A. Details of the framework for splitting and merging clusters

In this section, we provide details for the Metropolis-Hastings framework. In the Gaussian Mixture Model, we have three sets of parameters, (π_i, μ_i, Σ_i) where π_i is the mixture weight and μ_i, Σ_i are the mean and covariance matrix. These parameters are assumed to be sampled from a prior distribution. When μ_i and Σ_i are unknown for the multivariate Gaussian distribution, we adopt the Normal Inverse Wishart (NIW) distribution as the prior to sample them for algebraic convenience, because NIW distribution is a conjugate prior and the conjugacy property can lead to a closed-form expression of the posterior.

The Inverse Wishart (IW) distribution is defined as follows:

$$p(\Sigma_i) \sim \mathcal{W}^{-1}(\nu, \Psi) = \frac{|\nu\Psi|^{\frac{\nu}{2}}}{2^{\frac{\nu d}{2}}\Gamma_d(\frac{\nu}{2})}|\Sigma_i|^{-\frac{\nu+d+1}{2}}\exp(-\frac{1}{2}tr(\nu\Psi\Sigma_i^{-1})), \tag{7}$$

where Σ_i is a $d \times d$ Symmetric and Positive Definite (SPD) matrix, $\nu > d-1$, $\Psi \in \mathbb{R}^{d\times d}$ is SPD, and Γ_d is a d-dimensional multivariate factorial function. The positive real number ν and the SPD matrix Ψ are the parameters of the IW distribution. The data distribution determined by μ_i and Σ_i follows NIW distribution, if the joint probability density function is defined by

$$p(\mu_i, \Sigma_i) \sim \text{NIW}(\kappa, \mathbf{m}, \nu, \Psi) \triangleq \mathcal{N}(\mu_i; \mathbf{m}, \frac{1}{\kappa}\Sigma_i)\mathcal{W}^{-1}(\Sigma_i; \nu, \Psi), \tag{8}$$

where $\mathbf{m} \in \mathbb{R}^d$, $\kappa > 0$, and $\mathcal{N}(\mu_i; \mathbf{m}, \frac{1}{\kappa}\Sigma_i)$ is a d-dimensional Gaussian with mean $\mathbf{m}$ and covariance $\frac{1}{\kappa}\Sigma_i$ evaluated at μ_i.

Given a set of features $\mathcal{Z}_i$ (with $N_i = |\mathcal{Z}_i|$) assigned to the Gaussian component μ_i, Σ_i, we can have a posterior distribution of μ_i, Σ_i in a closed-form thanks to the conjugacy:

$$p(\mu_i, \Sigma_i|\mathcal{Z}_i) = \text{NIW}(\mu_i, \Sigma_i; \kappa^*, \mathbf{m}_i^*, \nu^*, \Psi_i^*), \tag{9}$$

where the posterior parameters are obtained by:

$$\kappa_i^* = \kappa + N_i \tag{10}$$

$$\mathbf{m}_i^* = \frac{1}{\kappa_i^*}[\kappa\mathbf{m} + \sum_{z_k\in\mathcal{Z}} z_k] \tag{11}$$

$$\nu_i^* = \nu + N_i \tag{12}$$

$$\Psi_i^* = \frac{1}{\nu^*}[\nu\Psi + \kappa\mathbf{m}\mathbf{m}^\top + (\sum_{z_k\in\mathcal{Z}_i} z_k z_k^\top) - \kappa_i^*\mathbf{m}_i^*{\mathbf{m}_i^*}^\top] \tag{13}$$

In Eq.5 and 6 of the main paper, we need to calculate the marginal likelihood function of the observed data $\mathcal{Z}_i$ by integrating out the μ_i and Σ_i parameters in the Gaussian. Let $\theta = (\mathbf{m}, \kappa, \Psi, \nu)$ be the parameters of the NIW distribution. The marginal likelihood can be defined as follows:

$$h(\mathcal{Z}_i; \theta) = \int p(\mathcal{Z}_i|\mu_i, \Sigma_i)p(\mu_i, \Sigma_i; \theta)d(\mu_i, \Sigma_i) \tag{14}$$

$$= \frac{1}{\pi^{Nd/2}}\frac{\Gamma_d(\nu^*/2)}{\Gamma_d(\nu/2)}\frac{|\nu\Psi|^{\nu/2}}{|\nu^*\Psi_i^*|^{\nu^*/2}}\frac{\kappa^{d/2}}{{\kappa^*}^{d/2}}, \tag{15}$$

with which we can compute the Eq. 5 and 6 in the main paper.

B. Estimating the number of clusters on validation set

In this section, we validate the choice of K^n_{init} using only the labelled data, to better reflect the real world use case. In particular, we further split the classes in the labelled data $\mathcal{D}^l$ into two parts, $\mathcal{D}^l_r$ and $\mathcal{D}^l_p$. We drop the labels in $\mathcal{D}^l_p$. We verify the effectiveness of different choice of K^n_{init} on $\mathcal{D}^l_p$ and report the results in Tab. 11. Interestingly, we observe that the initial guess of a number around $\frac{K^l}{2}$ often leads to a good estimate.

Table 11: **Results of varying the initial guessed K^n_{init}.** 'GT K^n' is the ground truth number of novel classes, split from the labelled set. K^n is the estimated number of novel classes.

Dataset	K^l	GT K^n	$K^n_{init} =$	1	3	5	10	20	25	50
CIFAR-10	3	2	$K^n =$	2	2	4	5	3	6	8
CIFAR-100	60	20	$K^n =$	15	18	18	20	21	22	29
ImageNet-100	25	25	$K^n =$	18	19	22	21	23	27	29
CUB	50	50	$K^n =$	38	37	41	46	49	52	50
SCars	49	49	$K^n =$	39	38	40	42	43	48	51

C. Convergence of the estimated class number in longer training

In Tab. 12, we show results by training the model for 800 epochs on CUB, demonstrating the convergence after longer training.

Table 12: **Results of estimated class number with longer training.**

Epoch	200	400	600	700	720	740	760	780	800
GT $K^n = 100$	112	122	114	108	109	107	108	106	107

D. Error bars for generalized category discovery performance

We repeatedly run our method and the previous state-of-the-art three times with different random seeds to show the mean and standard deviation values in Tab. 13 and Tab. 14, for both known and unknown class number cases. We can see that the variation is relatively small for all methods, and our method consistently outperforms the previous state-of-the-art across the board for both known and unknown class number cases.

Table 13: **Results on generic image classification datasets.**

				CIFAR10			CIFAR100			ImageNet-100		
No.	Methods	Known K	PCA	All	Old	New	All	Old	New	All	Old	New
(1)	Vaze *et al.* [48]	✓	✗	91.5±0.4	97.9±0.2	88.2±0.6	76.9±0.3	84.6±0.3	61.5±0.2	75.0±0.3	92.1±0.2	66.6±0.4
(2)	Ours (GPC)	✓	✗	91.9±0.2	98.2±0.3	88.6±0.1	77.6±0.4	84.9±0.4	62.7±0.4	76.7±0.4	94.3±0.2	68.8±0.3
(3)	Ours (GPC)	✓	✓	**91.9±0.4**	**98.2±0.3**	**89.1±0.2**	**77.8±0.3**	**85.3±0.2**	**63.5±0.2**	**77.3±0.4**	**94.6±0.4**	**71.1±0.3**
(4)	Vaze *et al.* [48]	✗	✗	88.6±0.5	96.2±0.4	84.9±0.6	73.2±0.4	83.5±0.4	57.9±0.4	72.7±0.4	91.8±0.5	63.8±0.6
(5)	Vaze *et al.* [48]	✗	✓	89.7±0.4	97.3±0.5	86.3±0.4	74.8±0.5	83.8±0.4	58.7±0.6	73.8±0.4	92.1±0.5	64.6±0.6
(6)	Ours (GPC)	✗	✗	88.2±0.4	97.0±0.5	85.9±0.3	75.1±0.5	84.4±0.4	59.9±0.6	74.9±0.5	93.2±0.4	65.5±0.3
(7)	Ours (GPC)	✗	✓	**90.6±0.3**	**98.2±0.4**	**87.1±0.4**	**75.7±0.5**	**84.7±0.6**	**60.9±0.4**	**75.7±0.3**	**93.4±0.4**	**66.8±0.5**

Table 14: **Results on Semantic Shift Benchmark datasets.**

				CUB			Stanford Cars			FGVC-aircraft		
No.	Methods	Known K	PCA	All	Old	New	All	Old	New	All	Old	New
(1)	Vaze *et al.* [48]	✓	✗	51.1±0.2	56.4±0.1	48.4±0.3	39.1±0.3	57.6±0.4	29.9±0.3	45.1±0.2	41.2±0.3	46.8±0.2
(2)	Ours (GPC)	✓	✗	54.5±0.2	54.6±0.4	50.3±0.2	42.0±0.2	58.9±0.2	32.0±0.3	46.3±0.2	42.3±0.2	47.1±0.3
(3)	Ours (GPC)	✓	✓	**55.3±0.4**	**58.1±0.3**	**53.2±0.4**	**42.7±0.3**	**60.0±0.4**	**33.0±0.2**	**46.5±0.3**	**42.8±0.5**	**47.2±0.1**
(4)	Vaze *et al.* [48]	✗	✗	47.2±0.4	55.1±0.3	44.8±0.2	35.0±0.3	56.0±0.4	24.8±0.3	40.1±0.2	40.8±0.4	42.8±0.1
(5)	Vaze *et al.* [48]	✗	✓	49.2±0.3	**56.2±0.2**	**46.3±0.4**	36.3±0.3	56.6±0.4	25.9±0.5	41.2±0.3	**40.9±0.4**	44.6±0.2
(6)	Ours (GPC)	✗	✗	50.5±0.3	52.5±0.4	45.8±0.5	37.0±0.6	56.6±0.3	26.1±0.2	39.8±0.3	39.7±0.2	42.5±0.2
(7)	Ours (GPC)	✗	✓	**52.1±0.3**	55.4±0.2	45.7±0.3	**38.9±0.4**	**58.9±0.3**	**28.6±0.5**	**43.4±0.3**	40.8±0.4	**44.7±0.3**

E. Standard deviation of category number estimation

In this section, we show the standard deviations of estimated category numbers by repeatedly running our method with different random seeds. The results are shown in Tab. 15. We can see that our method can estimate a more accurate category number to Vaze *et al.* [48].

Table 15: **Estimated category numbers.**

Estimated K^n	CIFAR-10	CIFAR-100	ImageNet-100	CUB-200	Stanford-Cars
Ours (GPC)	5±1.4	22±2.6	54±3.1	110±4.2	104±3.4
Vaze *et al.* [48]	4±1.2	23±1.5	59±4.3	131±5.6	132±2.5
Ground Truth	5	20	50	100	98

F. Further comparison with ORCA

ORCA [2] is originally pretrained only on the target dataset $\mathcal{D}$, *i.e.*, the data that our model is trained on. We have shown the comparison using ImageNet pretrained features from DINO [3] for both ORCA [2] and our method in the main paper. In Tab. 16 and Tab. 17, we provide additional comparison with ORCA, showing the effects of pretrained models using different data.

Table 16: **Comparison with ORCA [2] on generic classification datasets.**

			CIFAR10			CIFAR100			ImageNet-100		
No.	Methods	Pretrain	All	Old	New	All	Old	New	All	Old	New
(1)	ORCA [2]	ImageNet	91.4	88.0	**91.2**	68.9	76.1	46.6	**79.8**	93.6	**74.9**
(2)	Ours (GPC)	ImageNet	**92.0**	**98.3**	88.7	**77.4**	**84.8**	**62.4**	76.5	**94.0**	68.5
(3)	ORCA [2]	Target	90.6	87.2	90.1	64.7	73.2	42.1	78.7	**93.4**	72.4
(4)	Ours (GPC)	Target	**91.1**	**87.8**	**90.5**	**65.0**	**74.3**	**42.6**	**79.6**	93.3	**73.1**

Table 17: **Comparison with ORCA [2] on SSB [49].**

			CUB			SCars			FGVC-Aircraft		
No.	Methods	Pretrain	All	Old	New	All	Old	New	All	Old	New
(1)	ORCA [2]	ImageNet	45.2	**57.2**	29.7	37.0	**68.2**	22.6	**47.1**	**45.3**	42.3
(2)	Ours (GPC)	ImageNet	**54.2**	54.9	**50.3**	**41.2**	58.8	**31.6**	46.1	42.4	**47.2**
(3)	ORCA [2]	Target	42.9	52.0	28.4	40.3	57.0	31.4	44.4	40.7	44.1
(4)	Ours (GPC)	Target	**45.0**	**54.2**	**29.1**	**41.2**	**57.1**	**32.1**	**46.2**	**41.0**	**45.2**

G. Qualitative results

In this section, we provide the visualization of the images grouped using the DINO features and our GPC trained features. The results are presented in each row of Fig. 4 and Fig. 5. The DINO features are effective to some extent when grouping images, but the results are still not satisfactory as the features are not tuned on the downstream tasks with a clear objective (see Fig. 4). On the other hand, after tuning the representation using our method, images from the same category can be grouped together (see Fig. 5).

Figure 4: k-means grouping of features of DINO [3] on CUB-200 dataset. Notice that the grouping are roughly based on object pose or background, but we would want the clustering to be done to discriminate between different species. The kNN images to the randomly picked prototype (*i.e.*, cluster center) are shown, from left (nearest) to right (furthest).

Figure 5: The prototype (*i.e.*, the Gaussian mean vector in our method) and the retrieved nearest neighbor on GPC representations in the CUB-200 dataset. Images are grouped by different bird species. The kNN images to the randomly picked prototype (*i.e.*, cluster center) are shown, from left (nearest) to right (furthest).

We also present t-SNE projections of the learned features on both CUB-200 and ImageNet-100 datasets. From Fig. 6 we can see that on CUB-200, the DINO features can not separate different categories very well, while our method and [48] can have a clear category boundary. From Fig. 7, we found that although the t-SNE projections on ImageNet-100 appear to be

similarly discriminative among DINO, [48], and our method, while further finetuning the representation from DINO can significantly improve the performance for the task of GCD.

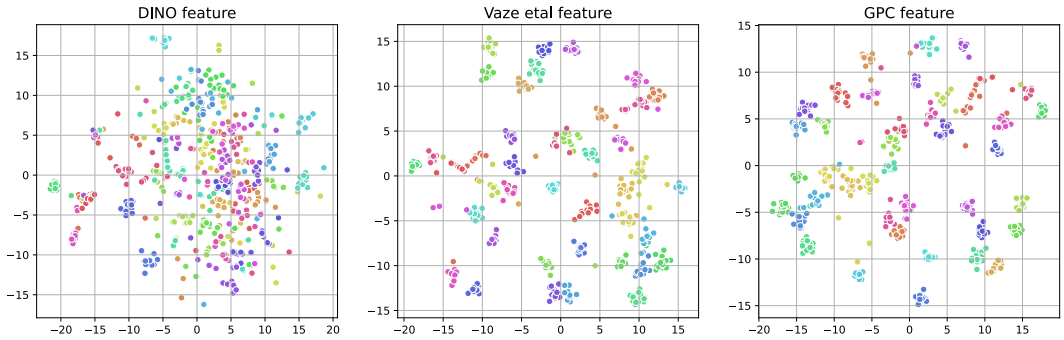

Figure 6: The t-SNE plot of the features on the CUB-200 dataset.

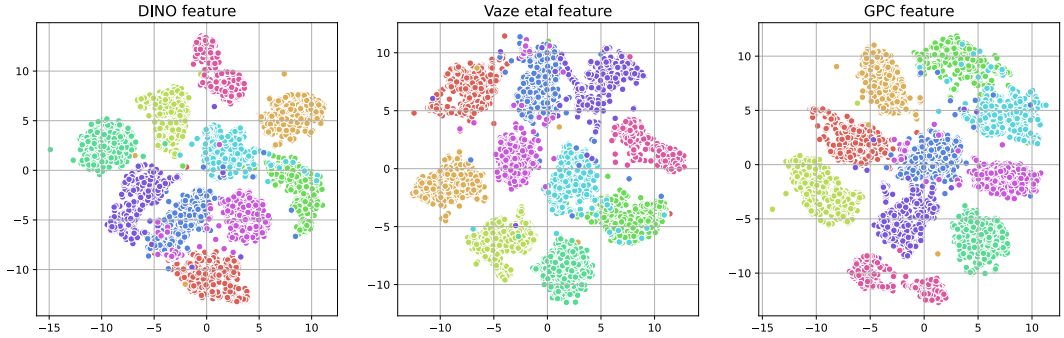

Figure 7: The t-SNE plot of the features on the ImageNet-100 dataset.

H. Limitation and negative societal impact

It should noted that although our method achieves the state-of-the-art results on the task of generalized category discovery, the classification performance is still far from those models trained with full human supervision. Furthermore, when the class number is unknown, there is still a noticeable performance gap w.r.t. the unknown category number case. Besides, real-world data is much more complex and difficult than the curated data we used. Therefore, careful validation and adaptation to specific application scenarios should be tested before deploying the model for any real-world use.

I. License of used datasets

All the datasets used in this paper are permitted for research use. CIFAR-10 and CIFAR-100 datasets [31] are released under the MIT license, allowing use for research purposes. The terms of access of the ImageNet dataset [8] allow the use for non-commercial research and educational purposes. Similar to ImageNet, the Stanford Cars [30] allows the use for research purposes. The FGVC aircraft [36] dataset was made available exclusively for non-commercial research purposes by the authors. The CUB-200 [50] dataset also allows research use.