Open-Set Representation Learning through Combinatorial Embedding
Supplementary Document

Geeho Kim[1] Junoh Kang[1] Bohyung Han[1,2]
Computer Vision Laboratory, ECE[1] & IPAI[2], Seoul National University
{snow1234, junoh.kang, bhhan}@snu.ac.kr

A. Analysis of combinatorial embeddings for novel classes

Figure 1 shows the class-wise codeword similarity between the original classes through the combinatorial embedding on CIFAR-10. When we investigate the top-2 nearest classes (red boxes in each row) from each of the novel classes (dog, deer, and automobile) on the learned embedding on CIFAR-10, semantically related classes are often located close to the novel classes. For example, the dog has the high similarity with the cat and deer (sorted in the order of similarity). Also, the automobile has high similarity with other transportations such as airplane and ship.

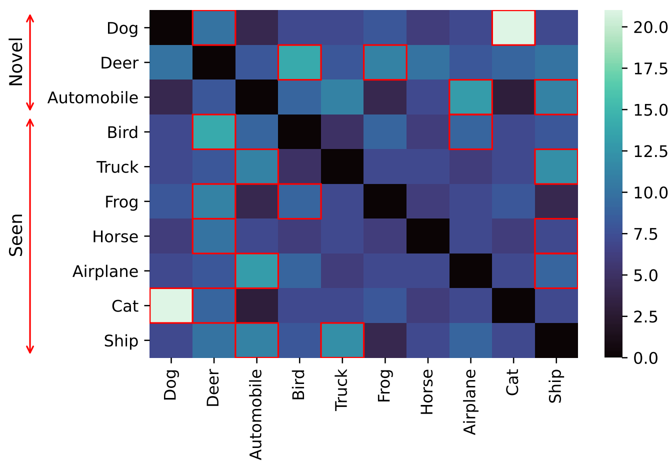


Figure 1. The class-wise codeword similarity map. We calculated the similarity between the two original classes as the following. First, using the unlabeled set, we obtain the codeword of each original class based on the index of the most frequently predicted meta-class from each meta-classifier. Then, the similarity of the two original classes' codewords can be examined by calculating the number of the same codes. The brighter the cell represents the higher the similarity, and the cells with the top-2 nearest classes are highlighted by red boxes. We leave out diagonal values to clearly show the inter-class similarities.

B. Ablation study for similarity loss

The proposed method estimates pseudo-positives based on the similarities of the combinatorial embedding vectors between each unlabeled example and the rest of the images in a batch. We then minimize the negative similarity between the feature vector of each unlabeled example in the original space and the combinatorial embedding of its positive pair, which corresponds to Eq. (6) of the main paper. To show the effectiveness of the current design of the similarity loss $\mathcal{L}_{\text{sim}}$, we evaluate the performance of the following three design variations: 1) selecting pseudo positives based on the vectors in original feature space ($\text{CombEmb}_{\text{O}\to\text{P}}$), 2) minimizing the negative similarity between the combinatorial embeddings of positive pairs ($\text{CombEmb}_{\text{C}\to\text{C}}$), and 3) minimizing the negative similarity between the combinatorial embeddings of positive pairs estimated based on vectors in original feature space ($\text{CombEmb}_{\text{O}\to\text{C}}$). Tab. 1 presents that the proposed pseudo-label consistency loss is more effective than other options. The result implies that estimating pairwise labels based on combinatorial embeddings are more robust than using the vectors in the original feature space. Note that learning the deep feature representations using the combination of the original and combinatorial embedding spaces contributes to additional performance gain.

Table 1. mAP scores of different strategies for pseudo-label consistency on CIFAR-10.

Ablation	$\mathcal{P}$	$\mathcal{L}_{\text{sim}}$	CIFAR-10 12bits	24 bits	48 bits
CombEmb (ours)	Combinatorial	Proposed	**0.667 ± 0.049**	**0.692 ± 0.047**	**0.720 ± 0.029**
$\text{CombEmb}_{\text{O}\to\text{P}}$	Original	Proposed	0.597 ± 0.060	0.690 ± 0.025	0.675 ± 0.053
$\text{CombEmb}_{\text{C}\to\text{C}}$	Combinatorial	Combinatorial	0.635 ± 0.053	0.688 ± 0.072	0.675 ± 0.053
$\text{CombEmb}_{\text{O}\to\text{C}}$	Original	Combinatorial	0.637 ± 0.057	0.666 ± 0.030	0.655 ± 0.045

C. Ablation study on meta-class set construction methods

As shown in Table 2, the combEmb outperforms GPQ [6] even with randomly constructed meta-class sets (Random), which implies that the combinatorial embedding in the proposed approach learns basic representation suitable for both seen and unseen examples. Moreover, the clustering-based algorithm (Clustering) brings additional improvement regardless of the bit-lengths. Note that the clustering based algorithm achieves more accuracy gain than Random at larger bit-lengths, which implies that using the meta-class sets based on the clustering better capture the inter-class relations and improve the representations of combinatorial embeddings.

Table 2. mAP scores on CIFAR-10 with different meta-class configurations.

Method	12 bits	24 bits	48 bits
GPQ	0.274	0.290	0.313
Random	0.556	0.660	0.656
Clustering	**0.667**	**0.692**	**0.720**

D. Implementation details

Our core algorithm is developed using Pytorch [8] and we conduct all experiments with either NVIDIA TITAN XP or NVIDIA TITAN V.

Image retrieval To simulate the open-set environment, we set up benchmark datasets following the protocol in [9]. Specifically, we divide the classes into 75% seen and 25% novel classes on all datasets. For training, we randomly select half of the examples from the seen classes to construct a labeled set, and use the rest of the seen class and half of the examples from the novel classes to construct an unlabeled set. For retrieval, we use the rest of novel classes as a query while using the unlabeled set as a database.

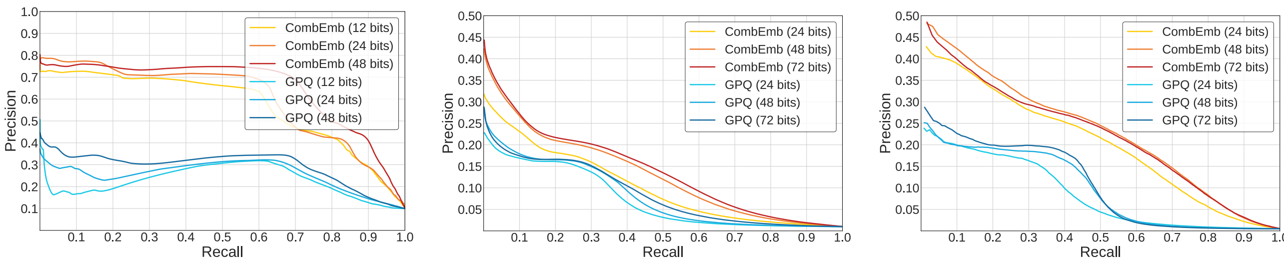
With regard to network training, we use the simple data augmentation with only random crop and horizontal flip. We adopt the AdamW [7] algorithm to optimize the network, and finetune our model for 200 epochs for all datasets with the exponential learning rate decay of the initial value 0.0002 for all benchmarks, except NUS-WIDE of which the learning rate is set to 0.00001. The learning rate for meta-classifiers is scaled up 100 times to stabilize the convergence. We use the batch size of 512 / 256 on $\{$CIFAR-10, CIFAR-100, Tiny-ImageNet$\}$ / $\{$NUS-WIDE, CUB200$\}$ with the same numbers of labeled and unlabeled images. Furthermore, following the standard practice in self-supervised learning [2, 3], we use the same projection head as in [2] for h in (7) of the main paper. we employ the modified VGG network for CIFAR-10/CIFAR-100, AlexNet for NUS-WIDE, and ResNet-18 for CUB200 as the feature extractor.

The hyperparameters specific to our algorithm are set as follows. Q for meta-class set generation is set to 50, λ is set to 20, and γ in (5) of the main paper is set to 0.95 for all benchmarks, except for CIFAR-10, which is trained with 0.85. τ in (4) of the main paper is set as 0.04 except for CIFAR-10 and CIFAR-100, which is set as 0.2 and 0.1, respectively. α and β in (8) are set as 0.16 and 12.5 in all experiments. We decide the algorithm-specific parameters to achieve the best mAP scores on the validation set. For GPQ, we follow the original implementation[1] and learning protocols as far as possible, referring to the original paper [6] for details.

Image categorization with novel class discovery All algorithms employ the ResNet-18 pre-trained with SimCLR while DTC [5], RankStats [4], and NCL [13] have an additional fine-tuning procedure with the labeled data. For our method, we maintain the same learning strategy and hyperparameters as the experiments in image retrieval for fine-tuning the model, with the best model selected using accuracy on the validation set. For DTC, RankStats, NCL, DualRank [12], ORCA [1], and GCD [11], we reimplemented them based on the original implementations[2][3][4][5][6][7] and follow the learning protocols from the original papers.

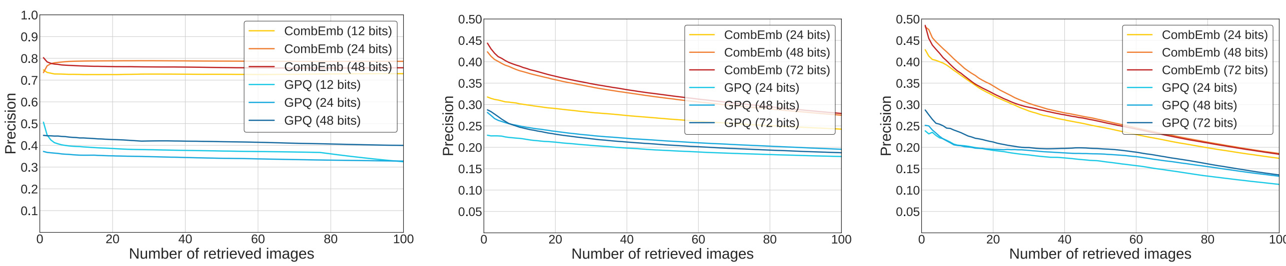
E. More comprehensive results of CombEmb for image retrieval

In Figure 2 and 3, the effectiveness of CombEmb is further verified in various bit-lengths. In all cases, we observe that CombEmb consistently shows higher precision than GPQ, regardless of recall level. Note that the area under the precision-recall curve corresponds to the mAP score.



(a) CIFAR-10 (b) CIFAR-100 (c) CUB-200

Figure 2. Precision-Recall curves on CIFAR-10, CIFAR-100, and CUB200 with various bit-lengths.



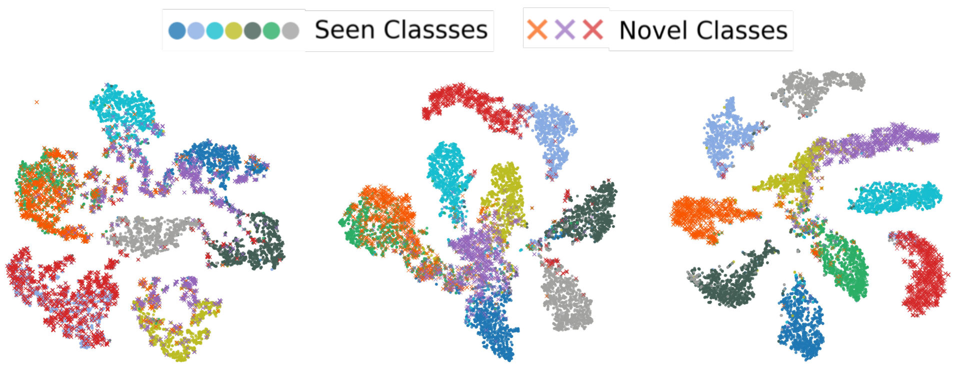
(a) CIFAR-10 (b) CIFAR-100 (c) CUB-200

Figure 3. Precision@100 curves on CIFAR-10, CIFAR-100, and CUB200 with various bit-lengths.

F. Visualization of embedding for image retrieval

Figure 4 visualizes the embeddings learned by GPQ, k-means, and the proposed method on CIFAR-10. According to the figure, GPQ fails to learn the distinct representations of novel classes; it tends to align the novel classes to the closest seen classes. The pseudo-labeling method given by k-means clustering with the oracle value of k also has trouble in learning the discriminative representation of the examples in unseen classes. To the contrary, the proposed approach learns the representations, effectively discriminating both known and novel classes, through supervised combinatorial classification followed by unsupervised pairwise relation learning.

[1]https://github.com/youngkyunJang/GPQ
[2]https://github.com/k-han/DTC
[3]https://github.com/k-han/AutoNovel
[4]https://github.com/zhunzhong07/NCL
[5]https://github.com/DTennant/dual-rank-ncd
[6]https://github.com/snap-stanford/orca
[7]https://github.com/sgvaze/generalized-category-discovery



(a) GPQ (b) k-means (c) CombEmb (ours)

Figure 4. t-SNE visualization of CIFAR-10 using VGG, learned by GPQ, k-means, and the proposed method for image retrieval task. Visualization is based on 7 seen classes and 3 novel classes on CIFAR-10. Colors represent their ground-truth labels. Note that the proposed method embeds known and novel classes in a more discriminative way than other baselines.

References

[1] Kaidi Cao, Maria Brbic, and Jure Leskovec. Open-world semi-supervised learning. In *ICLR*, 2022. 3

[2] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR*, 2021. 2

[3] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. In *NeurIPS*, 2020. 2

[4] Kai Han, Sylvestre-Alvise Rebuffi, Sebastien Ehrhardt, Andrea Vedaldi, and Andrew Zisserman. Automatically discovering and learning new visual categories with ranking statistics. In *ICLR*, 2020. 3

[5] Yen-Chang Hsu, Zhaoyang Lv, and Zsolt Kira. Learning to cluster in order to transfer across domains and tasks. In *ICLR*, 2018. 3

[6] Young Kyun Jang and Nam Ik Cho. Generalized product quantization network for semi-supervised image retrieval. In *CVPR*, 2020. 2, 3

[7] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 2

[8] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 2

[9] Alexandre Sablayrolles, Matthijs Douze, Nicolas Usunier, and Hervé Jégou. How should we evaluate supervised hashing? In *ICASSP*, 2017. 2

[10] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11), 2008.

[11] Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Generalized category discovery. In *CVPR*, 2022. 3

[12] Bingchen Zhao and Kai Han. Novel visual category discovery with dual ranking statistics and mutual knowledge distillation. In *NeurIPS*, 2021. 3

[13] Zhun Zhong, Enrico Fini, Subhankar Roy, Zhiming Luo, Elisa Ricci, and Nicu Sebe. Neighborhood contrastive learning for novel class discovery. In *CVPR*, 2021. 3