ML-VIG: MULTI-LABEL IMAGE RECOGNITION WITH VISION GRAPH CONVOLUTIONAL NETWORK

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-Label Image Recognition (MLIR) aims to predict multiple object labels in a single image. Graph representations have been used to model label correlation or visual relationships separately. However, the representations of label embeddings and visual features are not well aligned, which hinders effective representation learning and leads to inferior performance. In this work, we propose the first fully graph convolutional model, termed Multi-Label Vision Graph Convolutional Network (ML-ViG), for the task of MLIR. ML-ViG unifies the representation of visual features and label embeddings, enabling the graph structures to capture the (1) spatial relationship among visual region features, (2) semantic relationship among object labels, and (3) cross-level relationship between labels and regions. In order to effectively pass messages between visual features and labels, Multi-Label Graph Convolutional Network (MLG) module is proposed. ML-ViG achieves state-of-the-art performance with significantly lower computational costs on MS-COCO, VOC2007, and VG-500 datasets. Codes and models will be released.

1 INTRODUCTION

Multi-label image recognition (MLIR) (also referred to as multi-label classification) is a fundamental task in computer vision, which aims to predict a set of labels of a single image. Compared with single-label image recognition, MLIR is more challenging due to its combinatorial nature. It has received great attention because of its broad real-world applications, such as human attribute recognition (Li et al., 2016) and scene understanding (Shao et al., 2015).

It is common that some related objects may co-occur in the real world. For example, the tennis racket often occurs along with the sports ball. So capturing such label correlation (or label dependencies) is key for MLIR. Inspired by the success of graph models in relationship modeling, many works (Ye et al., 2020; Chen et al., 2019c; Wang et al., 2020; Zhao et al., 2021) utilize Graph Convolutional Network (GCN) to capture the label correlation as shown in Figure 1a. In this line of works, Convolutional Neural Network (CNN) is applied to predict the multi-labels, and GCN is adopted for message passing among these labels to refine the predictions. However, these works mainly focus on capturing semantic relations among labels and ignore the spatial relations inherent in the visual features. In addition, the representations of visual features and those of labels are not aligned and are processed individually, which hinders integral representation learning and results in limited performance.

More recently, the graph representation of images (Han et al., 2022) has attracted increasing research attention. As shown in Figure 1b, in the graph structure, image patches are viewed as graph nodes and the relationship and inter-dependencies between image patches are represented by graph edges. As the nodes are linked by content instead of by spatial position, ViG (Han et al., 2022) avoids the inductive biases in CNNs and is able to capture global and wider range relations among regions. However, as ViG is specially designed for single-label image recognition, it only explores the spatial relationship among visual region features without considering the semantic relationship among object labels.

In this paper, we propose the first fully graph convolutional network (GCN) for the task of multi-label image recognition, termed Multi-Label Vision Graph Convolutional Network (ML-ViG). ML-ViG utilizes effective and flexible graph representations for both representation learning and correlation extraction. As shown in Figure 1c, ML-ViG simultaneously captures three kinds of relationships:

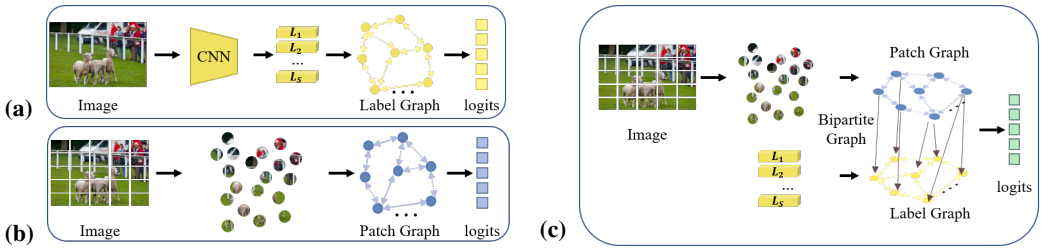


Figure 1: Different graph-based methods to solve MLIR task. (a) generates label embeddings via CNN and then further updates the label embeddings in the following GCN modules, where labels are treated as nodes in graph (Ye et al., 2020; Zhao et al., 2021; Wang et al., 2020; Chen et al., 2019c). (b) views image patches as nodes, then visual features are updated through GCN (Han et al., 2022). The spatial relationship among image patches is modeled, but the relationships between labels are unexplored. (c) is our work. We integrate spatial patch nodes and semantic label nodes in a unified graph, which takes into account both visual representation learning and label correlation learning.

(1) spatial relationship among visual region features, (2) semantic relationship among object labels, and (3) cross-level relationship between labels and regions. Specifically, our unified graph representations include two types of nodes, *i.e.* patch node (visual features of image patches) and label node (label embeddings). Three types of connections, *i.e.* patch-to-patch, patch-label, and label-to-label are explored by ViG block (Han et al., 2022) , Patch-Label GCN (PLG), and Label-Label GCN (LLG) respectively. An image can be viewed as a composition of multiple objects distributed in different spatial locations. ViG block (Han et al., 2022) constructs the graph of image patches and performs information exchange to capture spatial correlation of visual features. PLG dynamically constructs connections between patch nodes and label nodes to learn to locate the spatial locations of the target labels. With effective message passing, it explicitly learns to extract category-correlated interest region features. LLG creates the label correlation matrix to guide information propagation among the label embeddings.

Extensive experiments on several well-known benchmarks, *i.e.* MS-COCO, VOC2007, and VG-500, verify the effectiveness of the proposed method. Our proposed method achieves new state-of-the-art performance with significantly lower computational costs.

This paper's contribution can be summarized as follows:

- We propose Multi-Label Vision Graph neural network (ML-ViG for short) to build unified graph representations for both visual features and label embeddings. To our best knowledge, it is the first to successfully apply a fully graph convolutional model for the task of multi-label image recognition.
- MLG is proposed to explicitly model the relationship between labels and visual regions (PLG), and the correlations among labels (LLG) in a unified graph convolutional way.
- We validate the effectiveness of the proposed model on three widely used benchmark datasets, including MS-COCO, VOC2007, and VG-500. We show that our model consistently outperforms the previous state-of-the-art approaches with significantly lower computational complexity.

2 Related Work

2.1 Graph Neural Network for Vision Tasks

Graph is a flexible data structure and it can process any kind of data that can be converted into a set of nodes and edges. For example, non-euclidean data like a social network and euclidean data like images can be viewed as graph (Zhou et al., 2020). Graph Convolutional Network (GCN) has shown great effectiveness in representation learning for graph-structured data. Especially, GCN has been widely applied for message passing and correlation modeling in many computer vision tasks, *e.g.* multi-label classification (Chen et al., 2019c), scene graph generation (Zhu et al., 2022) and

human action recognition (Yan et al., 2018; Jain et al., 2016). However, these works only explore the semantic relationship, while ignoring the spatial relationship among the regional visual features. Recently, Han et al. (2022) proposes ViG to directly convert an image to the graph structure and learn visual representations. ViG divides an image into sets of patches and treats these patches as nodes in the graph. The graph is built by finding K-nearest neighbors for each patch node. With effective message passing, the spatial relationship of these visual patches is captured. Previous works have been independently explored in graphs on semantic space or spatial space, but how to unify these two kinds of graphs is still unexplored, which is a crucial question for the MLIR task. In this work, we design a graph convolutional network considering connections among spatial regions, connections among label semantic embeddings, and connections between regions and labels, which is key to multi-label classification.

2.2 Multi-Label Image Recognition

Multi-label image recognition (MLIR) (also referred to as multi-label classification) is an extension of single-label image recognition. MLIR aims to predict multiple mutually non-exclusive class labels for an input image. CNN (Convolutional Neural Network) serves as the standard network model for MLIR and many efforts have been dedicated (He et al., 2016; Zhu et al., 2017; Jia et al., 2021) to designing powerful CNN architectures for better performance. They view the task of MLIR as multiple binary classification tasks, and to predict all labels independently. Such approaches are simple and straightforward, however, they do not take into account the semantic relationship among the labels (or label co-occurrence). For example, some combinations of labels are very common (*e.g.* 'person' and 'tie') while some are rare (*e.g.* 'zebra' and 'train'). Such kinds of relationships can be important regularizers for our model. Recently, some works (Lanchantin et al., 2021; Liu et al., 2021; Cheng et al., 2022) explore to use Transformer based model for the task of MLIR to capture such kinds of relationship. Lanchantin et al. (2021) first apply the transformer encoder to multi-label classification and Liu et al. (2021) further add the transformer decoder to improve the model performance. Cheng et al. (2022) explore a convolutional free Transformer model to address the task of MLIR. These works typically require large computational costs, and especially the costs will increase rapidly as the image resolution increases.

Our work is mostly related to the GCN-based models (Chen et al., 2019c; Wang et al., 2020; Ye et al., 2020; Zhao et al., 2021). They propose to use CNN to generate semantic embeddings for each label and use GCN to model the label relationship, combining the benefits of both CNN and GCN to achieve better results. However, current GCN methods do not adequately explore the spatial relationship of region vision features and the interest region feature extraction. To handle this, we instead propose a fully graph convolutional network and learn unified graph representations for both visual features and label embeddings.

3 Method

3.1 Overview

We propose a fully graph convolutional network (GCN), termed Multi-Label Vision GCN (ML-ViG), for the task of multi-label image recognition. ML-ViG captures three kinds of relationships including visual patch connections, label connections, and patch-label connections to jointly predict multiple labels. As shown in Figure. 2, ML-ViG consists of multiple stages, and each stage includes L Vision GCN (ViG) blocks and M Multi-Label GCN (MLG) blocks. The input image is represented as patch nodes and is processed by ViG blocks. Then patch nodes and label nodes (represented by learnable label embeddings) will be sent to Multi-Label GCN (MLG) blocks to capture multi-label correlation and exchange information between visual features and label embeddings. After multi-stage refinement, the outputs of the ViG blocks and MLG blocks are combined together to predict the possibilities of categories. The technical details of ViG blocks are introduced in Section 3.2 and those of MLG blocks are presented in Section 3.3.

3.2 Recap: Vision GCN (ViG)

Vision Graph Convolutional Network (ViG) (Han et al., 2022) views a single image as the graph structure, which is a more flexible representation than the original grid structure. Instead of treating

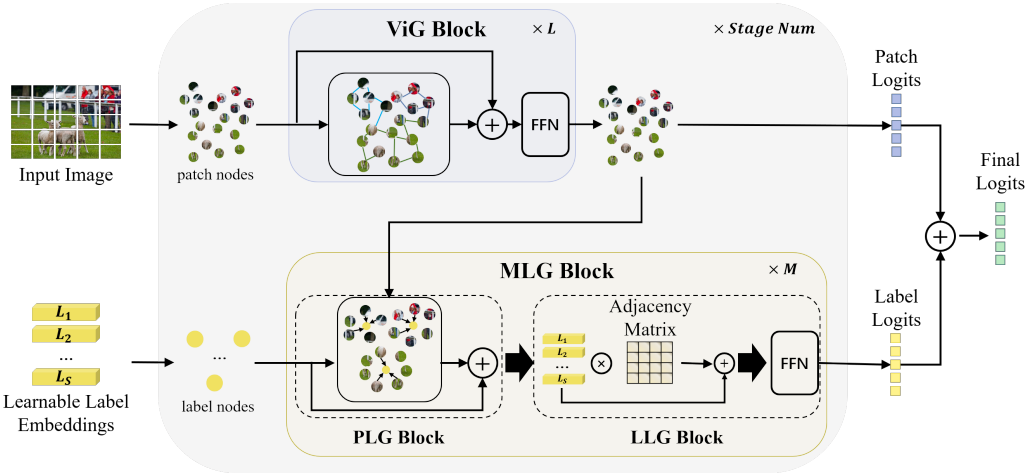


Figure 2: **Overview of ML-ViG.** An input image is firstly divided into a set of patches. These patches are viewed as nodes and their connections are learned with K-nearest neighbors (KNN) in ViG. After patch nodes are updated via GCN in ViG blocks, they will be input to MLG blocks together with label nodes. In MLG blocks, each label node will also connect patch nodes by KNN and extract features from these patch nodes, thus cross-level relation between semantic labels and spatial regions is constructed in PLG blocks. And label nodes will be updated with a learned adjacency matrix in LLG blocks. There are 4 stages in ML-ViG, and each stage consists of L ViG blocks and M MLG blocks, where the specific values of L and M at each stage are different. The outputs of the final ViG blocks and MLG blocks are combined together to predict the possibilities of categories.

each pixel as a graph node, which introduces a huge computational cost, the input image is divided into N patches. Each patch is represented by a feature vector via a fully-connected layer and is viewed as a patch node. Each patch node searches its K_{vig} nearest neighbors measured by cosine similarity and the edges are constructed among them. ViG blocks are applied to process the constructed graph structure, in which max-relative graph convolution (Li et al., 2019) is adopted. Feed-forward network (FFN) and the residual structure are used to relieve the over-smoothing problem in GCN. Compared to convolutional neural network (CNN), the receptive field of ViG is expanded to the whole image in theory by message passing among the nodes of the graph. Global context information can be captured to establish long-range dependencies, which is especially important for the task of multi-label classification, where the regions of interest are distributed in the image.

3.3 Multi-Label GCN (MLG)

Each target label may be related to several regions of interest, so capturing the relationship between labels and regions is critical for multi-label image recognition. Also, the label correlation and semantic relationship between multiple labels are expected to be modeled. Therefore, we propose the Multi-Label GCN (MLG) to capture the connections, where Patch-Label GCN (PLG) is designed for the relationship between visual regions and target labels, and Label-Label GCN (LLG) is proposed for the multi-label correlation.

3.3.1 Patch-Label GCN (PLG)

PLG block captures the relationship between visual patches and multiple labels. The nodes of the graph are represented by patch features and label embeddings, where patch nodes and label nodes are $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N\} \in \mathbb{R}^{N \times C}$ and $\boldsymbol{H} = \{\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_S\} \in \mathbb{R}^{S \times C}$ respectively. C is the dimension of patch features and label embeddings, and S is the number of labels. Each label node connects to its K_{plg} nearest neighbors in patch nodes measured by cosine distance. The label node $\boldsymbol{h}_i$ is updated by aggregating the features of the most related patch regions as follows:

$$\hat{\boldsymbol{h}}_i = concat(\boldsymbol{h}_i,\ max(\{\boldsymbol{h}_i - \boldsymbol{x}_j | j \in \mathcal{N}(\boldsymbol{h}_i)\})), \tag{1}$$

$$\boldsymbol{h}_i^{'} = \boldsymbol{h}_i + \hat{\boldsymbol{h}}_i \boldsymbol{W}. \tag{2}$$

where $\mathcal{N}(\boldsymbol{h}_i)$ is the set of nearest neighbors of the label node $\boldsymbol{h}_i$, *i.e.* there exists edges between the label node $\boldsymbol{h}_i$ and the patch node $\boldsymbol{x}_j$ for $j \in \mathcal{N}(\boldsymbol{h}_i)$. They are integrated by maximizing every dimension of $\boldsymbol{h}_i - \boldsymbol{x}_j \in \mathbb{R}^{1\times C}(j \in \mathcal{N}(\boldsymbol{h}_i))$ among all the K_{plg} neighbors following max-relative graph convolution (Li et al., 2019). $\boldsymbol{W} \in \mathbb{R}^{2C\times C}$ is the learnable update matrix. The label embedding is updated via the message passing from the corresponding visual features, which builds the connections between image regions and target labels. The residual structure is applied to preserve the diversity of the label embedding and avoids over-smoothing. PLG allows the label embedding to select the most distinctive visual features of local regions, reducing the interference of invalid background, as shown in Section 4.4.

3.3.2 LABEL-LABEL GCN (LLG)

Recent work (Wang et al., 2020; Chen et al., 2019c; Ye et al., 2020; Zhao et al., 2021) have proven the adjacency matrix is useful in explicitly modeling inter-label occurrence. We propose LLG to capture the connections between different labels. The updated label embeddings are used as graph nodes. They are fully connected, where there exists an edge between any pair of nodes. Each label node is further updated by aggregating the features of the associated label nodes.

$$\bar{\boldsymbol{H}} = \boldsymbol{A}\boldsymbol{H}^{'} + \boldsymbol{H}^{'}, \tag{3}$$

where $\boldsymbol{H}^{'} = [\boldsymbol{h}_1^{'}, \boldsymbol{h}_2^{'}, \cdots, \boldsymbol{h}_S^{'}] \in \mathbb{R}^{S\times C}$ is the matrix that concatenates the updated label embeddings output by PLG. $\boldsymbol{A} \in \mathbb{R}^{S\times S}$ is the learnable weights with random initialization. $\bar{\boldsymbol{H}} = [\bar{\boldsymbol{h}}_1, \bar{\boldsymbol{h}}_2, \cdot \cdot \cdot, \bar{\boldsymbol{h}}_S] \in \mathbb{R}^{S\times C}$ indicates the concatenation of S refined label node $\bar{\boldsymbol{h}}_i$. The label correlation is jointly learned during the training of $\boldsymbol{A}$. Then, a feed-forward network (FFN) consisting of two simple fully-connected layers is applied to encourage non-linear interaction among multiple labels.

3.4 CLASSIFIER AND LOSS FUNCTION

The classifier utilizes both patch nodes and label nodes for multi-label classification. For patch nodes, they are aggregated via global average pooling, in which the contextual information of the image is processed as a whole for all the label. A linear layer is then applied to compute all the classification scores:

$$\hat{\boldsymbol{Y}}_x = Linear(AvgPool(\boldsymbol{X})), \tag{4}$$

where $AvgPool()$ and $Linear()$ denote the average pooling operation and the linear layer respectively. $\hat{\boldsymbol{Y}}_x \in \mathbb{R}^{1\times S}$ indicates the classification scores predicted via patch nodes. For label nodes, S linear layers are applied to the refined label nodes $\bar{\boldsymbol{h}}_i$ independently:

$$\hat{y}_i^h = Linear_i(\bar{\boldsymbol{h}}_i). \tag{5}$$

The classification score of each label $\hat{y}_i^h$ is predicted according to each label embedding $\bar{\boldsymbol{h}}_i$. They are concatenated as $\hat{\boldsymbol{Y}}_{\boldsymbol{h}} = [\hat{y}_1^h, \hat{y}_2^h, ..., \hat{y}_S^h] \in \mathbb{R}^{1\times S}$ and are processed with $\hat{\boldsymbol{Y}}_x$ together for the final scores:

$$\hat{\boldsymbol{Y}} = Sigmoid(\hat{\boldsymbol{Y}}_x + \hat{\boldsymbol{Y}}_h), \tag{6}$$

where $\hat{\boldsymbol{Y}} = [\hat{y}_1, \hat{y}_2, ..., \hat{y}_S]$ is the output classification probability of ML-ViG. Label smooth loss $\mathcal{L}_{smooth}$ (Szegedy et al., 2016) and asymmetric loss $\mathcal{L}_{asy}$(Ridnik et al., 2021) are applied to supervise the training process using the ground truth multi-labels $\boldsymbol{Y} = [y_1, y_2, ..., y_S]$:

$$y_s^{'} = \begin{cases} \frac{\varepsilon}{S} & y_s = 0 \\ 1 - \varepsilon + \frac{\varepsilon}{S} & y_s = 1 \end{cases}, \qquad \mathcal{L}_{smooth} = -\frac{1}{S}\sum_{s=1}^{S} y_s^{'} log(\hat{y_s}). \tag{7}$$

$$\mathcal{L}_{asy} = -\frac{i}{S}\sum_{i}^{S} \begin{cases} (1-\hat{y}_s)^{\gamma+} log(\hat{y}_s) & y_s = 1 \\ (\hat{y}_s)^{\gamma-} log(1-\hat{y}_s) & y_s = 0 \end{cases}. \tag{8}$$

where ε is the parameter that controls the smoothness, and $y_s^{'}$ is the smoothed target of the s^{th} label y_s. $\gamma+$ and $\gamma-$ are different focal values for positive samples and negative samples. In our work, we set $\varepsilon = 0.1$, $\gamma+ = 0$, and $\gamma- = 2$. The total loss is $\mathcal{L} = \mathcal{L}_{smooth} + \mathcal{L}_{asy}$.

4 EXPERIMENTS

We evaluate our model on several benchmark datasets, including MS-COCO (Lin et al., 2014), Pascal VOC (Everingham et al., 2015), and VG-500 (Krishna et al., 2017). We follow common practice (Wang et al., 2020; Lanchantin et al., 2021; Ye et al., 2020) to report the average of Overall Recall (OR), Overall Precision (OP), Overall F1-score (OF1), per-Class Recall (CR), per-Class Precision (CP), per-Class F1-score (CF1) and the mean Average Precision (mAP) as the evaluation metrics. Since OR/OP/CR/CP are easily affected by the classification threshold, mAP/CF1/OF1 are more important evaluation metrics. We set the threshold as 0.5 for recall, precision, and F1 score in our experiments.

4.1 IMPLEMENTATION DETAILS

ML-ViG uses ViG (Han et al., 2022) as the backbone. We follow (Han et al., 2022) to set $K_{vig} = 9$ in ViG blocks. For PLG blocks, we set $K_{plg} = 33$ when the input image size is 448×448 or 576×576, and $K_{plg} = 9$ when the input image size is 224×224. For the number of ViG blocks and MLG blocks at each stage, we follow ViG (Han et al., 2022) to set $L = [2, 2, 6, 2]$ and find the best setting for M to be $M = [1, 1, 1, 3]$. During training, class-balance sampling strategy (Kang et al., 2020) is adopted to reduce the adverse effects of class imbalance. We over-sample the categories whose frequency is lower than τ, where $\tau = 0.01$ in MS-COCO and VOC, and $\tau = 0.02$ in VG-500. We utilize AdamW (Loshchilov & Hutter, 2018) optimizer with the weight decay rate of 0.05 and the momentum of 0.9. The initial learning rate is set to 1×10^{-4} and a linear warm-up with the ratio of 1×10^{-3} is used. On Pascal VOC dataset, we reduce the learning rate by a factor of 0.1 on epoch 5. On MS-COCO (Lin et al., 2014) and VG-500 (Krishna et al., 2017) datasets, we reduce the learning rate by a factor of 0.1 on epoch 10. All experiments are implemented based on MMClassification (Contributors, 2020).

Table 1: **Comparisons** with state-of-the-art methods on MS-COCO dataset. All reported results are pre-trained on ImageNet-1K dataset and the input image size is 224×224 when pre-training. We report multiple evaluation metrics (in %), among which mAP, CF1, and OF1 are the primary metrics. Our ML-ViG outperforms existing models in terms of both accuracy and efficiency. The best results are marked as bold.

	All										Top3	
	Resolution	Param(M)	Flop(G)	mAP	CP	CR	CF1	OP	OR	OF1	CF1	OF1
ResNet-101(He et al., 2016)	224×224	44.5	7.8	78.3	80.2	66.7	72.8	**83.9**	70.8	76.8	69.7	73.6
SRN(Zhu et al., 2017)	224×224	76.8	9.0	77.1	81.6	65.4	71.2	82.7	69.9	75.8	67.4	72.9
Mltr(Cheng et al., 2022)	224×224	**33.0**	-	81.9	80.7	71.5	75.2	81.4	**76.3**	78.1	-	-
ML-ViG (Ours)	224×224	43.2	**6.9**	**82.1**	**82.4**	**72.3**	**77.0**	83.7	75.6	**79.4**	**73.5**	**76.0**
CADM(Chen et al., 2019b)	448×448	-	-	82.3	82.5	72.2	77.0	84.0	75.6	79.6	73.5	76.0
ML-GCN(Chen et al., 2019c)	448×448	44.9	31.5	83.0	85.1	72.0	78.0	85.8	75.4	80.3	74.6	76.7
KSSNet(Wang et al., 2020)	448×448	173.8	-	83.7	84.6	73.2	77.2	87.8	76.2	81.5	-	-
MS-CMA(You et al., 2020)	448×448	-	-	83.8	82.9	74.4	78.4	84.4	77.9	81.0	74.9	77.1
MCAR(Gao & Zhou, 2021)	448×448	-	-	83.8	85.0	72.1	78.0	**88.0**	73.9	80.3	75.1	76.7
TDRG(Zhao et al., 2021)	448×448	68.3	42.2	84.6	**86.0**	73.1	79.0	86.6	76.4	81.2	75.0	77.2
Q2L(ResNet101)(Liu et al., 2021)	448×448	193.6	51.4	84.9	84.8	74.5	79.3	86.6	76.9	81.5	73.3	75.4
ML-ViG (Ours)	448×448	**43.9**	**24.7**	**86.9**	84.8	**79.0**	**81.8**	86.0	**80.9**	**83.4**	**77.4**	**78.9**
SSGRL(Chen et al., 2019a)	576×576	92.3	68.5	83.8	91.9	62.5	72.7	93.8	64.1	76.2	76.8	79.7
MCAR(Gao & Zhou, 2021)	576×576	-	-	84.5	84.3	73.9	78.7	86.9	76.1	81.1	75.3	77.0
ADD-GCN(Ye et al., 2020)	576×576	48.2	52.7	85.2	84.7	75.9	80.1	84.9	79.4	82.0	75.8	77.9
C-Trans(Lanchantin et al., 2021)	576×576	120.4	84.2	85.1	86.3	74.3	79.9	87.7	76.5	81.7	76.0	77.6
TDRG(Zhao et al., 2021)	576×576	68.3	69.8	86.0	87.0	74.7	80.4	87.5	77.9	82.4	76.2	78.1
Q2L(ResNet101)(Liu et al., 2021)	576×576	193.6	80.8	86.5	85.8	76.7	81.0	87.0	78.9	82.8	76.5	78.3
ML-ViG (Ours)	576×576	**44.6**	**44.6**	**87.9**	85.5	**80.4**	**82.9**	86.6	**82.2**	**84.3**	**78.3**	**79.6**

4.2 COMPARISONS WITH THE STATE OF THE ARTS

4.2.1 MS-COCO

MS-COCO (Lin et al., 2014) is one of the most widely used benchmark datasets for multi-label classification. It contains 122,218 images which are composed of 82,081 training images and 40,137 validation images. In total, there are 80 kinds of object labels, with 2.9 labels on average for each image.

There are three commonly used input resolution settings for MS-COCO, *i.e.* 224×224, 448×448, and 576×576. For fair comparisons, we report the results of our ML-ViG on all these settings and

compare them with the state-of-the-art methods in Table 1. All the models adopt the pre-trained model on ImageNet1k (Deng et al., 2009) with 224×224 input size for initialization.

As shown in Table 1, our model obtains the best performance on all the input resolution settings in terms of mAP, CF1, and OF1. ML-GCN (Chen et al., 2019c), KSSNet (Wang et al., 2020), MS-CMA (You et al., 2020), ADD-GCN (Ye et al., 2020), SSGRL (Chen et al., 2019a), and TDRG (Zhao et al., 2021) are also GCN based models. We find that our proposed ML-ViG outperforms them by a large margin in all experiments, which demonstrates the superiority of our proposed unified graph representations for both visual features and label embeddings. For the input resolution of 448×448, our model improves upon ML-GCN (Chen et al., 2019c), KSSNet (Wang et al., 2020), MS-CMA (You et al., 2020), and TDRG (Zhao et al., 2021) by 3.9%, 3.2%, 3.1%, and 2.3% respectively. And for the higher resolution of 576×576, our model improves upon SSGRL (Chen et al., 2019a), ADD-GCN (Ye et al., 2020), TDRG (Zhao et al., 2021) by 4.1%, 2.7%, and 1.9% respectively. We also compare ML-ViG with recently proposed transformer-based methods (*e.g.* C-Trans (Lanchantin et al., 2021), and Q2L (Liu et al., 2021)). It is worth noting that our method achieves significantly lower computational complexity. For example, for high resolution (576×576) experiments, our model outperforms the previous state-of-the-art model (Q2L (Liu et al., 2021)) by 1.4 mAP, with only about a quarter of the parameter numbers (44.6M vs 193.6M) and half the computational complexity (44.6 GFlops vs 80.8 GFlops)

Table 2: **Comparisons** with state-of-the-art methods on VOC2007 dataset. All reported results are obtained by pre-training on the MS-COCO dataset. The best results are marked as bold.

	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
SSGRL(Chen et al., 2019a)	99.7	98.4	98.0	97.6	85.7	96.2	98.2	98.8	82.0	98.1	89.7	98.8	98.7	97.0	99.0	86.9	98.1	85.8	99.0	93.7	95.0
ASL(Ridnik et al., 2021)	**99.9**	98.4	98.9	98.7	86.8	98.2	98.7	98.5	83.1	98.3	89.5	98.8	99.2	98.6	99.3	89.5	99.4	86.8	**99.6**	95.2	95.8
ADD-GCN(Ye et al., 2020)	99.8	99.0	98.4	99.0	86.7	98.1	98.5	98.3	85.8	98.3	88.9	98.8	99.0	97.4	99.2	88.3	98.7	90.7	99.5	97.0	96.0
Q2LLiu et al. (2021)	**99.9**	98.9	99.0	98.4	87.7	98.6	**98.8**	99.1	84.5	98.3	89.2	99.2	99.2	**99.2**	99.3	90.2	98.8	88.3	99.5	95.5	96.1
ML-ViG (Ours)	**99.9**	**99.2**	**99.5**	**99.6**	**88.6**	**98.8**	98.7	**99.4**	**88.9**	**99.0**	**92.4**	**99.5**	**99.5**	99.1	**99.5**	**90.5**	**99.3**	**91.2**	99.4	**97.3**	**97.0**

Table 3: **Comparisons** with state-of-the-art methods on VG-500 dataset. All reported results are pre-trained on ImageNet-1K dataset and the input image size is 224×224 when pre-training. The best results are marked as bold.

	All							Top3					
	mAP	CP	CR	CF1	OP	OR	OF1	CP	CR	CF1	OP	OR	OF1
ResNet101(He et al., 2016)	30.9	39.1	25.6	31	61.4	35.9	45.4	39.2	11.7	18	75.1	16.3	26.8
ML-GCN(Chen et al., 2019c)	32.6	42.8	20.2	27.5	**66.9**	31.5	42.8	39.4	10.6	16.8	77.1	16.4	27.1
SSGRL(Chen et al., 2019a)	36.6	-	-	-	-	-	-	-	-	-	-	-	-
KGGR(Chen et al., 2020)	37.4	47.4	24.7	32.5	**66.9**	36.5	47.2	48.7	12.1	19.4	78.6	17.1	28.1
C-Tran(Lanchantin et al., 2021)	38.4	**49.8**	27.2	35.2	**66.9**	39.2	49.5	**51.1**	12.5	20.1	80.2	17.5	28.7
ML-ViG (ours)	**39.0**	45.8	**35.3**	**39.9**	60.4	**48.6**	**53.9**	49.4	**12.9**	**20.4**	**80.3**	**17.7**	**28.9**

4.2.2 PASCAL VOC

Pascal VOC 2007 (Everingham et al., 2015) is also a common dataset in the MLIR task, which contains 20 label categories. VOC 2007 has 9,963 images and is divided into a train-val dataset (5,011 images) and a test dataset (4,952 images). We follow the common settings (Ridnik et al., 2021; Ye et al., 2020; Chen et al., 2019a; Liu et al., 2021) to train the model on the train-val dataset and evaluate on the test dataset.

For fair comparisons, we follow previous works (Chen et al., 2019a; Ye et al., 2020) to pre-train the model on the MS-COCO dataset and report the results at the input resolution of 576×576. As shown in Table 2, our model has 0.9% improvement on mAP compared with the previous state-of-the-art methods. Especially, we report the AP of each category and show that our model achieves the best performance for 16 of all the 20 categories and very competitive results for the remaining 4 categories. This demonstrates that our model brings general and consistent improvement.

4.2.3 VG-500

Visual Genome (Krishna et al., 2017) contains 108,249 images. The train set has 98,249 images and the test set has 10,000 images. It has 80,138 categories with densely annotated objects, attributes, and relationships. About 500 categories are frequently used and the remaining categories are relatively rare (Lanchantin et al., 2021). The dataset with 500 frequent categories is called VG-500

dataset. Compared with MS-COCO and Pascal VOC dataset, VG-500 is much more challenging because of a more complex and larger output label space and a more severe class imbalance. The label space not only contains specific object categories but also includes abstract attributes (*e.g.* 'yellow') and relationship categories (*e.g.* 'on').

Our experimental results are demonstrated in Table 3. For fair comparisons, all results are reported at the input resolution of 576×576 and the models are pre-trained on ImageNet1k (Deng et al., 2009) with 224×224 input resolution. We show that our method establishes a new state of the art on the VG-500 dataset in terms of mAP, CF1, and OF1. Especially, compare with the previous state-of-the-art model C-Tran (Lanchantin et al., 2021), we show that our model obtains an 8.1% gain on ALL-CR and 9.4% gain on ALL-OR, which in turn brings 4.7% improvement on ALL-CF1 and 4.4% improvement on ALL-OF1. This indicates that ML-ViG can better model label occurence and improve the recall of those categories with a few occurrences.

4.3 Ablation Study

Effect of each component in MLG. Our proposed MLG block includes a PLG module to extract features from the local region, and an LLG module to further capture the relationship between labels. We conduct ablation studies to demonstrate the importance of these two modules separately in Table 4. The experiments are conducted on the MS-COCO dataset with the setting of 576×576 input resolution. We show that adding PLG brings large gains (87.3% vs 86.2% mAP), which demonstrates the necessity and effectiveness of extracting features from the interest regions. And we also show that adding LLG on the basis of PLG further brings improvements (87.9% vs 87.3% mAP). This validates the importance of modeling label occurrence. Both of these two modules are indispensable for the success of ML-ViG.

Table 4: **Ablation studies** on the MS-COCO dataset. The experiments are conducted with the setting of 576×576 input resolution. Both PLG and LLG modules can improve the model performance.

PLG	LLG	mAP	CF1	OF1
		86.2	80.7	83.8
✓		87.3	82.1	83.8
✓	✓	**87.9**	**82.9**	**84.3**

Effect of the number of nearest neighbors (K_{plg}). In our proposed PLG block, K-nearest neighbors (KNN) is used for graph construction, connecting each label node with K_{plg} nearest neighbor patch nodes. The number of neighbors K_{plg} is a hyperparameter controlling the range of the area where region features will be extracted and aggregated. Large K_{plg} will lead to feature oversmoothing and involve interference of the invalid background, while small K_{plg} will affect feature extraction and message passing. We explored the effect of K_{plg} from 12 to 48 on the MS-COCO dataset and show the results in Figure 3. The experiments are conducted with the setting of 576×576 image size. We find that the model performance is not sensitive to the setting of K_{plg}. The best performance is achieved when $K_{plg} = 33$ and the computation costs are almost unaffected by K_{plg}. Therefore, we select the best setting $K_{plg} = 33$ for all the experiments.

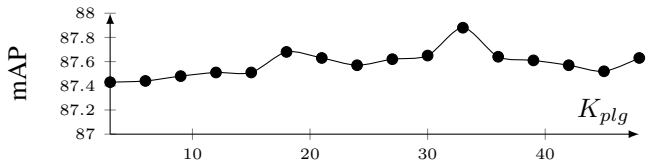
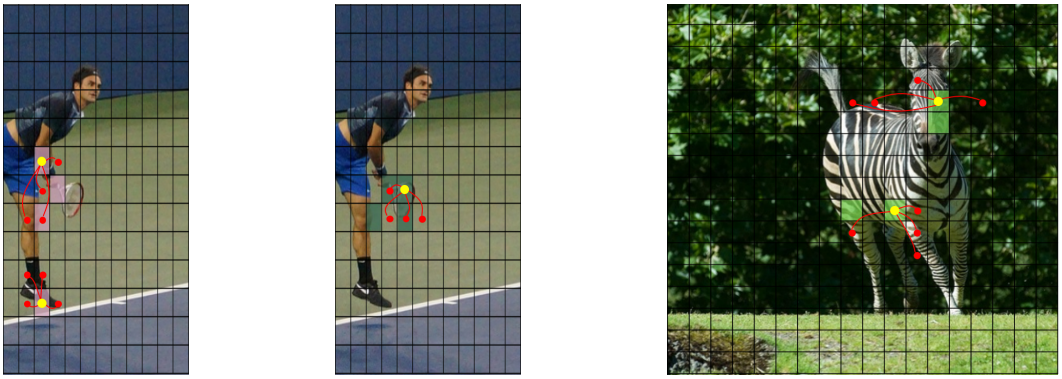


Figure 3: **Effect of different K_{plg} of PLG.** Experiments are conducted on MS-COCO dataset with the setting of 576×576 input resolution. Our model is not sensitive to the changes of K_{plg}.



(a) person **(b)** tennis racket **(c)** zebra

Figure 4: Visualization of the learned connections between patch nodes and label nodes (MLG), and also the connections between patch nodes (ViG). (a), (b), and (c) correspond to "person", "tennis racket", and "zebra", respectively. The colored blocks are the label nodes' nearest patch nodes. For clarity, we visualize the top-4 nearest neighbor patch nodes for each label node. For patch-level graph visualization, the yellow dot is the center node and the red dots connecting to it are the neighbor nodes. The red lines represent the connections between patch nodes.

4.4 Visualization and Analysis

In this section, we visualize the learned graph structure in both ViG and MLG to better understand how our ML-ViG works. Specifically, we visualize the connections between patch nodes and label nodes in the last PLG block, and the connections between patch nodes in the last ViG block.

As shown in Figure 4, (a), (b), (c) are person, tennis, racket, respectively. Especially, (a) and (b) correspond to two different labels existing in the same image. Images are divided into a set of patches and the colored blocks are the label nodes' nearest patch nodes. For clarity, we visualize the top-4 nearest neighbor patch nodes for each label node. In Figure 4a, the pink patches are the neighbors of the label "people", and we see that these regions locate on human body parts, including arms, legs, and feet. And in Figure 4b and Figure 4c, we can observe that the label "tennis racket" and the label "zebra" accurately find the related regions corresponding to the object label. This validates that the representations of label embeddings and visual features are well aligned in our proposed PLG block.

We further visualize the learned connections between patch nodes in ViG block. The yellow dot is the center node and the red dots connecting to it are the neighbor nodes. The red lines represent the connections between patch nodes. For clarity, we only visualize a small number of center nodes. In Figure 4a, we can see that the patch on the arm is linked to other parts (*e.g.* the hand and the knee) of the person. Figure 4b and Figure 4c also show that the ViG block has the flexibility to aggregate features from both surrounding areas, and other areas at a distance since they are linked by semantic content instead of by position distance. This indicates that our model can learn better visual representations based on global and wider range relations, which avoids inductive biases of CNN and helps to better categorization.

5 Conclusion

In this paper, we propose ML-ViG, a novel and flexible fully graph convolutional model for the task of MLIR. We pioneer to study the unified graph representations for both visual features and label embeddings. And three kinds of relationships are effectively captured in the graph structure, including region space relations, label semantic relations, and cross-relations between labels and regions. Comprehensive experiments on public benchmark datasets, *i.e.* MS-COCO, VOC2007, and VG-500, demonstrate the effectiveness of our proposed method. We hope our work will draw the community's attention to unified graph representations for general vision tasks.

REFERENCES

Tianshui Chen, Muxin Xu, Xiaolu Hui, Hefeng Wu, and Liang Lin. Learning semantic-specific graph representation for multi-label image recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 522–531, 2019a.

Tianshui Chen, Liang Lin, Xiaolu Hui, Riquan Chen, and Hefeng Wu. Knowledge-guided multi-label few-shot learning for general image recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

Zhao-Min Chen, Xiu-Shen Wei, Xin Jin, and Yanwen Guo. Multi-label image recognition with joint class-aware map disentangling and label correlation embedding. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 622–627. IEEE, 2019b.

Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, and Yanwen Guo. Multi-label image recognition with graph convolutional networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5177–5186, 2019c.

Xing Cheng, Hezheng Lin, Xiangyu Wu, Dong Shen, Fan Yang, Honglin Liu, and Nian Shi. Mltr: Multi-label classification with transformer. In *2022 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6. IEEE, 2022.

MMClassification Contributors. Openmmlab's image classification toolbox and benchmark. `https://github.com/open-mmlab/mmclassification`, 2020.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Mark Everingham, SM Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1):98–136, 2015.

Bin-Bin Gao and Hong-Yu Zhou. Learning to discover multi-class attentional regions for multi-label image recognition. *IEEE Transactions on Image Processing*, 30:5920–5932, 2021.

Kai Han, Yunhe Wang, Jianyuan Guo, Yehui Tang, and Enhua Wu. Vision gnn: An image is worth graph of nodes. In *NeurIPS*, 2022.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Ashesh Jain, Amir R Zamir, Silvio Savarese, and Ashutosh Saxena. Structural-rnn: Deep learning on spatio-temporal graphs. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pp. 5308–5317, 2016.

Jian Jia, Xiaotang Chen, and Kaiqi Huang. Spatial and semantic consistency regularizations for pedestrian attribute recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 962–971, 2021.

Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *Eighth International Conference on Learning Representations (ICLR)*, 2020.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017.

Jack Lanchantin, Tianlu Wang, Vicente Ordonez, and Yanjun Qi. General multi-label image classification with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16478–16488, 2021.

Guohao Li, Matthias Muller, Ali Thabet, and Bernard Ghanem. Deepgcns: Can gcns go as deep as cnns? In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9267–9276, 2019.

Yining Li, Chen Huang, Chen Change Loy, and Xiaoou Tang. Human attribute recognition by deep hierarchical contexts. In *European conference on computer vision*, pp. 684–700. Springer, 2016.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Shilong Liu, Lei Zhang, Xiao Yang, Hang Su, and Jun Zhu. Query2label: A simple transformer way to multi-label classification. *arXiv preprint arXiv:2107.10834*, 2021.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. 2018.

Tal Ridnik, Emanuel Ben-Baruch, Nadav Zamir, Asaf Noy, Itamar Friedman, Matan Protter, and Lihi Zelnik-Manor. Asymmetric loss for multi-label classification. pp. 82–91, 2021.

Jing Shao, Kai Kang, Chen Change Loy, and Xiaogang Wang. Deeply learned attributes for crowded scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4657–4666, 2015.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Ya Wang, Dongliang He, Fu Li, Xiang Long, Zhichao Zhou, Jinwen Ma, and Shilei Wen. Multi-label classification with label graph superimposing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 12265–12272, 2020.

Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

Jin Ye, Junjun He, Xiaojiang Peng, Wenhao Wu, and Yu Qiao. Attention-driven dynamic graph convolutional network for multi-label image recognition. In *European conference on computer vision*, pp. 649–665. Springer, 2020.

Renchun You, Zhiyao Guo, Lei Cui, Xiang Long, Yingze Bao, and Shilei Wen. Cross-modality attention with semantic graph embedding for multi-label classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 12709–12716, 2020.

Jiawei Zhao, Ke Yan, Yifan Zhao, Xiaowei Guo, Feiyue Huang, and Jia Li. Transformer-based dual relation graph for multi-label image recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 163–172, 2021.

Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020.

Feng Zhu, Hongsheng Li, Wanli Ouyang, Nenghai Yu, and Xiaogang Wang. Learning spatial regularization with image-level supervisions for multi-label image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5513–5522, 2017.

Guangming Zhu, Liang Zhang, Youliang Jiang, Yixuan Dang, Haoran Hou, Peiyi Shen, Mingtao Feng, Xia Zhao, Qiguang Miao, Syed Afaq Ali Shah, et al. Scene graph generation: A comprehensive survey. *arXiv preprint arXiv:2201.00443*, 2022.