Appendix

A. Constrained K-means Additional Details

Qian et al. [32] proposed the online mini-batch solver for the constrained K-means objective (Eqn. 3) proposed by [6], and used it for unsupervised representation learning. In our method, we adopted the same solver but for a different purpose; we use online clustering as an alternative to offline nearest neighbour search to identify neighbourhood of images and leverage such information to perform our label refinement procedure. To that end, due to the empirical observation that the maximal value of dual variables is well bounded, our Eqn. 6 is an approximation of the original dual variables update proposed by Qian et al. after each mini-batch:

$$\rho_k^t = \Pi_{\Delta_\delta}(\rho_k^{t-1} - \eta\frac{1}{B}\sum_{i=1}^{B}(\mu_{i,k}^t - \frac{\gamma}{N})), \tag{1}$$

where Π_{Δ_δ} projects the dual variables to the domain $\Delta_\delta = \{\rho|\forall k, \rho_k \geq 0, \|\rho\|_1 \leq \delta\}$.

We refer the readers to the original paper for guarantees of performance complete proofs.

Constrained vs unconstrained clustering. Our purpose in PROTOCON is to use K-means as an alternative for offline nearest-neighbours retrieval, which automatically mandates that we use equi-partition clustering by constraining minimum cluster size γ to be the number of nearest neighbours n. However, we relax this constraint to $\gamma = 0.9n$ to allow cluster sizes to slightly vary to capture the inherent imbalance in salient properties of different classes. Empirically, we found this to work well across the datasets we used. We also tested the setting with $\gamma = 0$ which translates to unconstrained clustering. This setting was unstable and did not lead to performance gains; where we found that clustering collapses to only a few clusters. For example in CIFAR-10 (40 labels) setting, K-means converged to only 20 clusters. The consequence is that we have only 20 cluster pseudo-labels to use for refining all the unlabeled samples in subsequent epochs which is a very general summary of neighbourhoods and hence it hurts the performance rather than help it. Please refer to Tab. 4 for further ablations on the value of n.

Mini-batch updates vs Epoch updates Another decision choice is the frequency of cluster centroids updates (Eqn. 7). Since PROTOCON does not memorise image representations, centroids can be updated either every mini-batch, or by accumulating representations of images based on their cluster assignment throughout an epoch and then performing the update once at the end of the epoch. The former solution is useful in helping K-means convergence which requires multiple assignment-update iterations, however it leads to higher variance due to the stochastic nature of mini-batches. On the other hand, the latter solution is also sub-optimal as it requires long time for clusters to converge. Accordingly, we adopted a warmup period during which we use mini-batch updates to speed up convergence, henceforward, we switch to epoch updates to stabilise the centroids and exhibit less variance. We found that for smaller datasets, 20 epochs of warmup are sufficient, while for the larger datasets with more classes, we increase the warmup period to 70 epochs.

B. Additional Training Dynamics Analysis

Here, to further understand PROTOCON, we examine more of its training dynamics.

Clustering purity vs pseudo-label accuracy. First, we investigate the properties of the clusters as training proceeds. We follow a similar setup like that used to obtain Fig. 4, but this time, we use the captured statistics to calculate cluster purity for each class. Specifically, by the end of each epoch, we count the members of each cluster (*e.g.* for CIFAR-10, we use $K = 250$, so we count the number of images assigned by K-means to each of the 250 clusters), then for each cluster, we check the most dominant class among its members based on their ground truth labels. Subsequently, we calculate the purity of each cluster as the ratio between the number of images belonging to the dominant class to the total number of cluster members. Finally, to calculate purity for a given class, we average the described ratio across all clusters for which that class is the dominant one. In Fig. 5, we display cluster purity per class of CIFAR-10 during the first 130 epochs of training side-by-side to the pseudo-label accuracy for each class. This is to allow us to investigate the clustering effectivness in the critical initial phase of training and how it affects the obtained pseudo-labels quality. We see that for the more distinguishable classes (*e.g.* truck or ship), clustering purity increases significantly faster than others matching with a corresponding increase in pseudo-label accuracy. Whereas for more confusing classes (*e.g.* horse and deer), the cluster purity suffers a slow increase accompanied with what seem to be high disagreement between cluster and classifier pseudo-labels leading to an overall slow increase of pseudo-label accuracy (note that we display the refined pseudo-label accuracy in the figure). Finally, the most confusing classes (*e.g.* cat and dog) have the lowest cluster purity leading to a low pseudo-label accuracy at first, but we notice that once the majority of other classes are learnt (*i.e.* have higher accuracy, the more confusing classes start to catch up (notice the cat and dog curves towards the end of Fig. 5-b). This is in line with our expectation that easy classes are first learnt by the network, then it moves on to discriminate the less obvious ones.

Pseudo-label Retention Ratio. Like the state-of-the-

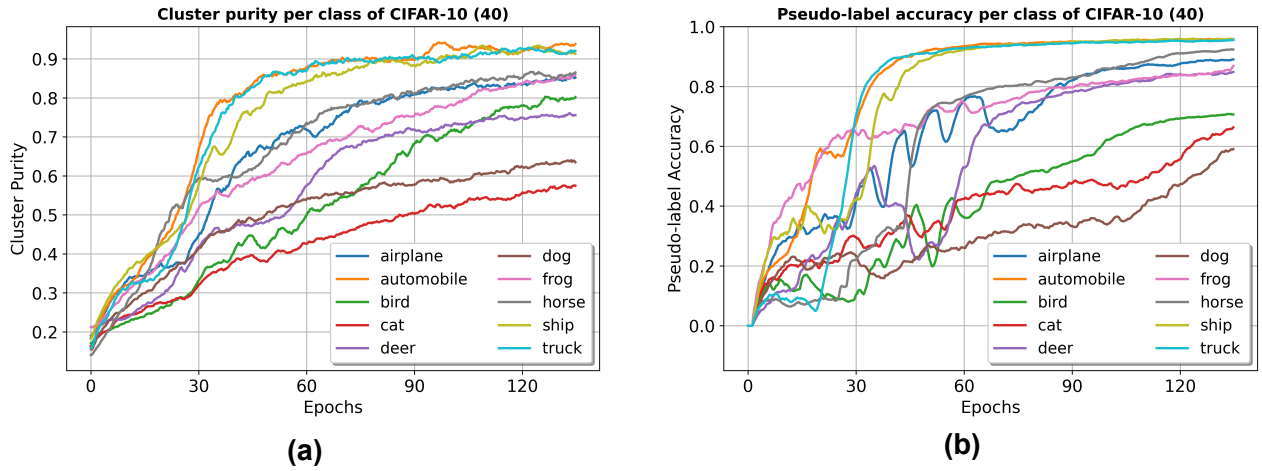

Figure 5. **Analysis Plots. (a)**: Cluster Purity per class of CIFAR-10 vs training epochs, when trained using PROTOCON with 4 images per class. **(b):** Pseudo-label accuracy per class vs training epochs. Best viewed in color.

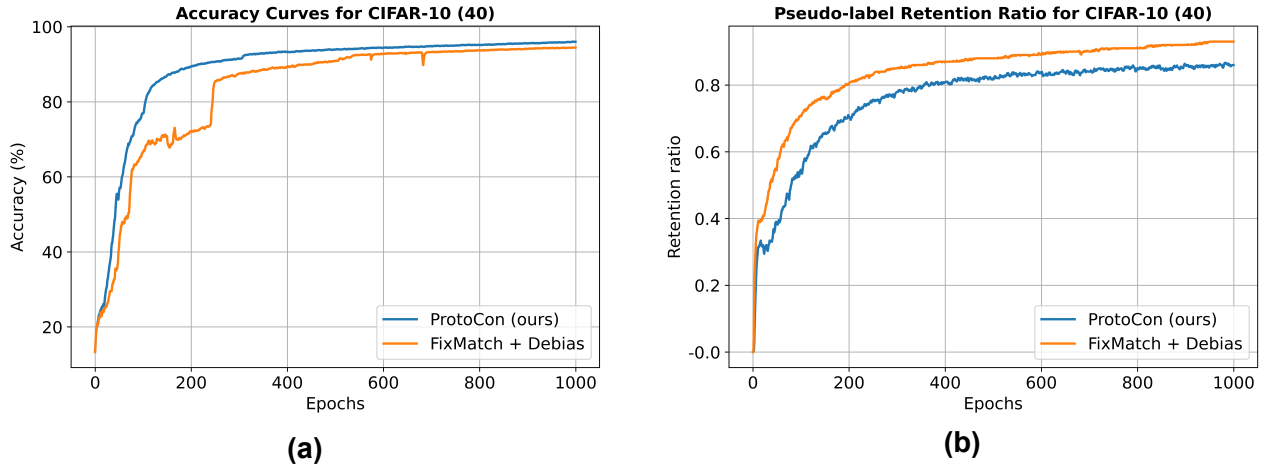

Figure 6. **Analysis Plots. (a)**: Pseudo-label accuracy vs epochs. **(b):** Retention rate vs epochs which denotes the ratio of unlabeled samples retained by each method for pseudo-labeling (*i.e.* with maximum confidence score higher than the threshold τ.)

art SSL method (DebiasPL [42]), PROTOCON is also a confidence-based pseudo-labeling method albeit with additional ingredients. Hence, both methods only retain high-confidence unlabeled samples for pseudo-labeling. In Fig. 6, we examine the retention rate (*i.e.* ratio of samples with maximum confidence exceeding the threshold τ) for both methods as training proceeds (b), and compare it with the pseudo-labeling accuracy exhibited by each (a). We observe that even though our method outperforms DebiasPL, in terms of accuracy, throughout the training, it consistently retains almost 10% less samples for pseudo-labeling. This finding speaks to our original motivation (see Sec. 1) with regards to the over-confidence problem underpinning the lower performance of SOTA methods in label-scarce regime. Compared to its counterparts, PROTOCON is more conservative when it comes to admitting a sample as "reliable" for pseudo-labeling; primarily because the refined pseudo-labels we employ is a combination of the original classifier pseudo-label and the neighbourhood pseudo-label. As we show in Fig. 3-a, the disagreement between the two results in a lower overall confidence in predictions. Such conservative nature of PROTOCON is key to avoiding confirmation bias even when there is only a few labeled samples available.

C. PROTOCON in Moderate-label Regime

In this section, we examine our method performance when more than 10 images per class are available (which we call moderate-label regime). To recap, our method primarily aims to address confirmation bias in label-scarce settings. Yet, intuitively, the refinement strategy might also

Table 5. CIFAR and Mini-ImageNet accuracy in moderate-label regime for different amounts of labeled samples averaged over 3 different splits. All results are produced using the same codebase and same splits.

	CIFAR-100		Mini-ImageNet	
Total labeled samples	2500	4000	2500	4000
FixMatch [38]	71.71±0.35	74.08±0.13	44.53±0.44	50.21±0.09
FixMatch + DB [42]	72.44±0.15	74.43±0.06	46.18±0.23	52.00±0.04
PROTOCON	**73.31±0.43**	**75.18±0.02**	**48.61±0.34**	**53.67±0.06**
delta against best baseline	+0.87	+0.75	+2.43	+1.67

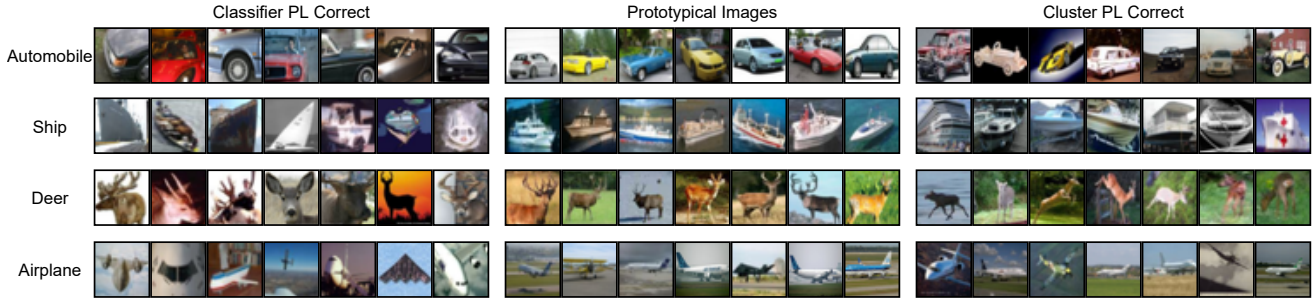

Figure 7. Additional examples to complement Fig. 4.

help moderate-label regimes. As such, we investigate this hypothesis by running additional experiments on CIFARs and Mini-ImageNet with 25, and 40 images per class. We find that for CIFAR-10, performance already saturates after 10 images per class and most of the compared methods perform similarly. As for the other two datasets with 100 classes each, we find PROTOCON to still provide performance gains. However, with more labels available, we find that using less neighbouring samples to perform the refinement (*i.e.* less n) works better. Specifically, we reduce n by a factor of 10 (*i.e.* $n = 25$ instead of $n = 250$). Additionally, since with more labels, all the compared methods exhibit significantly less variance, we report results only based on 3 runs instead of 5. Please refer to the results in Tab. 5.

D. Additional Quantitative Examples

Here, we detail our experimental setup for obtaining Fig. 4 and we provide additional examples in Fig. 7.

Experimental Setup. As training proceeds, for each epoch, we capture per-image statistics such as: the classifier pseudo-label and its max score (*i.e.* $\arg\max \boldsymbol{p}_w$ and $\max \boldsymbol{p}_w$ respectively); cluster pseudo-label and its max score (*i.e.* $\arg\max \boldsymbol{z}^a$ and $\max \boldsymbol{z}^a$ respectively), sample prototypical score (*i.e.* $\boldsymbol{q}^w \cdot \mathcal{P}_{\hat{y}}$) denoting how close a sample is to its class prototype. Subsequently, to obtain the prototypical images (in middle panel of Fig. 4 and 7), we rank images of each class based on their prototypical score averaged over the first 500 epochs of training. Additionally, we identify images for which the cluster pseudo-labels are, on average, more accurate than that of the classifier (and the other way around) by comparing the respective pseudo-labels with the ground truth label of each image. Thus, we display on the left panel images for which the classifier pseudo-label is, on average, more accurate than the cluster pseudo-label, and the opposite on the right panel.

Additional Examples. In Fig. 7, we provide more examples to complement those in Fig. 4. To reiterate, we see that the cluster pseudo-labels which capture the samples' neighbourhood in the prototypical space (trained via our prototypical loss) are usually more accurate if images are more prototypical even if they are lacking discriminative features (*e.g.* blurry images or zoomed out images). In contrast, the pseudo-labels in the class probability space (trained via one-hot cross entropy) are usually more accurate for images with discriminative features (*e.g.* car bumpers or deer horns) even if they lack prototypicality. The diversity of views captured via the different labels is key to PROTOCON's effectiveness as it helps the classifier learns via the disagreement between the two views through the refined label.