A. Appendix

A.1. Semi-supervised k-means (ssKM)

Parameters initialization for ssKM. We denote the centroid parameters of ssKM as $\boldsymbol{W} = (\boldsymbol{W}^{old}, \boldsymbol{W}^{new})$, where

$$\begin{cases} \boldsymbol{W}^{old} = (\boldsymbol{w}_k^{old})_{1 \leq k \leq K^{old}} \\ \boldsymbol{W}^{new} = (\boldsymbol{w}_k^{new})_{K^{old}+1 \leq k \leq K^{old}+K^{new}}. \end{cases}$$

Here, $\boldsymbol{W}^{old}$ and $\boldsymbol{W}^{new}$ are the centroids for the known and novel classes, respectively. We initialize $\boldsymbol{W}$ by using both the labeled set and the unlabeled set jointly. As in [42], we first produce the centroids for the known classes using the labeled set, such that: $\boldsymbol{w}_k^{old} = (\sum_{\boldsymbol{z}_i \in \mathcal{Z}_L} y_{i,k} \boldsymbol{z}_i) / \sum_{\boldsymbol{z}_i \in \mathcal{Z}_L} y_{i,k}$. Then, we initialize the centroids corresponding to the novel classes with kmeans++ initialization.

ssKM objective and clustering process. As in [42], we update the cluster centroids of ssKM algorithm, with the following objective:

$$\begin{aligned} L_{\text{ssKM}}(\mathbf{Y}; \mathbf{U}; \boldsymbol{W}) = & \left(\sum_{k=1}^{K} \sum_{\boldsymbol{z}_i \in \mathcal{Z}_L} y_{i,k} \|\boldsymbol{z}_i - \boldsymbol{w}_k\|_2 \right) \\ & + \left(\sum_{k=1}^{K} \sum_{\boldsymbol{z}_i \in \mathcal{Z}_U} u_{i,k} \|\boldsymbol{z}_i - \boldsymbol{w}_k\|_2 \right), \end{aligned} \tag{11}$$

where $\|.\|_2$ denotes the Euclidean distance, and $\boldsymbol{u}_i = (u_{i,k})_{1 \leq k \leq K}$ denotes the latent binary vector assigning point $\boldsymbol{z}_i$ to cluster k. $\mathbf{U} \in \{0,1\}^{NK}$ denotes the latent assignment matrix composed of the latent binary vectors. ssKM algorithm proceeds with the same block-coordinate descent approach as the standard unsupervised k-means [33]. Thus, in order to minimize L_{ssKM}, it proceeds with the following cluster assignment and centroid updates cycle:

- **U**-update: Do the label assignment of the unlabeled points such that,

$$u_{i,k} = \begin{cases} 1 & \text{if } \arg\min_k \|\boldsymbol{z}_i - \boldsymbol{w}_k\|_2 = k \\ 0 & \text{otherwise.} \end{cases}$$

- $\boldsymbol{W}$-update: Find $\arg\min_{\boldsymbol{W}} L_{\text{ssKM}}(\mathbf{Y}; \mathbf{U}; \boldsymbol{W})$

Note that the latent label assignment update step is not applied on the labeled points, for which we simply keep using the available ground-truth labels in the first term in (11), all along the clustering process.

A.2. Supplementary results

Effect of prototypes initialization. All along our article experiments, we found empirically consistent to use ssKM (detailed in Sec. A.1) centroids to initialize the prototypes of the proposed partitioning model PIM. In this section, in order to endorse this choice, we empirically emphasize across Tab. 6 the effect of prototypes initialization on PIM prediction performances. We thus compare the following three different possible initializations (INIT): -SSRDM INIT consists of using labeled points for prototypes of known classes, and random points for prototypes of novel classes; -SSKM++ INIT consists of using labeled points for prototypes of known classes, and kmeans++ points for prototypes of novel classes (w.r.t. known class prototypes); -SSKM INIT consists of using ssKM centroids as INIT prototypes. The results observed on Tab. 6 show that the proposed approach is overall almost insensitive to prototypes initialization. One can also note that PIM performances are overall slightly improved when using directly SSKM++ INIT.

	CUB			STANFORD CARS			HERBARIUM19		
INIT	ALL	OLD	NEW	ALL	OLD	NEW	ALL	OLD	NEW
SSRDM	62.1	76.2	55.1	43.0	64.3	32.7	42.7	55.3	36.0
SSKM++	64.3	76.3	58.4	43.5	65.6	32.8	42.3	55.3	35.4
SSKM	62.7	75.7	56.2	43.1	66.9	31.6	42.3	56.1	34.8
	CIFAR10			CIFAR100			IMAGENET-100		
INIT	ALL	OLD	NEW	ALL	OLD	NEW	ALL	OLD	NEW
SSRDM	94.6	97.4	93.2	78.2	85.4	64.0	82.0	95.4	75.2
SSKM++	94.7	97.4	93.3	78.5	84.3	66.9	83.6	95.4	77.8
SSKM	94.7	97.4	93.3	78.3	84.2	66.5	83.1	95.3	77.0

Table 6. **Prototypes initialization effect** on PIM ACC performances on fine-grained and generic datasets.

Effect of PIM objective components on generic datasets. Tab. 7 shows the effect of PIM objective components on the generic datasets, in particular on CIFAR100 and IMAGENET-100.

			CIFAR10			CIFAR100			IMAGENET-100		
LOSS TERMS USED	$\mathcal{H}(Y)$ ON	S.T. $\underbrace{\boldsymbol{y}_i = \boldsymbol{p}_i}_{\forall \boldsymbol{z}_i \in \mathcal{Z}_L}$	ALL	OLD	NEW	ALL	OLD	NEW	ALL	OLD	NEW
$-\mathcal{H}(Y\|Z)$	-	$\times$	94.5	97.5	93.1	2.3	1.0	5.0	39.1	82.3	17.3
$-\mathcal{H}(Y\|Z)$	-	✓	94.5	97.5	93.0	66.3	73.1	52.8	54.7	85.6	39.1
$\mathcal{H}(Y) - \mathcal{H}(Y\|Z)$	$\mathcal{Z}_U$	$\times$	92.3	97.9	89.5	72.4	82.5	52.3	79.2	87.6	75.0
$\mathcal{H}(Y) - \mathcal{H}(Y\|Z)$	$\mathcal{Z}$	$\times$	94.8	97.2	93.6	78.8	82.4	71.5	83.1	93.9	77.6
$\mathcal{H}(Y) - \mathcal{H}(Y\|Z)$	$\mathcal{Z}_U$	✓	92.4	98.1	89.5	73.9	84.7	52.4	79.3	94.6	71.7
$\mathcal{H}(Y) - \mathcal{H}(Y\|Z)$	$\mathcal{Z}$	✓	94.7	97.4	93.3	78.3	84.2	66.5	83.1	95.3	77.0

Table 7. **PIM objective components effects** on generic datasets in terms of ACC scores.

Standard deviation ($\pm$) of PIM for the generalized category discovery partitioning is shown on Tab.8.

	CUB			STANFORD CARS			HERBARIUM19		
APPROACH	ALL	OLD	NEW	ALL	OLD	NEW	ALL	OLD	NEW
PIM STD.-DEV. ($\pm$)	$\pm$0.5	$\pm$0.8	$\pm$0.7	$\pm$0.3	$\pm$0.9	$\pm$0.3	$\pm$0.3	$\pm$0.8	$\pm$0.2
	CIFAR10			CIFAR100			IMAGENET-100		
APPROACH	ALL	OLD	NEW	ALL	OLD	NEW	ALL	OLD	NEW
PIM STD.-DEV. ($\pm$)	$\pm$0.0	$\pm$0.0	$\pm$0.0	$\pm$1.0	$\pm$0.5	$\pm$2.8	$\pm$0.3	$\pm$0.0	$\pm$0.5

Table 8. **Standard deviation** ($\pm$) for ACC scores of PIM (Tab. 1) over 5 trainings, across fine-grained and generic datasets.