Supplementary Material: Class-incremental Novel Class Discovery

This supplementary material is organised as follows: In Sec. A we report additional details about the experimental set-up. In Sec. B we dissect the evaluation protocols for existing incremental NCD method, and compare the performance of ResTune [4] with our proposed FRoST. Finally, in Sec. C we report additional experiments on CIFAR100 for the two-step class-iNCD setting.

A Experimental Set-up

Datasets. In the Tab. A we report the standard splits between the old and the new classes for the three benchmarks. Note that for the CIFAR100 and Tiny-ImageNet there is an imbalance between the old and the new classes.

Table A1. Dataset statistics for class-incremental novel class discovery.

Dataset	Labeled Set (Old classes)		Unlabeled Set (New classes)		Test Set	
	#image	#class	#image	#class	#image	#class
CIFAR-10	25K	5	25K	5	5K	10
CIFAR-100	40K	80	10K	20	5K	100
Tiny-ImageNet	90K	180	10K	20	10K	200

Implementation Details. We have trained our model with the SGD optimizer and the initial learning rate set to 0.1, which is then decayed by a factor of 10 after 170 epochs. The total training epoch is 200 and the batch size is set to 128. For the mean-squared error loss, following [2] we adopt the ramp-up function with weight $\gamma = \{5, 50, 50\}$ and ramp-up length $T = \{50, 150, 150\}$ for CIFAR-10, CIFAR-100 and Tiny-ImageNet, respectively. For the self-training loss, we use the same ramp-up length of corresponding data set, but use the weight $\gamma = 0.05$ for all data sets.

B Comparison with ResTune

B.1 ResTune in the Original Setting versus class-iNCD

In this work we argue that an incremental NCD algorithm should be evaluated in a task-agnostic fashion with a joint classifier (see Sec. 3 and Fig. 2 (b) in the main paper), such that, at any stage in the lifetime of the model, the predicted classes should fall in the corresponding bucket of class indices seen in a given training session. In other words, if the model sees samples from task $\mathcal{T}^{[\texttt{old}]}$ containing classes of indices 0-4 in a given stage of training, then at any future inference stage, say after having trained on $\mathcal{T}^{[\texttt{new}]}$, the model must assign test samples from $\mathcal{T}^{[\texttt{old}]}$ to the first five logits. This particular evaluation protocol has been introduced as the class-iNCD setting in our work. While ResTune (RT) [4]

Table A2. Comparison of the evaluation protocols alongside the classifier heads used in *Original*$_{\text{RT}}$ and our proposed class-iNCD. The metrics and the corresponding classifier heads used to obtain them are color coded

Metric	Classifier head	
	Original$_{\text{RT}}$ Protocol	class-iNCD Protocol
Old$_{\text{RT}}$/**Old**	old-head	joint-head
New$_{\text{RT}}$/**New**	new-head	joint-head
All$_{\text{RT}}$/**All**	concat-head	joint-head

has been proposed for the task of incremental NCD, its evaluation differs from our class-iNCD. Concretely, ResTune reports three evaluation accuracy in their work: *(i)* the task-aware accuracy on the old classes (abbreviated as **Old**$_{\text{RT}}$) that uses the "old" classifier head; *(ii)* the task-aware accuracy on the new classes (abbreviated as **New**$_{\text{RT}}$) that uses the "new" classifier head; and *(iii)* the task-agnostic clustering accuracy on all classes (abbreviated as **All**$_{\text{RT}}$) that employs the concatenated "old" and "new" classifier heads (or concat-head). As discussed in Sec. 3 of the main paper, the evaluation using the Hungarian Assignment on all the test data set to obtain **All**$_{\text{RT}}$ is unreasonable because some of the new classes may get assigned to old classes, making the inference inconsistent between tasks. We denote the evaluation of ResTune as *Original*$_{\text{RT}}$ protocol.

On the other hand, for the class-iNCD setting, we use a single classification head to report three accuracies: **Old**, **New** and **All** corresponding to the samples from the old, new and all the classes, as described in the Sec. 4.1 of the main paper. One crucial difference between the *Original*$_{\text{RT}}$ and class-iNCD is that in class-iNCD evaluation protocol we always use a joint classifier head (or the concat-head for ResTune) to evaluate all the metrics. Moreover, our class-iNCD also uses Hungarian Assignment for the **All** metric, but only for obtaining the re-assigned ground truth for the "new" classes (being unsupervised) and *not* on all the data set (see Fig. 2). This ensures that the samples from the old classes and the new classes are evaluated correctly and rightfully results in a drop in the performance if cross-task class assignment occurs (see Fig. 2 (b)), which should be the desired behaviour for any incremental predictor. We visually demonstrate the difference between the *Original*$_{\text{RT}}$ and class-iNCD protocols in Tab. A2. Note than when ResTune is evaluated in the class-iNCD setting, the concat-head is used because ResTune by construction has two separate heads.

Next we show that the *Original*$_{\text{RT}}$ protocol introduced in [3] is flawed and gives a false sense of improvement in performance over the erstwhile baselines, given the split chosen between labelled and unlabelled classes. To this end, we re-run ResTune, using the official code published by the authors[3] of [4], to obtain the performance in the *Original*$_{\text{RT}}$ setting. We report the numbers of ResTune in the left halves of the Tab. A3, A4 and A5 under the *Original*$_{\text{RT}}$ setting for CIFAR-10, CIFAR-100 and Tiny-ImageNet, respectively. Except for the CIFAR-10, the ResTune numbers are mostly reproducible for the metric **All**$_{\text{RT}}$, which

[3] https://github.com/liuyudut/ResTune

Table A3. The comparison of the ResTune with FRoST using the $\textit{Original}_{\text{RT}}$ and the class-iNCD evaluation protocols on the CIFAR-10 data set (5 old classes and 5 new classes). While the ResTune fairs well in the $\textit{Original}_{\text{RT}}$ setting, the new classes performance is dramatically low when tested under the class-iNCD. Contrarily, FRoST maintains a balanced performance over all the classes by consistently outperforming ResTune

Dataset	CIFAR-10 (#Old:5; #New:5)					
Protocol	$\textit{Original}_{\text{RT}}$			class-iNCD		
Method	**Old**$_{\text{RT}}$	**New**$_{\text{RT}}$	**All**$_{\text{RT}}$	**Old**	**New**	**All**
ResTune[4] (in paper)	85.5	89.0	52.1	-	-	-
ResTune[4] (reproduce)	91.7	76.7	46.9	**91.7**	0.0	45.9
FRoST (Ours)	-	-	-	77.5	**49.5**	**63.4**

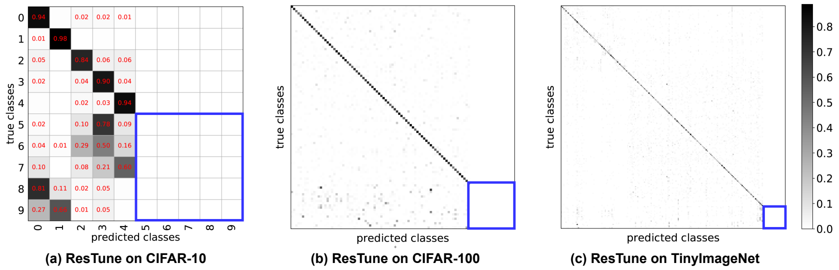
Table A4. The comparison of the ResTune with FRoST using the $\textit{Original}_{\text{RT}}$ and the class-iNCD evaluation protocols on the CIFAR-100 data set (80 old classes and 20 new classes). While the ResTune fairs well in the $\textit{Original}_{\text{RT}}$ setting, the new classes performance is dramatically low when tested under the class-iNCD. Contrarily, FRoST maintains a balanced performance over all the classes by consistently outperforming ResTune

Dataset	CIFAR-100 (#Old:80; #New:20)					
Protocol	$\textit{Original}_{\text{RT}}$			class-iNCD		
Method	**Old**$_{\text{RT}}$	**New**$_{\text{RT}}$	**All**$_{\text{RT}}$	**Old**	**New**	**All**
ResTune[4] (in paper)	73.8	63.7	59.1	-	-	-
ResTune[4] (reproduce)	73.8	56.0	59.0	**73.8**	0.0	59.0
FRoST (Ours)	-	-	-	64.6	**45.8**	**59.2**

Table A5. The comparison of the ResTune with FRoST using the $\textit{Original}_{\text{RT}}$ and the class-iNCD evaluation protocols on the Tiny-ImageNet data set (180 old classes and 20 new classes). While the ResTune fairs well in the $\textit{Original}_{\text{RT}}$ setting, the new classes performance is dramatically low when tested under the class-iNCD. Contrarily, FRoST maintains a balanced performance over all the classes by consistently outperforming ResTune

Dataset	TinyImageNet (#Old:180; #New:20)					
Protocol	$\textit{Original}_{\text{RT}}$			class-iNCD		
Method	**Old**$_{\text{RT}}$	**New**$_{\text{RT}}$	**All**$_{\text{RT}}$	**Old**	**New**	**All**
ResTune[4] (in paper)	58.0	46.3	41.2	-	-	-
ResTune[4] (reproduce)	44.3	27.3	40.4	44.3	0.0	39.9
FRoST (Ours)	-	-	-	**54.5**	**33.7**	**52.3**

is of interest to us since it is obtained with the concat-head without the need of task-id. To better understand the distribution of predictions for ResTune in the task-agnostic classification case we plot the confusion matrix (CM) for all the data sets in Fig. A1. We can immediately notice from the CMs that all the samples in the data set have been predicted as one of the old classes. This signifies that the ResTune simply can not predict any new classes correctly when evaluated in the task-agnostic setting (*i.e.*, **All**$_{\text{RT}}$), indicated by the *empty* blue box in Fig. A1 (a)-(c). The illusion of performance for the **All**$_{\text{RT}}$ comes from the old classes because cardinality of old classes dominate the new classes (*e.g.*, 80 old classes vs 20 new classes for CIFAR-100, etc). While the **New**$_{\text{RT}}$ is reasonably



(a) ResTune on CIFAR-10 (b) ResTune on CIFAR-100 (c) ResTune on TinyImageNet

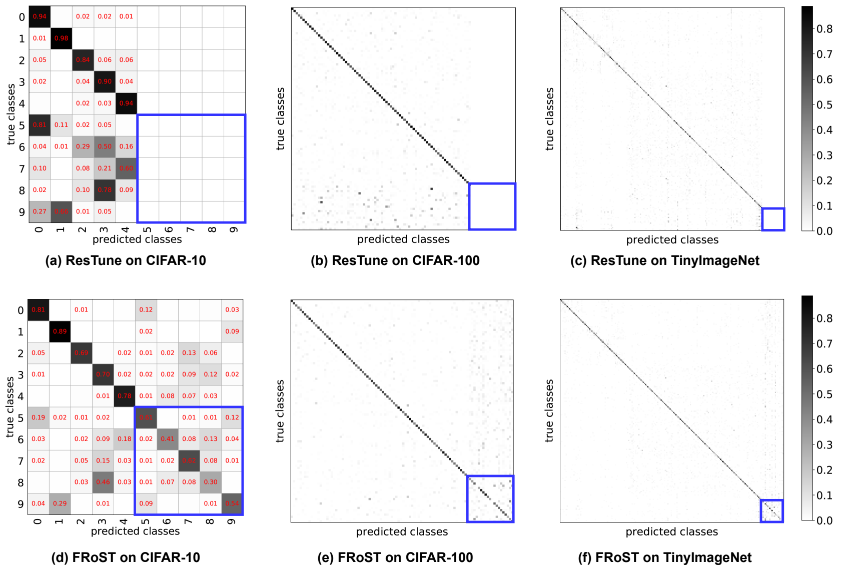
Fig. A1. The confusion matrices for ResTune when evaluated in the task-agnostic case (or $\textbf{All}_{\text{RT}}$) of the $\textit{Original}_{\text{RT}}$ setting. The □ denotes the part of the confusion matrix which corresponds to the new classes. The ResTune always predicts test samples from the new classes as belonging to the old classes. Digital zoom is recommended

good, it is task-id dependant and thus, meaningless in practical applications. Therefore, the $\textit{Original}_{\text{RT}}$ protocol introduced in [4] does not truly reflect the classification capability of a incremental NCD algorithm.

Given, the inherent flaws of the evaluation protocol of $\textit{Original}_{\text{RT}}$, in our work we propose the class-iNCD and then evaluate ResTune using this protocol. In the class-iNCD, the concat-head is always used for evaluating the ResTune, and the performance is reported in the right halves of the Tab. A3, A4, A5. We notice that the **All** metric for ResTune do not vary much from the $\textbf{All}_{\text{RT}}$. However, there is an acute drop in performance for the **New** metric in comparison to the $\textbf{New}_{\text{RT}}$, dropping all the way to 0% for all the data sets. As shown previously in Fig. A1, this is a consequence of the ResTune always getting activated in the old logits given any test sample from the new classes. This is expected because in class-iNCD, just like the **All** metric, the **New** metric is also task-agnostic. Thus, it can be concluded that the protocol of our class-iNCD can accurately evaluate if an incremental learner is well-behaved for both the new and old classes simultaneously.

B.2 Comparison of FRoST with ResTune

In this section we compare the ResTune with our FRoST under the class-iNCD evaluation protocol. The comparison on all the data sets have been reported in the right halves of the Tab. A3, A4, and A5. Overall, under the **All** metric our proposed FRoST consistently outperforms the ResTune. Given the **All** metric can be misleading, with the old classes dominating the performance over the new classes, we also compare the two methods using the **New** accuracy metric. As evident from the results of the **New** metric, we find that our joint classifier can satisfactorily classify the new classes in the task-agnostic evaluation setting, when compared to the ResTune. The ResTune results in a dismal performance of 0%, meaning that ResTune is not suitable for discovering new classes in the task-agnostic setting. This can indeed be verified by visual inspection of the CMs reported in Fig. A2. As can be observed, the CMs of FRoST is more diagonal



(a) ResTune on CIFAR-10 (b) ResTune on CIFAR-100 (c) ResTune on TinyImageNet

(d) FRoST on CIFAR-10 (e) FRoST on CIFAR-100 (f) FRoST on TinyImageNet

Fig. A2. Comparison of the confusion matrices (CMs) for the ResTune (a)-(c) with our FRoST (d)-(f) evaluated on all the data sets. The □ denotes the part of the confusion matrix which corresponds to the new classes. While, the ResTune always predicts test samples from the new classes as belonging to the old classes, the CMs of FRoST is mostly diagonal. This means that FRoST can satisfactorily classify the new samples into the corresponding new classes. Digital zoom is recommended

than that of ResTune, especially in the bottom part of CM which corresponds to the new classes. The well-behaved nature of FRoST comes at the price of reduced **Old** accuracy. However, this is acceptable because the goal of class-iNCD task is to simultaneously perform well in both old and new classes, unlike the highly skewed response in ResTune.

B.3 Discussion

We conjecture that the skewed predictions from the joint-head (or concat-head) of the ResTune is caused by the decoupled training of the separate classifier heads: old-head and new-head. Specifically, in ResTune, the old head is trained with the objective of LwF [3], whereas the new head is trained with a modified DTC [1] objective. Due to lack of synergy between the two heads that receive gradients of different magnitudes, causes the predictions to be skewed (or biased) towards one of them. Contrarily, in our FRoST the joint-head is always trained with cross-entropy (CE) loss. In more details, the CE loss is constructed from the feature-replay from the old class prototypes and the pseudo-labels from the novel-head corresponding to the new samples. Due to the usage of a homogeneous training objective for the joint classifier, the norm of the weights of the classifier

Table A6. The comparison of FRoST with the state-of-the-art methods in the two-step class-iNCD setting for the CIFAR-100 data set, where new classes arrive in two steps, instead of one. In the first step the metrics are: **New-1-J**: new classes performance with the joint-head (or concat-head); **New-1-N**: new classes performance from new-head. In the second step the metrics are: **New-1-J**: the performance of the previous 10 new classes; **New-2-J**: new classes performance

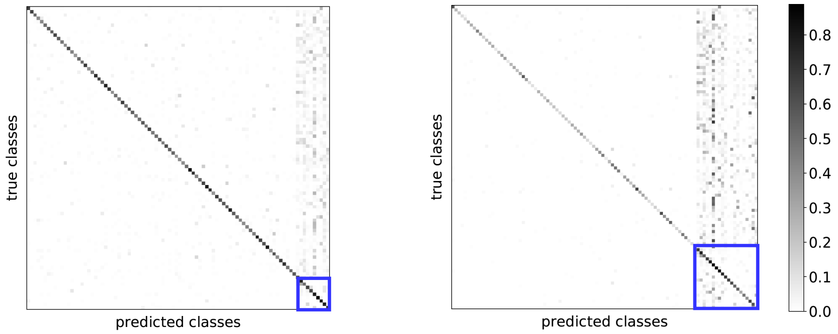
Methods	CIFAR-100									
	First Step (80-10)				Second Step (80-10-10)					
	Old	**New-1-J**	**All**	**New-1-N**	**Old**	**New-1-J**	**New-2-J**	**All**	**New-1-N**	**New-2-N**
DTC[1]	61.0	0.0	54.2	51.5	50.0	0.0	0.0	40.0	42.6	58.9
NCL[5]	**70.1**	8.0	**63.2**	55.3	**70.4**	0.0	7.1	**57.0**	28.7	67.6
ResTune[4]	61.8	0.0	54.9	**79.6**	59.1	0.0	0.0	47.3	50.5	78.7
FRoST	56.4	**72.8**	58.2	77.5	25.8	**75.0**	**48.4**	33.0	**77.3**	**79.6**

is much balanced (see Fig. 4 in the main). Thus, our FRoST can well predict both the old and the new classes, leading to a balanced performance and further justifying the validity of the proposed components. We believe that this insight is quite important and can be exploited in the future works.

C Additional Experiments for Two-Step class-iNCD

In this section we present a detailed analysis of the two-step class-iNCD setting, in addition to the experiments reported in the Sec. 4.3 of the main paper. We run a two-step class-iNCD on CIFAR-100 (with 80 base classes) where in the first step we have 10 new classes. Subsequently, in the second class-iNCD step we have another 10 new classes, resulting in a total of 90 old classes and 10 new classes. In the *First Step* of Tab. A6 we report the following metrics: *(i)* **Old** is the performance on the first 80 old classes; *(ii)* **New-1-J** is the performance on the 10 new classes seen during step 1; and *(iii)* **All** is the combined performance on all the classes seen until the end of the first step. All these metrics are computed with the joint-head for our FRoST or using the concat-head for the baselines which do not support a joint-head. Note that we additionally report **New-1-N** that describes the performance of the new classes obtained using the new-head at the first step. Similarly, in the *Second Step* when another 10 new classes are added we further report the **New-2-J** and **New-2-N** that deals with the performance of the newly added 10 classes from the joint-head and new-head, respectively.

As can be seen from the Tab. A6 our FRoST achieves a well-balanced performance for both the old and the new classes in both the steps, in contrast to the ResTune, which fails to detect the new classes when evaluated with the concat-head. To prove that ResTune can discover the new classes when evaluated in a task-aware protocol, we report the performance of the new-head through the metrics **New-1-N** and **New-2-J**. Indeed, the task-aware new classes performance is at par with FRoST in the first step, but experiences a drop in the second step. Thus FRoST suffers from less forgetting as far as the first 10 new classes are



(a) First step: 80-10

(b) Second step: 80-10-10

Fig. A3. Confusion matrices (CMs) of the FRoST in the sequential two-step class-iNCD setting on CIFAR-100. (a) First step denotes the stage when we have 80 old classes and 10 new classes; and (b) Second step denotes the stage when we have an additional 10 new classes. The □ denotes the part of the CMs which corresponds to the new classes seen after the supervised training on the old classes. The CMs are quasi-diagonal even after two steps and well-balanced over the old and the new classes

concerned. We visually inspect the distribution in predictions through the CM of FRoST in the Fig. A3 and observe that the CM is still quasi-diagonal, with some tendency to predict more the new classes in the second-step. Nevertheless, we improve over the baselines by a large margin when the overall performance is concerned.

References

1. Han, K., Vedaldi, A., Zisserman, A.: Learning to discover novel visual categories via deep transfer clustering. In: ICCV (2019) 5, 6
2. Han, K., Rebuffi, S.A., Ehrhardt, S., Vedaldi, A., Zisserman, A.: Automatically discovering and learning new visual categories with ranking statistics. In: ICLR (2020) 1
3. Li, Z., Hoiem, D.: Learning without forgetting. TPAMI (2017) 2, 5
4. Liu, Y., Tuytelaars, T.: Residual tuning: Toward novel category discovery without labels. TNNLS (2022) 1, 2, 3, 4, 6
5. Zhong, Z., Fini, E., Roy, S., Luo, Z., Ricci, E., Sebe, N.: Neighborhood contrastive learning for novel class discovery. In: CVPR (2021) 6