DS^{3}L[4]	77.6	45.3	40.2	55.1	23.7	24.0
CGDL[17]	72.3	44.6	39.7	49.3	22.5	23.5
DTC [6]	53.9	39.5	38.3	31.3	22.9	18.3
RankStats[5]	86.6	81.0	82.9	36.4	28.4	23.1
SimCLR[2]	58.3	63.4	51.7	28.6	21.1	22.3
UNO[3]	91.6	69.3	80.5	68.3	36.5	51.5
ORCA[1]	88.2	90.4	89.7	66.9	43.0	48.1
Ours	96.8	92.8	94.8	80.2	49.3	64.7

Table 3: Accuracy on **Tiny ImageNet** dataset with 50% labeled data. We consider 50% classes as seen and 50% classes as novel.

Method	Seen	Novel	All
DTC [6]	28.8	16.3	19.9
RankStats [5]	5.7	5.4	3.4
UNO [3]	46.5	15.7	30.3
Ours	59.1	24.2	41.7

of labeled examples (10%). In this section, we provide additional results with more labeled data to provide additional comparison with other methods. To this end, we conduct experiments with 50% labeled data on CIFAR-10, CIFAR-100, and Tiny ImageNet. Tab. 2 reports the results on CIFAR-10 and CIFAR-100 datasets. We observe that similar to experiments with 10% labeled data, our proposed method outperforms all the other techniques. To be specific, in parallel to outperforming ORCA[1], our proposed algorithm also outperforms popular self-supervised learning method, SimCLR[2], and a recently proposed open-set recognition method, CGDL[17]. On CIFAR-10 dataset our porposed method outperforms ORCA[1] by 5.1% and on CIFAR-100 dataset it outperforms the second best method UNO[3] by 13.2%.

We conduct similar experiments on Tiny ImageNet dataset. Similar to our experiments in the main text, we compare our performance with DTC[6], RankStats[5], and UNO[3]. We observe that our proposed method outperforms the second best method UNO by 11.4%. The experiments on these three datasets demonstrate that our proposed method can perform reasonably well even with more labeled data.

Table 4: Accuracy on **CIFAR-10** dataset with 50% classes as seen and 50% classes as novel.

Method	1% Data			5% Data		
	Seen	**Novel**	**All**	**Seen**	**Novel**	**All**
UNO[3]	48.4	67.1	52.6	74.6	68.4	71.8
Ours	92.0	91.1	91.6	90.9	91.4	91.2

6 Reducing the Number of Labeled Data

In this section, we discuss additional experiments with lower number of labeled data. We report results on CIFAR-10 dataset with only 1% and 5% labeled data in Tab. 4. Since the source code for ORCA[1] is not publicly available, we restrict our comparison to UNO[3], which is the previous best method on this dataset. We observe that our proposed method significantly outperforms UNO in both of these experimental setups. We also notice that the performance of UNO on seen classes significantly degrades when only 1% labeled data is available, which is not the case for our porposed algorithm. Furthermore, even though we do not directly compare our results with ORCA and other baseline methods for these challenging experiments, *we notice that our proposed method with 1% and 5% labeled data, is able to outperform ORCA with higher number of labeled data, both 10% and 50% labeled data, on seen/novel/all class performances.*

Table 5: Accuracy on **CIFAR-100** dataset with 5% labeled data. We consider 50% classes as seen and 50% classes as novel.

Method	**Seen**	**Novel**	**All**
UNO[3]	44.0	31.7	36.5
Ours	60.1	47.4	54.4

Next, we conduct experiments on CIFAR-100 dataset with only 5% labeled data. The results are depicted in Tab. 5. For this experiment we also compare our results with UNO[3], which is the previous best method on this dataset. We notice that similar to the results on CIFAR-10, our proposed method outperforms UNO by a large margin and achieves 15.7% improvement over UNO on novel classes. Besides, the performance on novel classes and all classes is better than ORCA even when ORCA uses 50% of labeled data. The results on these two datasets demonstrate that our proposed algorithm is much more label efficient than ORCA, and can achieve strong performance even when only a handful of labeled examples (250 labeled examples in CIFAR-10 1% labeled data experiment) are available.

Finally, we conduct experiments on the fine-grained datasets with only 25% labeled data. We choose 25% labeled data for these experiments since even with

Table 6: Accuracy on **Oxford-IIIT Pet**, **FGVC-Aircraft**, and **Stanford-Cars** datasets with 25% labeled data. We consider 50% classes as seen and 50% classes as novel.

Method	Oxford-IIIT Pet			FGVC-Aircraft			Stanford-Cars		
	Seen	Novel	All	Seen	Novel	All	Seen	Novel	All
UNO[3]	35.6	19.1	25.8	28.2	20.7	21.9	26.5	10.3	17.2
Ours	59.1	31.4	45.6	52.4	36.5	45.3	67.4	32.5	50.0

such a large portion of labeled data, the number of labeled samples is not greater than ∼1000 samples. This experimental setup is challenging, since handling such large resolution images with a large number of classes is always difficult for neural networks. We present the results in Tab. 6. We compare our results with UNO[3] since the source code for ORCA is not publicly available. The results in Tab. 6 demonstrates that similar to 50% labeled data experiment, our proposed method outperforms UNO significantly on all of these datasets. These results further validate the effectiveness of our method by demonstrating that it can work on challenging fine-grained classification tasks with a large number of classes while using only a handful of labeled examples.

Table 7: Estimation of the number of novel classes. The table shows the estimated number of classes on different datasets with and without sample reassignment technique.

Dataset	GT	Estimated	
		w/o reassignment	w reassignment
CIFAR-10	10	10	10
CIFAR-100	100	87	117
ImageNet-100	100	84	139
Tiny ImageNet	200	132	192

7 Estimating Number of Novel Classes

For estimating the number of novel classes we vary the value of k from the number of labeled classes to 400. We can potentially use a higher upper limit but our experiment demonstrate that using a higher number does not change the number of estimated classes. For each value of k, we average the results from 3 independent runs of the k-means clustering algorithm to obtain more stable performance. For sample reassignment, first we perform Hungarian matching [10] for the labeled samples. Such matching provides us the dominant clusters for the labeled samples. After that, we select the misclassified labeled examples

and reassign them to these dominant labeled clusters using their nearest neighbor cluster. To obtain the final estimate of the number of classes we average the top 10 values and use that as our estimate to make the prediction more stable.

Table 7 reports the performance of our number of class estimation procedure with and without misclassified labeled example reassignment. Overall, we are able to estimate the number of classes reasonably well on the four common benchmark datasets that we investigated in our study. We observe that generally without the reassignment step, the estimated number of classes tends to be lower than the actual number of classes. On the more challenging Tiny ImageNet dataset, the reassignment step seems crucial for obtaining more accurate estimates.

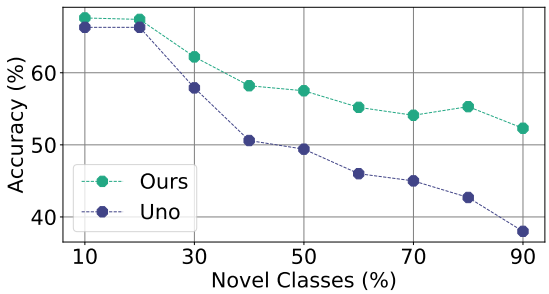

Fig. 1: Performance on **CIFAR-100** dataset for novel class discovery (NCD) task with different numbers of novel classes.

8 Varying the Number of Novel Classes for NCD Task

In the main text, we provide results on novel class discovery (NCD) task on CIFAR-100 dataset. In this section, we provide additional results by varying the percentage of novel classes. The results are provided in Fig. 1. We compare our method only with UNO since it outperforms the previous works with a significant margin. For this comparison we use the official code available for UNO. In this comparison, we report 'task-agnostic' accuracy, which is a more realistic evaluation. In 'task-agnostic' setting, we assume that we do not have any knowledge about the sample belonging to seen classes or novel classes. We observe that UNO achieves similar performance to our method when the number of novel classes is lower. However, as the percentage of novel classes increase, performance of UNO falls behind significantly. Moreover, we observe a predictable drop in performance for both methods as we increase the percentage of novel classes. In summary, these results demonstrate that our method can work well in more challenging scenarios where the number of novel classes are significantly higher than the seen classes.

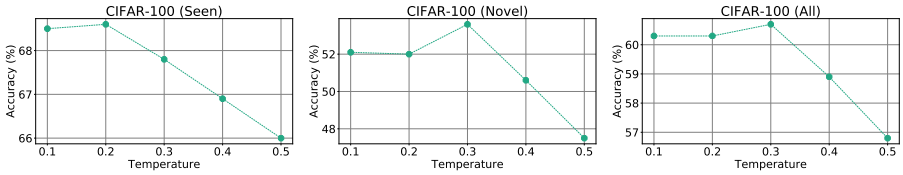

Fig. 2: Accuracy on **CIFAR-100** dataset with different temperature values. These graphs suggest that our proposed method is not sensitive to change of temperature parameter over a large interval of temperature values.

9 Analysis of Temperature

In this section, we discuss the effect of changing the temperature parameter of our proposed algorithm (Alg. 1). To this end, we conduct a series of experiments by varying the temperature parameter on CIFAR-100 dataset with 10% labeled data. The results are reported in Fig. 2. We notice that our proposed method is relatively stable over a large range of temperature values; even though we use a temperature of 0.1 in most of our experiments, results in Fig. 2 suggest that temperature values of 0.2 and 0.3 also yield similar performances. However, the performance on seen/novel/all classes deteriorates for temperature values greater than 0.3. Overall these results demonstrate that the performance of our method is relatively stable to the choice of temperature hyperparameter.

10 Confusing Novel Classes with Seen Classes

To perform an in-depth analysis of the performance of our proposed method on both seen and novel classes, we plot the confusion matrix for all the test samples of CIFAR-100 dataset (10% labeeld data). In this analysis, we use Hungarian algorithm[10] to match the predictions for all classes to the ground-truth labels. We report the results in Fig. 3. These results provide evidence that our proposed algorithm can successfully recognise novel classes without confusing them with seen classes. Moreover, interestingly, our proposed method performs well even for a difficult problem, such as CIFAR-100, where it encounters a high number of novel classes.

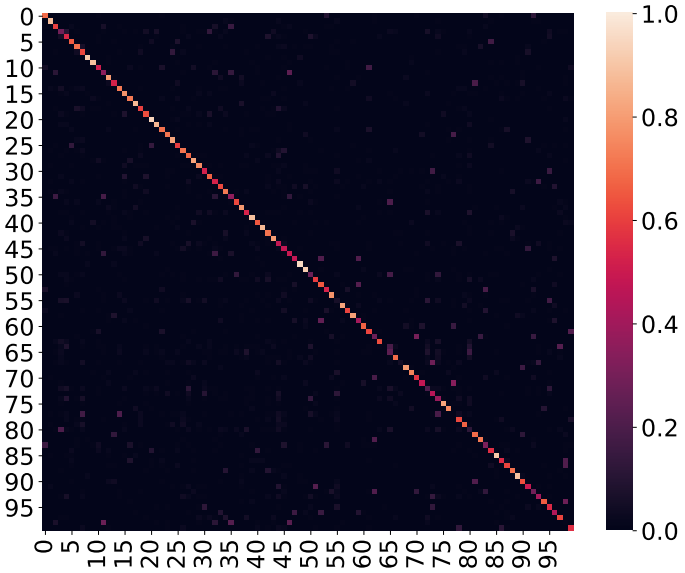

Fig. 3: Confusion matrix for test samples of **CIFAR-100** dataset. This matrix shows that our proposed method successfully recognises novel classes without confusing them with seen classes.

References

1. Cao, K., Brbic, M., Leskovec, J.: Open-world semi-supervised learning. In: International Conference on Learning Representations (2022), https://openreview.net/forum?id=O-r8LOR-CCA 2, 4, 5
2. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. arXiv preprint arXiv:2002.05709 (2020) 2, 4
3. Fini, E., Sangineto, E., Lathuilière, S., Zhong, Z., Nabi, M., Ricci, E.: A unified objective for novel class discovery. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9284–9292 (2021) 2, 4, 5, 6
4. Guo, L.Z., Zhang, Z.Y., Jiang, Y., Li, Y.F., Zhou, Z.H.: Safe deep semi-supervised learning for unseen-class unlabeled data. In: International Conference on Machine Learning. pp. 3897–3906. PMLR (2020) 4
5. Han, K., Rebuffi, S.A., Ehrhardt, S., Vedaldi, A., Zisserman, A.: Automatically discovering and learning new visual categories with ranking statistics. In: International Conference on Learning Representations (2020) 2, 4
6. Han, K., Vedaldi, A., Zisserman, A.: Learning to discover novel visual categories via deep transfer clustering. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8401–8409 (2019) 4
7. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: 4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13). Sydney, Australia (2013) 3
8. Krizhevsky, A., Nair, V., Hinton, G.: Cifar-10 (canadian institute for advanced research) 3
9. Krizhevsky, A., Nair, V., Hinton, G.: Cifar-100 (canadian institute for advanced research) 3
10. Kuhn, H.W.: The hungarian method for the assignment problem. Naval research logistics quarterly **2**(1-2), 83–97 (1955) 6, 8
11. Le, Y., Yang, X.: Tiny imagenet visual recognition challenge. CS 231N **7**(7), 3 (2015) 3
12. Maji, S., Kannala, J., Rahtu, E., Blaschko, M., Vedaldi, A.: Fine-grained visual classification of aircraft. Tech. rep. (2013) 3
13. Parkhi, O.M., Vedaldi, A., Zisserman, A., Jawahar, C.V.: Cats and dogs. In: IEEE Conference on Computer Vision and Pattern Recognition (2012) 3
14. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International journal of computer vision **115**(3), 211–252 (2015) 3
15. Sinkhorn, R., Knopp, P.: Concerning nonnegative matrices and doubly stochastic matrices. Pacific Journal of Mathematics **21**(2), 343–348 (1967) 1
16. Sohn, K., Berthelot, D., Li, C.L., Zhang, Z., Carlini, N., Cubuk, E.D., Kurakin, A., Zhang, H., Raffel, C.: Fixmatch: Simplifying semi-supervised learning with consistency and confidence. arXiv preprint arXiv:2001.07685 (2020) 4
17. Sun, X., Yang, Z., Zhang, C., Ling, K.V., Peng, G.: Conditional gaussian distribution learning for open set recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13480–13489 (2020) 4
18. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. In: International Conference on Learning Representations (2018) 2