Supplementary Material
GKGNet: Group K-Nearest Neighbor based Graph Convolutional Network for Multi-Label Image Recognition

Ruijie Yao[1,4], Sheng Jin[2,4], Lumin Xu[3], Wang Zeng[4], Wentao Liu[4], Chen Qian[1,4] ✉, Ping Luo[2,5], and Ji Wu[1] ✉

[1] Tsinghua University [2] The University of Hong Kong [3] The Chinese University of Hong Kong [4] SenseTime Research and Tetras.AI [5] Shanghai AI Laboratory
{yrj21@mails, qianc18@mails, wuji_ee@mail}.tsinghua.edu.cn

S1 Implementation Details in General Image Classification

In Table 5 of the main text, we demonstrate the efficiency of Group KNN in general image classification, highlighting its robust performance in graph construction. We maintain uniform training settings for both models, with the only difference being the use of our proposed Group KNN.

We adhere to widely-used settings for each dataset, and specific details for each dataset are provided below:

ImageNet-1K We follow the training settings of ViG [2], keeping hyperparameters consistent. Data augmentation involves Mixup and CutMix techniques, and input images are resized to 224×224. The AdamW optimizer is used with parameters: learning rate (lr) of 0.002, weight decay of 0.05, and a cosine learning rate update policy. The training process spans up to a maximum of 300 epochs.

CIFAR-10 and CIFAR-100 Given our adoption of $K = 9$ for KNN graph construction, the standard 32×32 input size in CIFAR-10 and CIFAR-100 doesn't provide sufficient patches in the last three stages. Therefore, we modify the input size to 224×224, consistent with ImageNet-1K. The remaining training configurations align with MMClassification [1]. Data augmentation includes image resizing and random flipping. We use the SGD optimizer with parameters: learning rate (lr) of 0.1, weight decay of 0.0001, and a step learning rate update policy that reduces the learning rate at epochs 100 and 150. The training process continues for a maximum of 200 epochs.

Flowers We set the image size to 224×224. Data augmentation includes image resizing and random flipping. The SGD optimizer is used with parameters: learning rate (lr) of 0.1, weight decay of 0.0001, and a step learning rate update policy that reduces the learning rate at epochs 30, 60, and 90. The training process spans up to a maximum of 100 epochs.

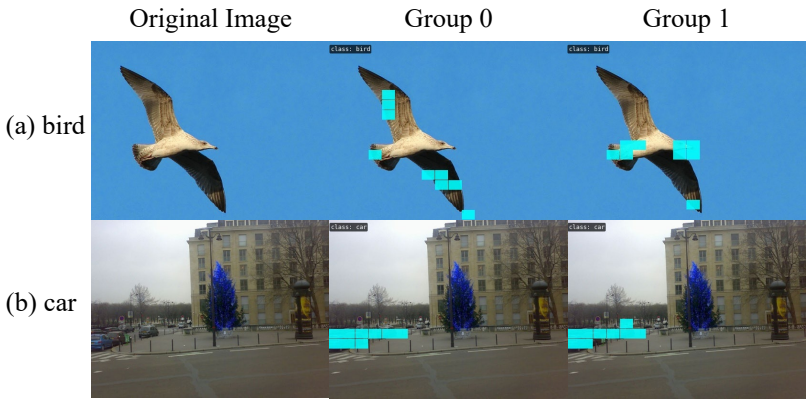

Fig. S1: Visualization of the learned connections between patch nodes and label nodes for **objects of varying sizes**. (a) and (b) correspond to "bird" and "car" respectively. The azure-colored blocks in the figure are the patch nodes nearest to the label nodes.

S2 More Qualitative Analysis

S2.1 Learned connections between the label node and the patch nodes in Group KGCN module

In the section, we will show more visualization of the learned connections between patch nodes and label nodes in the last GKG block to demonstrate the effectiveness of our model on the MLIR task.

Objects of varying sizes. As shown in Fig. S1 (a), "bird" occupies a large region in images, and using a fixed small number of K can hardly cover enough patches to obtain sufficient distinguished features. However, our Group KGCN module learns to pay attention to different areas in different groups, *e.g.* the body and wings . The label "bird" selects 17 different patch nodes, which are close to the maximum $K \times G = 18$ patches, to utilize features from enough parts and avoid information loss. As shown in Fig. S1 (b), "car" only occupies a small area in the images. If we extract global features by global pooling in CNNs or global attention in Transformers, most features will come from the background regions which will hinder feature representation learning. In contrast, our Group KGCN learns to select two groups of neighbors that have a large overlap with each other, focusing on the foreground objects and eliminating background distractions. As a result, the label "car" selects 10 different patch nodes, which are close to the minimum $K = 9$ patches, to avoid background interference.

Co-occurring Categories. Modeling multi-label correlations are critical for MLIR tasks, especially when the corresponding areas of the target label are occluded or unclear. As shown in Fig. S2(a), the neighbor patch nodes of "sofa" in the

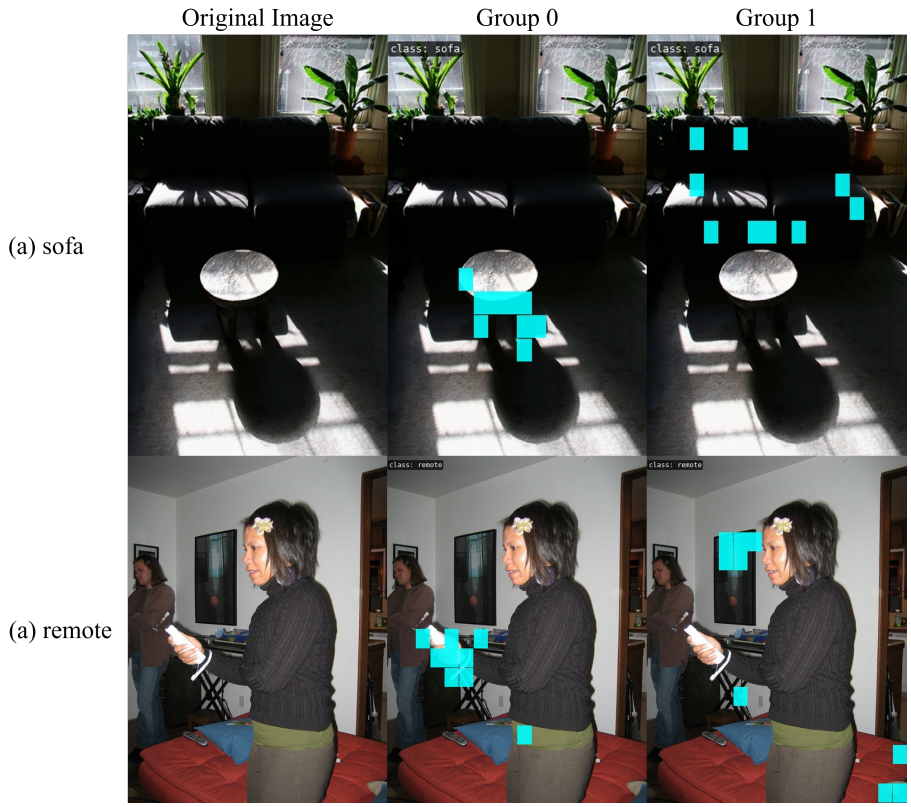

Fig. S2: Visualization of the learned connections between patch nodes and their label nodes for **co-occurring categories**. (a) and (b) correspond to "sofa" and "remote" respectively. The two groups of nodes have found the patches corresponding to their own and their correlated classes respectively.

first group precisely locate the sofa which is in the dark, and the second group pays attention to the table in front of the sofa. In Fig. S2(b), the first group of "remote" concentrates on the area partly covered by a person's hand where the remote is located, while the other group focuses on the TV. Capturing the features from co-occurring labels with our multi-group design will provide more robust relevant information which helps to understand the whole scene.

Spatially Distributed Objects. Objects are commonly spatially distributed in the image. As shown in Fig. S3, there are two people in Fig. S3(a) and three cows in Fig. S3(b), and all objects corresponding to the target labels are selected by two groups, which demonstrates our Group KGCN module can effectively extract features from the long-range area.

Fig. S3: Visualization of the learned connections between patch nodes and label nodes for **spatially distributed objects**. (a) and (b) correspond to "person" and "cow" respectively. The two groups of nodes completely found multiple objects related to the label in both (a) and (b).

Multiple labels. For MLIR, it is very common that multiple labels occur in a single image. As shown in Fig. S4, both two images have two categories. And each label node exactly finds its related image patches without confusion, which validates the effect of our fully graph network in unifying the representations of patch nodes and label nodes.

S3 Detailed Results on MS-COCO Dataset

In Table S1, we report and compare the per-category recognition accuracy on MS-COCO dataset [3]. We observe that our model achieves the best performance for 76 out of 80 categories and achieve very competitive results for the rest 4 categories. Especially, for small and challenging object categories, *e.g.* hair drier, scissors, apple and toothbrush, the performance improvements are more significant.

References

1. Contributors, M.: Openmmlab's image classification toolbox and benchmark. `https://github.com/open-mmlab/mmclassification` (2020)
2. Han, K., Wang, Y., Guo, J., Tang, Y., Wu, E.: Vision gnn: An image is worth graph of nodes. In: Adv. Neural Inform. Process. Syst. (2022)
3. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Eur. Conf. Comput. Vis. pp. 740–755 (2014)

	person	bicycle	car	motorcycle	airplane	bus	train	truck
TDRG	99.2	81.6	89.9	94.6	97.1	89.6	96.7	77.1
Q2L	**99.3**	81.9	89.9	94.4	97.5	89.7	96.7	77.3
GKGNet	**99.3**	**88.9**	**93.2**	**96.4**	**98.0**	**91.8**	**98.1**	**81.4**
	boat	traffic light	fire hydrant	stop sign	parking meter	bench	bird	cat
TDRG	91.3	86.6	85.8	80.3	71.8	69.0	83.6	96.8
Q2L	91.1	86.9	86.0	81.1	72.7	70.2	84.0	97.1
GKGNet	**93.2**	**91.6**	**90.4**	**85.8**	**77.2**	**74.5**	**87.5**	**97.2**
	dog	horse	sheep	cow	elephant	bear	zebra	giraffe
TDRG	91.2	94.4	96.5	93.3	98.6	98.0	99.4	99.7
Q2L	92.0	95.3	96.5	93.9	98.8	**98.6**	99.5	**99.8**
GKGNet	**94.6**	**96.3**	**96.8**	**94.8**	**99.0**	97.8	**99.6**	**99.8**
	backpack	umbrella	handbag	tie	suitcase	frisbee	skis	snowboard
TDRG	55.2	87.4	57.8	86.1	76.5	94.9	95.3	87.4
Q2L	54.1	88.2	57.7	86.2	77.6	94.9	96.1	87.0
GKGNet	**59.4**	**90.5**	**62.9**	**88.1**	**81.9**	**97.1**	**96.6**	**89.5**
	sports ball	kite	baseball bat	baseball glove	skateboard	surfboard	tennis racket	bottle
TDRG	88.8	97.6	95.4	96.1	97.6	96.7	98.9	76.3
Q2L	89.2	98.3	96.5	**96.7**	98.0	96.5	99.2	76.9
GKGNet	**92.4**	**98.6**	**96.6**	96.5	**98.5**	**97.8**	**99.3**	**82.1**
	wine glass	cup	fork	knife	spoon	bowl	banana	apple
TDRG	80.9	78.1	80.3	71.4	67.5	76.2	87.5	71.4
Q2L	80.6	79.3	82.4	72.9	68.5	76.4	87.7	72.3
GKGNet	**86.7**	**82.9**	**85.7**	**77.4**	**72.1**	**81.4**	**92.9**	**79.9**
	sandwich	orange	broccoli	carrot	hot dog	pizza	donut	cake
TDRG	77.3	82.5	93.2	81.5	80.5	95.1	86.7	84.9
Q2L	78.2	83.3	93.8	82.2	82.0	95.6	86.0	84.6
GKGNet	**81.8**	**88.3**	**94.5**	**86.3**	**85.1**	**95.9**	**88.9**	**88.2**
	chair	couch	potted plant	bed	dining table	toilet	tv	laptop
TDRG	80.8	84.9	71.6	89.0	80.4	97.8	89.5	91.0
Q2L	82.0	85.5	72.7	89.9	82.1	98.0	90.5	90.6
GKGNet	**84.6**	**87.0**	**76.9**	**90.5**	**82.2**	**98.2**	**91.5**	**93.1**
	mouse	remote	keyboard	cell phone	microwave	oven	toaster	sink
TDRG	89.9	82.2	89.2	70.0	82.4	88.8	**40.8**	92.2
Q2L	90.6	82.2	89.9	70.0	**84.9**	90.1	35.7	92.6
GKGNet	**92.2**	**86.4**	**91.3**	**74.5**	84.7	**90.7**	40.0	**93.4**
	refrigerator	book	clock	vase	scissors	teddy bear	hair drier	toothbrush
TDRG	82.8	70.7	84.2	79.9	65.5	88.0	45.1	76.5
Q2L	82.7	71.7	84.6	81.0	68.1	89.2	43.5	75.4
GKGNet	**86.4**	**77.1**	**87.6**	**83.0**	**76.4**	**91.1**	**62.0**	**80.6**

Table S1: AP for each category obtained by competing methods on MS-COCO dataset. The best scores are highlighted in bold.

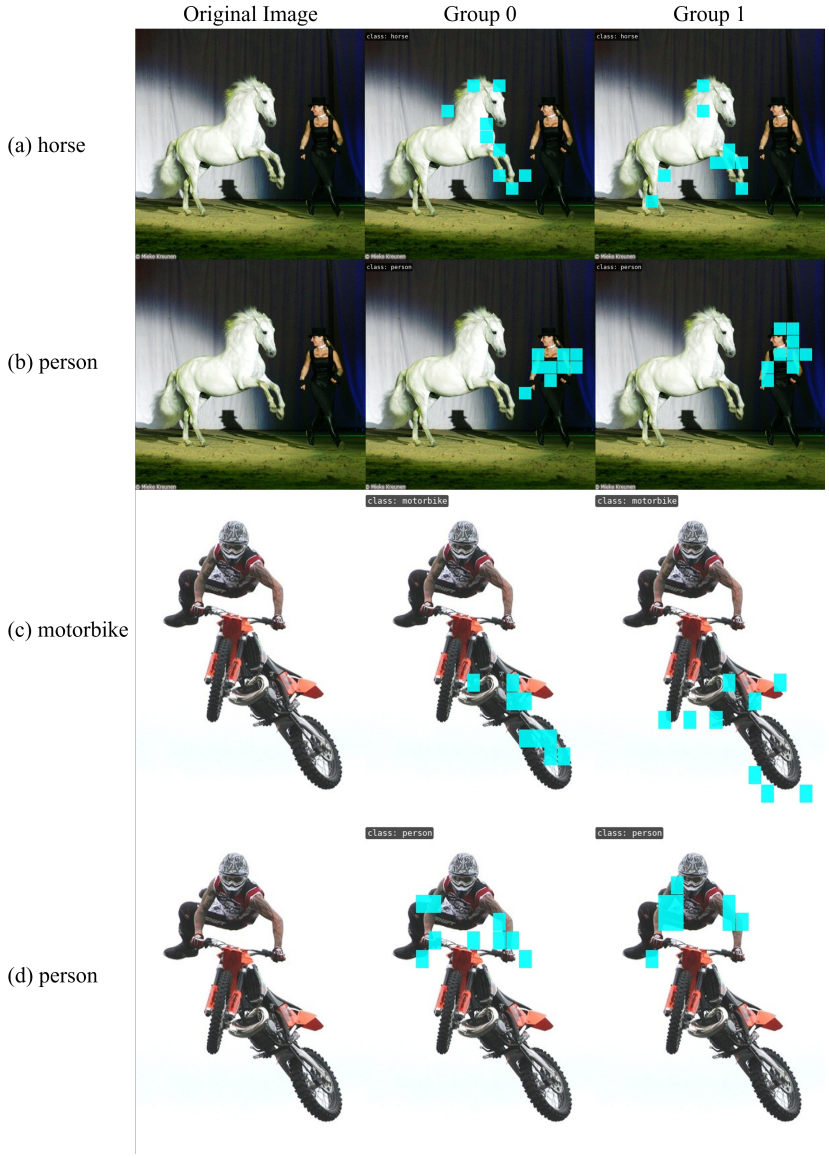

Fig. S4: Visualization of the learned connections between patch nodes and label nodes, where there are **multiple labels in one image**. Every two figures refer to two labels in the same image. Each label node has found its correct corresponding patch areas in the case where multiple classes exist simultaneously.