Neural Computing and Applications (2022) 34:10051–10063
https://doi.org/10.1007/s00521-022-06990-3

ORIGINAL ARTICLE

STMG: Swin transformer for multi-label image recognition with graph convolution network

Yangtao Wang[1] · Yanzhao Xie[2] · Lisheng Fan[1] · Guangxing Hu[2]

Received: 4 August 2021 / Accepted: 30 January 2022 / Published online: 21 February 2022

Abstract
Vision Transformer (ViT) has achieved promising single-label image classification results compared to conventional neural network-based models. Nevertheless, few ViT related studies have explored the label dependencies in the multi-label image recognition field. To this end, we propose STMG that combines transformer and graph convolution network (GCN) to extract the image features and learn the label dependencies for multi-label image recognition. STMG consists of an image representation learning module and a label co-occurrence embedding module. Firstly, in the image representation learning module, to avoid computing the similarity between each two patches, we adopt Swin transformer instead of ViT to generate the image feature for each input image. Secondly, in the label co-occurrence embedding module, we design a two-layer GCN to adaptively capture the label dependencies to output the label co-occurrence embeddings. At last, STMG fuses the image feature and label co-occurrence embeddings to produce the image classification results with the commonly-used multi-label classification loss function and a L2-norm loss function. We conduct extensive experiments on two multi-label image datasets including MS-COCO and FLICKR25K. Experimental results demonstrate STMG can achieve better performance including the convergence efficiency and classification results compared to the state-of-the-art multi-label image recognition methods. Our code is open-sourced and publicly available on GitHub: https://github.com/lzHZWZ/STMG.

Keywords Swin transformer · Graph convolution network · Multi-label image recognition

1 Introduction

In the past few years, transformers have achieved great success in the natural language processing (NLP) [8, 29] domain by learning the intrinsic relationships between elements of a sequence with the self-attention mechanism. Inspired by the effectiveness of this transformer architecture, researchers begin to explore how to apply transformers to computer vision or multi-modal learning tasks and have proposed a series of excellent transformer-based models such as Vision Transformer (ViT) [10] and its variant DeiT [28] for image classification, Action Transformer [12] for video understanding, SETR [36] for semantic segmentation, DERT [2] for object detection, and LXMERT [27] for visual question answering.

Compared to the conventional convolution neural networks (CNNs) like commonly-used ResNet [14] and InceptionNet [26], ViT has achieved promising performance on single-label image classification tasks that predict and allocate one label for each image. Unlike CNN-based models that cannot extract the global image feature owing to the local convolution operation, ViT [10] first splits each image into a sequence of embedded image patches and then adopts the standard transformer [29] encoder to compute the similarity between each two patches to generate the global image representation. However, on the one hand, ViT requires large-scale datasets (like

✉ Yanzhao Xie
yzxie@hust.edu.cn

Yangtao Wang
ytaowang@gzhu.edu.cn

Lisheng Fan
lsfan@gzhu.edu.cn

Guangxing Hu
Garson_hu@hust.edu.cn

[1] School of Computer Science and Cyber Engineering, Guangzhou University, Guangzhou, China

[2] Huazhong University of Science and Technology, Wuhan, China

ImageNet21K [7] and JFT300M [25]) for training to obtain better image classification results. On the other hand, owing to much higher resolution of pixels in images compared with words in text, the computational complexity of the self-attention mechanism in ViT becomes quadratic to image size. To address this issue, Swin transformer [21] first computes the similarity between small-sized image patches within each window and then designs shifted windows to compute the similarity between these patches in the neighboring windows, which reduces the computational complexity to linear level. Although these approaches can effectively learn the image feature of single-label images, they cannot be directly applied to multi-label image recognition tasks that aim to predict a set of labels corresponding to those objects that co-occur in an image. Besides, the existing vision transformer-based models rarely explore the label dependencies between different objects in multi-label images. As shown in Fig. 1, a ”dolphin” will always appear in the ”ocean”, and a ”book” is more likely to be put on a ”desk”, but it is impossible that a ”dolphin” will co-occur with a ”desk” in the real world. This phenomenon reflects the objective rule that semantically related objects will be more likely to appear in the same image while those objects without direct relationships will hardly co-occur in an image. To model the label dependencies, both ML-GCN [5] and A-GCN [19] utilize graph convolution network (GCN) to explore the correlations between different label embeddings over the data distribution, which obtain good classification results but bring low model convergence efficiency caused by the cross-modal fusion from image features and label word vectors. Consequently, F-GCN [33] has to add a cross-modal fusion component to speed up the model convergence, but it increases the complexity of the model. To be sure, these methods effectively explore the label dependencies and give us great inspiration.

In order to learn the global image feature as well as capture the label dependencies between objects, in this paper, we integrate GCN into Swin transformer and propose STMG for multi-label image recognition. STMG mainly consists of two key modules: an image representation learning module and a label co-occurrence embedding module. Firstly, in the image representation learning module, to avoid computing the similarity between each two patches, we improve Swin transformer to adapt to our task to generate the image feature for each multi-label image. Secondly, in the label co-occurrence embedding module, we design a two-layer GCN to adaptively capture the label dependencies to output the label co-occurrence embeddings. At last, STMG fuses the image feature and label co-occurrence embeddings to produce the image classification results with the commonly-used multi-label classification loss function and a L2-norm loss function. We conduct extensive experiments on two multi-label image datasets including MS-COCO [20] and FLICKR25K [16]. Experimental results demonstrate STMG can achieve better performance including the convergence efficiency and classification results compared with the state-of-the-art multi-label image recognition methods.

The main contributions of this paper can be summarized as follows.

- We propose STMG and this is the first time that we improve Swin transformer to adapt to our multi-label image recognition task, which can effectively address the cross-modal embeddings fusion problem.
- We first design a two-layer GCN to smoothly learn the relationships between label embeddings and then integrate it into Swin transformer to both learn the global

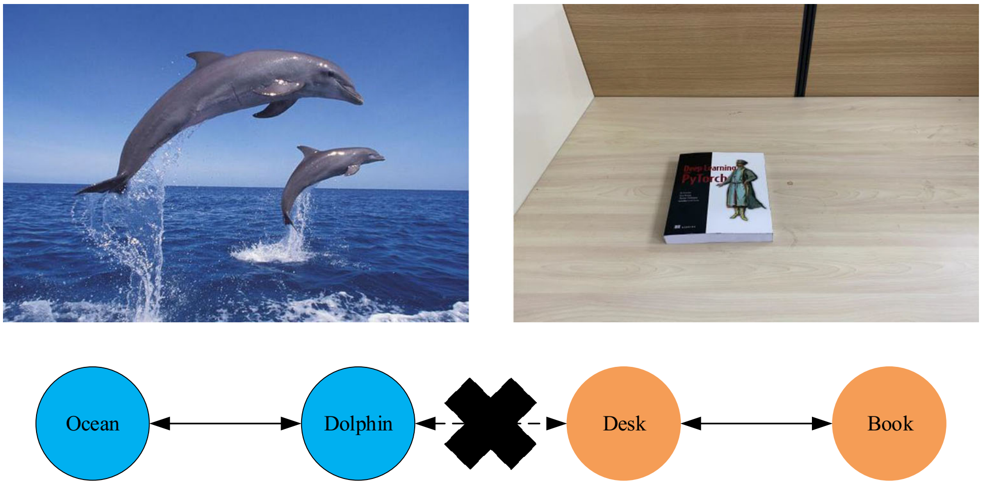


Fig. 1 Label dependencies between different objects. Related objects are more likely to appear in the same image. For example, a “dolphin” will always appear in the “ocean”, and so is the combination of “book” and “desk”, but we never see a “dolphin” co-occurs with a “desk”

image feature and capture the label dependencies between objects.
- Extensive experiments on MS-COCO and FLICKR25K demonstrate STMG can achieve better performance including the convergence efficiency and classification results compared with the state-of-the-art multi-label image recognition methods.

The remainder of this paper is organized as follows. Section 2 talks about related works. We respectively elaborate our proposed STMG in Sect. 3 and analyze the experimental results in Sect. 4. At last, we conclude this paper in Sect. 5.

2 Related works

In this section, we will talk about the existing works related to our research including the mainstream multi-label image recognition methods, vision transformer models as well as GCN based studies.

2.1 Multi-label image recognition

Early studies treat multi-label image recognition task as multiple individual single-label classification tasks and train a classifier for each label. However, the classification performance of these binary approaches has been essentially limited owing to that they fail to take the relationships between objects into consideration. As a result, many researchers begin to explore these label relationships in various ways. For instance, Wang et al. [30] combine CNN with recurrent neural network to embed semantic labels into vectors, which models the label dependencies in a sequential fashion. Recently, some works begin to associate the image region with its corresponding label via attention mechanisms. Zhu et al. [37] propose SRN that uses image-level supervisions to learn an attention map for each given label. Combined with a long short-term memory unit, Wang et al. [34] efficiently perceive attentional regions via a spatial transformer to capture the label dependencies. Guo et al. apply the class activation mapping mechanism and propose VACIT [13] to learn the attention map distances between the original image and its transformed one. These methods effectively explore the label correlation within an image but ignore the label dependencies between different objects over the data distribution.

Furthermore, others utilize GCN to capture the global label dependencies for multi-label image recognition. For example, Chen et al. propose ML-GCN [5] to utilize GCN to learn the label co-occurrence relationship and achieve good results on multi-label images. Similarly, A-GCN [19] designs an adaptive graph to explore the label dependencies according to the label embeddings, and F-GCN [33] introduces a cross-modal component to speed up the model convergence and also obtains comparable performance. EmotionGCN [15] is a similar work that adopts CNN for image feature and GCN for emotion distribution learning. In addition, SS-GRL [4] generates class-specific image representations via semantic graph networks learning to pay more attention to the semantic regions of images. These methods aim to capture the co-occurrence probability of training samples, which may reduce the model generalization ability, especially for those rare co-occurrence objects. To overcome this, ADD-GCN [35] designs an attention driven dynamic GCN, which can dynamically generate a specific graph for each image to model the relationships between content perception categories generated by semantic attention module. All of the above methods adopt a CNN-based model as the backbone to extract the image feature, so vision transformer has not been explored for multi-label image recognition.

2.2 Vision transformer

In the NLP [8, 29] domain, transformers have made a great breakthrough via an encoder-decoder work-flow with its effective self-attention mechanism in recent years and inspired researchers to apply these transformer-based models to the computer vision community. ViT [10] is the first work that gets rid of convolution operations and applies a standard transformer encoder for image classification by splitting each image into a sequence of embedded image patches, but it requires large-scale datasets (i.e., ImageNet21K [7] and JFT300M [25]) for pre-training to help achieve the similar performance as CNNs. To this end, DeiT [28] introduces a novel native distillation approach into transformers, which is able to train its model on mid-sized datasets with relatively shorter training time compared to ViT. Despite good performance on image classification, owing to much higher resolution of pixels in images compared to words in text, the computational complexity of the self-attention mechanism in these ViT-based models becomes quadratic to image size as they have to compute the similarity between each two image patches. To address this issue, Swin transformer [21] first computes the similarity between small-sized image patches within each window and then designs shifted windows to compute the similarity between these patches in the neighboring windows, which reduces the computational complexity to linear level and obtains comparable image classification results against top-performing CNN models.

In addition to image classification, transformers have also been applied to various compute vision tasks. For example, Carion et al. combines CNN with transformer and propose DERT [2] for object detection, which aims to predict the set of object bounding boxes given a set of image features. Similarly, Zheng et al. treat semantic segmentation as a sequence-to-sequence prediction task and propose SETR [36], which first adopts a pure transformer encoder to embed a set of image patches and then conducts progressive upsampling or multi-level feature aggregation in the decoder layers to generate its semantic segmentation model. Neimark et al. [23] propose Video Transformer Network which first splits each video into a set of frames and then directly utilizes ViT to learn the representations of these frames for the final video classification. These vision transformer-based works are shining in computer vision community and will inspire others to perform more forward-looking research in this field.

2.3 Graph convolution network

Graph convolution network (GCN) [17] works as an effective feature extractor on graph based structure, which has been widely applied in node classification, graph classification, link prediction, etc. The input of GCN contains the nodes' features and the correlation matrix between these nodes, and GCN will update a node's feature by means of aggregating the features of its neighbors and itself into this node. The graph propagation process can be formulated as follows:

$$H^{l+1} = a^l(\hat{A}H^lW^l), \tag{1}$$

where H^l, W^l, a^l respectively, denote the input features, weights, non-linear activation function of the l-th graph convolution layer and $\hat{A}$ denotes the normalized version of correlation matrix A:

$$\hat{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}, \tag{2}$$

where $\tilde{A} = A + I$, I is the identity matrix and $\tilde{D}$ is a diagonal matrix with $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$.

In recent years, GCN has attracted wide attention in computer vision and NLP tasks. For example, Wang et al. [31] adopts GCN to generate the semantic embeddings and learn knowledge graphs for zero-shot image classification. Defferard et al. [6] apply GCN to N-grams for text categorization. More recently, researchers begin to utilize GCN to explore the label dependencies on multi-label images. Chen et al. [5] propose ML-GCN to learn the label co-occurrence embeddings according to the label statistical information. Li et al. [19] propose A-GCN that designs an adaptive label graph to capture the label correlations. To overcome the low model efficiency of ML-GCN and A-GCN, Wang et al. [33] introduce a cross-modal fusion component and propose F-GCN to greatly speed up the model convergence for multi-label image recognition. Besides, ADD-GCN [35] adopts a dynamic GCN to learn a specific graph to obtain each image representation and SS-GRL [4] generates class-specific image representations via semantic graph networks learning. Similarly, EmotionGCN [15] adopts GCN to effectively capture the emotion distribution and NRDH [32] treats each image as a node to learn the similarity between nodes with GCN for image retrieval.

The above works give us great inspirations, especially ML-GCN, A-GCN and F-GCN. As we mentioned, both ML-GCN and A-GCN adopt CNN to extract image features and GCN to generate label co-occurrence embeddings, but they suffer from low convergence efficiency due to cross-modal fusion process. To alleviate this problem, F-GCN has to add another cross-modal component to speed up the model convergence but brings higher model complexity. In order to overcome the cross-modal embeddings fusion challenge, we observe Swin transformer can extract features in an NLP way, so we expect to address this problem by fusing two kinds of embeddings that are both generated from NLP techniques. Based on this, we integrate GCN into vision transformer to extract the image features and learn the label dependencies for multi-label image recognition.

3 Proposed methodology

In this section, we integrate GCN into Swin transformer and propose STMG for multi-label image recognition. STMG mainly consists of two key modules: an image representation learning module and a label co-occurrence embedding module. We present the overall framework of STMG in Fig. 2. In the following, we will introduce the work-flow of our approach in detail.

3.1 Image representation learning module

As shown in the blue frame of Fig. 2, in this part, we aim to adopt Swin transformer to learn the image representation for each image.

Given each RGB image $I \in R^{H \times W \times 3}$, we will divide this image into a set of non-overlapping image patches, where H and W respectively denote the height and width of each image. Similar to ViT [10], each patch will be first flattened and then transformed into a linear embedding (treated as a token). The original ViT computes the similarity between every two tokens, which leads to quadratic complexity with respect to the number of tokens. To overcome this shortcoming, we adopt the self-attention computation method that has been proved to be linear to number of tokens in Swin transformer [21] to learn the image representation via dividing an image into non-overlapping windows. Swin

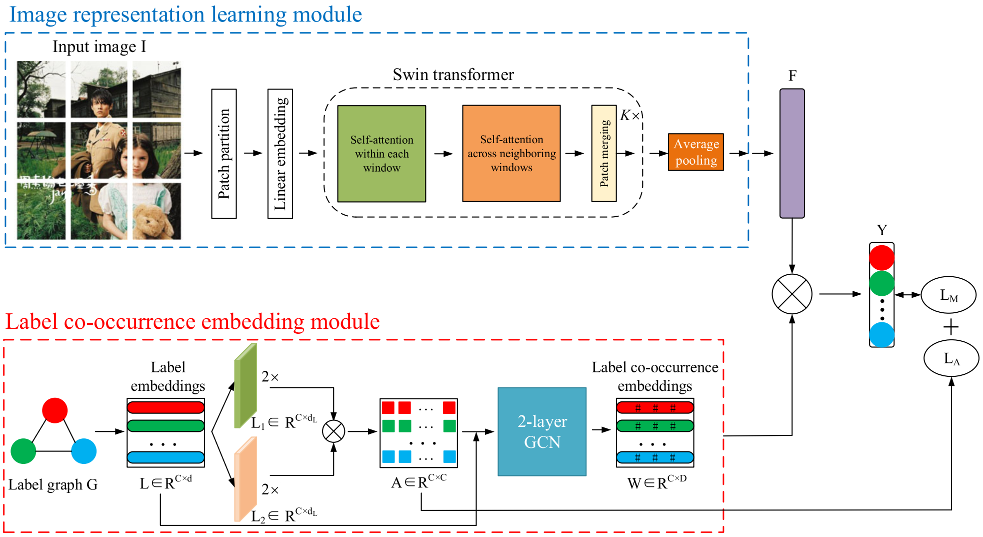


Fig. 2 The overall framework of STMG consists of an image representation learning module and a label co-occurrence embedding learning module, where the former adopts Swin transformer to learn each image feature and the latter utilizes GCN to generate the label co-occurrence embeddings. These two embeddings will be fused to obtain the prediction labels for each image and our model will be updated in an end-to-end manner

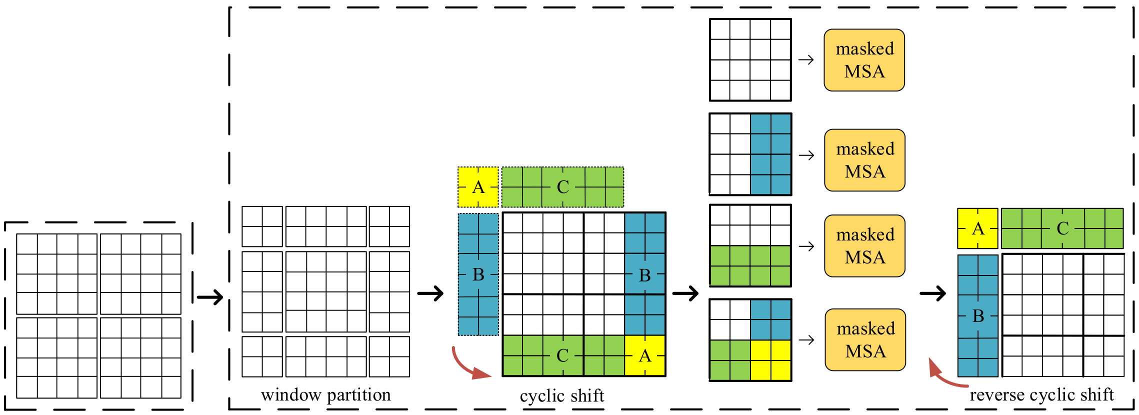
transformer contains K blocks, each of which completes the self-attention computation within each window as well as the self-attention computation across non-overlapping windows. We present an illustration of this self-attention mechanism in Fig. 3 and introduce the details in the following.

3.1.1 Self-attention within each window

As shown in Fig. 3a, we first divide an image into multiple non-overlapping windows, each of which contains $M \times M$ (set as 7 as default) patches. In each local window, we conduct the standard multi-head self-attention (MSA) mechanism as follows:

$$\begin{aligned} z_w^k &= W\text{-MSA}(LN(z^{k-1})) + z^{k-1}, \\ z_w^k &= \text{MLP}(LN(z_w^k)), \end{aligned} \tag{3}$$

where $k \in \{1, 2, \cdots, K\}$, z^{k-1} denotes the input features of the l-th block, W-MSA denotes MSA operation in each local window, LN denotes the Layer Normalization [1] operation, z_w^k denotes the output features of the k-th block after self-attention computation within each window. In this way, we can effectively capture the similarity between patches within each window. However, there is a lack of connections across non-overlapping windows, which results in that patches in different windows have no interaction with each other and will greatly limit the model performance. To this end, referring to Swin transformer, we add a window partition scheme to conduct the self-



(a) Self-attention within each window

(b) Self-attention across non-overlapping windows

Fig. 3 Illustration of self-attention computation in Swin transformer

attention computation across non-overlapping windows to complete the cross-window interaction process.

3.1.2 Self-attention across non-overlapping windows

In order to compute the similarity of patches among different windows, we adopt a shifted window partition scheme to cross the boundaries of the previous windows and result in new windows. Take Fig. 3 as an example, we re-divide 4 windows into 9 new “small” windows. If we directly perform MSA within these 9 windows, the computation overhead will be 2.25 (i.e., 9/4) times greater than the previous one. To reduce the computation overhead, we first conduct cyclic shift to organize these 9 windows into 4 “big” windows, as shown in Fig. 4. Then, we conduct the same *MSA* operation within these 4 “big” windows, but we will add a mask operation to avoid the unnecessary cross-window computation and only retain the self-attention computation within the original 9 individual “small” windows. Take Fig. 4 as an example, 4 masked *MSA* operations within the 4 “big” windows are equal to 9 MSA operations within the 9 “small” windows, but the computation overhead has been greatly reduced. This masked process can be formulated as follows:

$$\begin{aligned} \hat{z}_a^k &= M\text{-MSA}\big(LN(z_w^k)\big) + z_w^k, \\ z_a^k &= \text{MLP}\big(LN(\hat{z}_a^k)\big), \end{aligned} \qquad (4)$$

where $k \in \{1, 2, \cdots, K\}$, z_w^k (i.e., the output features after the previous self-attention computation within each window in Eq. 3) denotes the input features before the self-attention computation across non-overlapping windows in the k-th block, M-MSA denotes masked MSA operation, z_a^k denotes the output features of the k-th block after self-attention computation across non-overlapping windows. In this way, we can effectively capture the similarity between patches across different windows with the linear computation complexity. After completing this self-attention computation, we conduct the reverse cycle shift (shown in Fig. 3b) to restore these “big” windows to the original “small” windows, which will be sent to the next block to complete these self-attention computation processes within and across windows. Note that we include a relative position bias in each self-attention computation and add a patch merging layer in each block after self-attention computation to reduce the number of tokens by concatenating the neighboring patches. We will not repeat these here, and please see reference [21] for more details of this relative position bias as well as down-sampling process.

After completing all the K blocks self-attention computation, we conduct average pooling on each small window (containing 7×7 patches as default in the experiments) of the last block to output the D-dimensional feature vector F for image I. This feature will be fused with the label co-occurrence embeddings (see Sect. 3.2) to generate the classification results of image I. This part

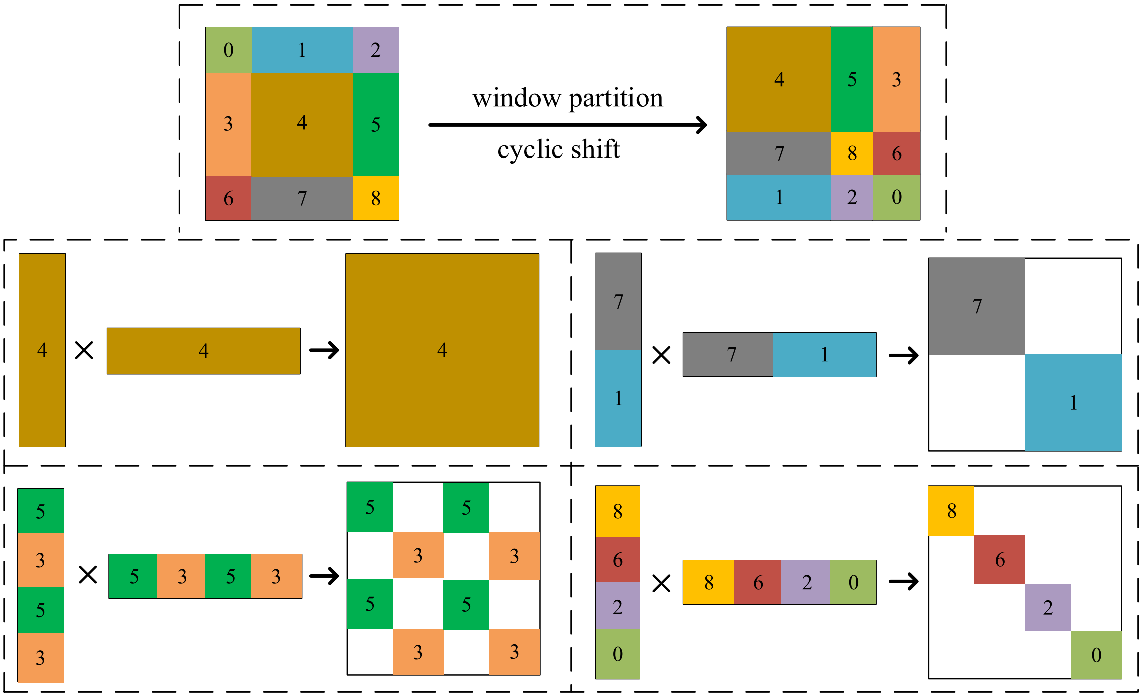


Fig. 4 Illustration of masked MSA

effectively extracts the global image feature with Swin transformer, and we will introduce how to obtain the label co-occurrence embeddings below.

3.2 Label co-occurrence embedding module

In this part, we design GCN with two layers to complete the label graph learning process to obtain the label co-occurrence embeddings. This process aims to capture the label dependencies between different objects.

In the red frame of Fig. 2, we treat each object as a node of our label graph G consisting of C nodes, where C denotes the number of object categories in a dataset. Different from ML-GCN [5] and A-GCN [19], we adopt the Bert [9] model to map each object into a d-dimensional word vector. As a result, we will generate a $C \times d$ label embeddings matrix L for all the C object labels. After obtaining the node features, we begin to construct the correlation matrix based on these nodes' features. ML-GCN [5] has to manually count the occurrence times of each object as well as the conditional probability between different objects to model the label dependencies, which becomes inflexible and may bring sub-optimal results for multi-label image recognition. To get rid of this hand-crafted correlation graph, we improve A-GCN [19] to automatically learn the label correlations and obtain the correlation matrix in an end-to-end manner.

Based on the label embeddings matrix L, we first use two 1×1 convolution layers to respectively generate two intermediate $C \times d_L$ matrices L_1 and L_2, then adopt a dot product operation to generate the $C \times C$ correlation matrix A. This process can be formulated as:

$$\begin{aligned} L_1 &= 2T(f_\alpha(L;\theta_\alpha)), \\ L_2 &= 2T(f_\beta(L;\theta_\beta)), \\ A &= \frac{1}{C} L_1 \otimes L_2^T, \end{aligned} \tag{5}$$

where f_α and f_β respectively, denote the convolution operations of these two branches, θ_α and θ_β respectively, denote the network parameters of these two branches, $2T(\cdot)$ denotes this operation will be conducted twice to fully learn the relationships between label embeddings, and $\otimes$ denotes the dot product operation. Following the commonly-used normalization trick [17], we normalize A to $\hat{A}$ as follows:

$$\hat{A} = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}, \tag{6}$$

where $\tilde{A} = A + I_C$ and $\tilde{D}$ is a diagonal matrix with $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$ for $i, j \in [1, C]$, I_C is a $C \times C$ identity matrix.

After obtaining both the label embeddings matrix L and normalized label correlation matrix $\hat{A}$, we design GCN with two layers to update the nodes' features as follows:

$$L_{l+1} = f_l(\hat{A} L_l U_l), \quad l \in [0, 1] \tag{7}$$

where L_l, U_l and f_l respectively, denote the nodes' features, weights and non-linear function (ReLU) in the l-layer. It's worth mentioning that the input of this sub-network contains L and $\hat{A}$, and the output is a $C \times D$ label co-occurrence embeddings matrix W. Next, we will fuse W with the image feature F to generate the predicted labels for each input image.

Note that in the GCN propagation process, the feature of each node may become indistinguishable due to the over-smoothing problem. To address this issue, similar to A-GCN [19], we design L2-norm loss to enforce a spare correlation constraint on $\hat{A}$ as follows:

$$L_A = ||\hat{A} - I_C||_2. \tag{8}$$

where $||\cdot||_2$ denotes the L2-norm distance, $\hat{A}$ denotes the normalized correlation matrix and I_C is a $C \times C$ identity matrix. On the one hand, L_A can avoid over-smoothed features in GCN by highlighting self-correlation weights. On the other hand, L_A will be combined with the multi-label loss function (see Eq. 10) to optimize our model.

3.3 Loss function

As shown in Fig. 2, after obtaining both the D-dimensional image feature F and $C \times D$-dimensional label co-occurrence embeddings matrix W, we directly use dot product to fuse W and F to generate the C-dimensional predicted labels Y for image I:

$$Y = W \otimes F, \tag{9}$$

where $\otimes$ denotes the dot product operation. This process can integrate the label dependencies into image features generated from the above transformer. After that, we adopt a commonly-used multi-label loss function to calculate the loss between Y and the C-dimensional ground truth labels $\hat{Y}$ of image I:

$$\begin{aligned} L_M = \sum_{i=1}^{C} & \hat{Y}^i \log(\mathrm{Sigmoid}(Y^i)) \\ & + (1 - \hat{Y}^i) \log(1 - \mathrm{Sigmoid}(Y^i)), \end{aligned} \tag{10}$$

where Sigmoid is the commonly-used activation function, $\hat{Y}^i$ denotes the i-th element of $\hat{Y}$ and $\hat{Y}^i = \{1, 0\}$ denotes whether the label i appears in the image I or not. At last, we combine L_M and L_A (see Eq. 8) to train the whole network in an end-to-end manner as follows:

$$L_{\text{total}} = L_M + \lambda L_A, \tag{11}$$

where $\lambda \in [0, 1]$ is a weighted factor to balance these two losses.

4 Experiments

In this section, we first describe the experimental settings including the implementation details and evaluation metrics, then briefly introduce the datasets, and finally analyze the experimental results of STMG in detail.

4.1 Experimental settings

4.1.1 Implementation details

All the experiments are implemented on a Linux server with 4 Tesla V100 GPUs using PyTorch. In the image representation learning module, we employ Swin-L [21] to generate the image feature for each image with 384 × 384 resolution. Note that we get rid of the last fully-connected (fc) classification layer of Swin-L (which contains $K = 4$ blocks as default) and obtain the 1536-dimensional (i.e., $D = 1536$) feature vector F from the average pooling layer. In the label co-occurrence embedding module, we utilize the Bert [9] model to output the 768-dimensional (i.e., $d = 768$) word embedding for each label word. The output of two convolution branches in Eq. (5) are both $C \times 1024$ (i.e., $d_L = 1024$) matrices. In addition, we design GCN with two layers to, respectively, output 1024 and 1536 dimensional label co-occurrence embeddings after each layer. Our model will be trained using stochastic gradient descent (SGD) to update all the network parameters with a momentum of 0.9, a batchsize of 32, a weight decay of 10^{-4}, and an initial learning rate of 0.4 which decays by a factor of 10 every 40 epochs. One can also see our code for more details.

4.1.2 Evaluation metrics

(1) Following mainstream methods [5, 19, 33], we report two important evaluation metrics to measure the multi-label image classification performance, i.e., the average precision (AP) for each category and mean average precision (mAP) over all categories. In addition, we also calculate the per-class precision (CP), per-class recall (CR), per-class F1 (CF1) and the overall precision (OP), overall recall (OR), overall F1 (OF1). For fair comparisons with existing methods, we further list the experimental results (i.e., CP-3, CR-3, CF1-3, OP-3, OR-3, OF1-3) on top-3 labels of the classification scores. (2) We also compare the convergence efficiency (i.e., how many epochs it will take to achieve the highest mAP) of our model with other state-of-the-art methods.

4.2 Datasets

MS-COCO [20] contains 80 semantic label concepts with 82,783 training images, 40,504 validation images and 40,775 test images. Owing to that the ground truth labels of the test set are not available, we train STMG on the train set and evaluate its performance on the validation set.

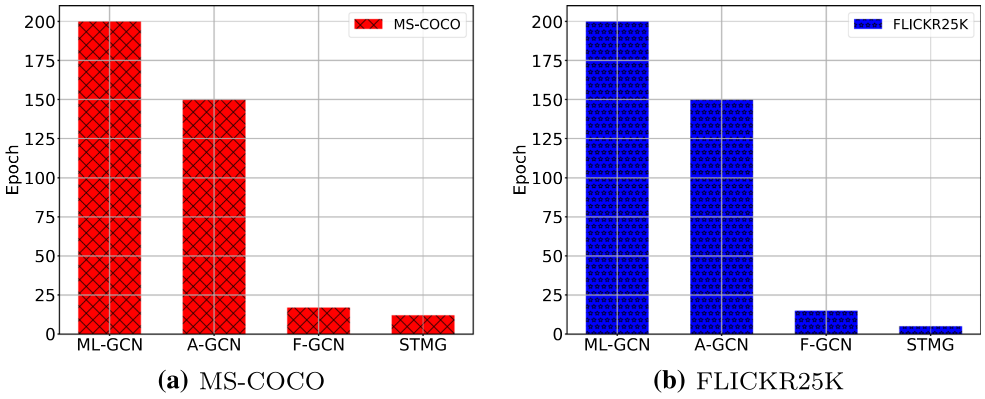
FLICKR25K [16] consists of 25,000 multi-label images that belong to 24 labels, and each image is averagely annotated by 4.7 labels. We randomly split the train set and test set with a ratio of 5:5, and then evaluate the performance on the test set.

4.3 Experimental results

4.3.1 Convergence efficiency

In this section, we compare the convergence efficiency of STMG with ML-GCN [5], A-GCN [19] and F-GCN [33]. We record how many epochs it will take when these models, respectively, achieve the highest mAP.

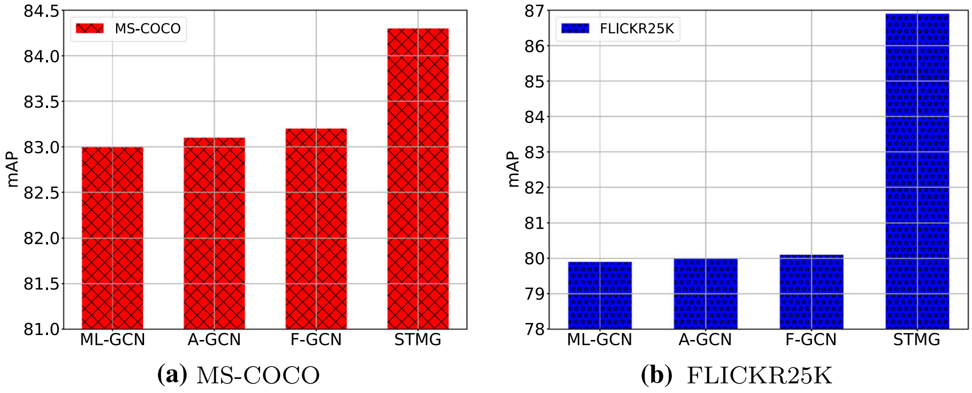
As shown in Fig. 5a, ML-GCN and A-GCN will, respectively, take about 200 and 150 epochs to converge to achieve the optimal performance. This low convergence efficiency mainly lies in that both ML-GCN and A-GCN first adopt ResNet-101 to extract image features, and then use dot product to fuse image features and label embeddings. However, as we know, these two embeddings come from different modalities, so dot product will severely limit the model convergence. F-GCN only takes 17 epochs to complete its training process because it adds a cross-modal component to greatly speed up the model convergence, but on the other hand, it also increases the model complexity with too many components. To our surprise, STMG, which also adopts dot product, efficiently finishes its training process within 12 epochs. This is because our STMG adopts transformer to extract image features in a NLP manner, which gets rid of the cross-modal fusion process when combined with label embeddings to generate the classification results. Besides, as shown in Fig. 6a, ML-GCN, A-GCN and F-GCN all achieve the mAP value around 83.1 ± 0.1, but to our satisfaction, STMG obtains a 84.3 mAP within much fewer epochs. On the one hand, Swin transformer can effectively capture the similarity between patches to learn the global image feature compared to CNNs. On the other hand, the adaptive label co-occurrence embeddings can help integrate the label dependencies into image features to further promote the classification performance. Figures 5a and 6a illustrate the



(a) MS-COCO **(b)** FLICKR25K

Fig. 5 Convergence efficiency comparisons of STMG with ML-GCN, A-GCN and F-GCN

Fig. 6 mAP comparisons of STMG with ML-GCN, A-GCN and F-GCN



(a) MS-COCO **(b)** FLICKR25K

efficiency and effectiveness of our designed STMG on MS-COCO.

We further collect the convergence efficiency and mAP results of these methods on FLICKR25K and find our STMG obtains a more obvious superiority than others. As shown in Fig. 5b, there appears a similar convergence trend among these four models. Our STMG again efficiently converges within 5 epochs owing to its NLP based feature extraction way, but ML-GCN and A-GCN have to spend hundreds of epochs to complete the training process using the same dot product operation. Besides, as shown in Fig. 6b, ML-GCN, A-GCN and F-GCN, respectively, achieve the mAP values around 80.0 ± 0.1, our STMG obtains the highest 86.9 mAP within much fewer epochs. These experimental results on MS-COCO and FLICKR25K demonstrate the effectiveness that integrating the label dependencies between objects into Swin transformer can help boost the multi-label image recognition performance. Both Figs. 5 and 6 verify the superiority of our method. Furthermore, we also compare other evaluation metrics of STMG with other state-of-the-art methods in Sect. 4.3.2.

4.3.2 Comparisons with the state-of-the-art methods

In this section, we compare the multi-label image recognition performance (i.e., mAP, AP, CP, CR, CF1, OP, OR, OF1, CP-3, CR-3, CF1-3, OP-3, OR-3, OF1-3) of STMG with the state-of-the-art methods on MS-COCO and FLICKR25K.

Results on MS-COCO In this part, we compare the performance of STMG with the state-of-the-art multi-label image recognition methods including CNN-RNN [30], RNN-Attention [34], Order-Free RNN [3], ML-ZSL [18], SRN [37], Multi-Evidence [11], ResNet-101 [14], ML-GCN [5], A-GCN [19] and F-GCN [33] on MS-COCO. As shown in Table 1, STMG obviously outperforms all candidate methods by at least 1.1% mAP improvement and also produces higher results on all other evaluation metrics. Besides, as we see, compared with those methods (such as ResNet-101) that did not adopt GCN, STMG greatly promotes the performance with 6.9% mAP improvement. The main reason is that we construct a label graph that can adaptively capture the relationships between different objects to help generate more accurate image features. Especially, we list the representative metric (i.e., mAP) between STMG, ML-GCN, A-GCN and F-GCN in Fig. 6a,

Table 1 Performance comparisons on MS-COCO. Note that DP denotes the dot product operation

Method	All							Top-3					
	mAP	CP	CR	CF1	OP	OR	OF1	CP	CR	CF1	OP	OR	OF1
CNN-RNN	61.2	–	–	–	–	–	–	66.0	55.6	60.4	69.2	66.4	67.8
RNN-Attention	–	–	–	–	–	–	–	79.1	58.7	67.4	84.0	63.0	72.0
Order-Free RNN	–	–	–	–	–	–	–	71.6	54.8	62.1	74.2	62.2	67.7
ML-ZSL	–	–	–	–	–	–	–	74.1	64.5	69.0	–	–	–
SRN	77.1	81.6	65.4	71.2	82.7	69.9	75.8	85.2	58.8	67.4	87.4	62.5	72.9
Multi-Evidence	–	80.4	70.2	74.9	85.2	72.5	78.4	84.5	62.2	70.6	89.1	64.3	74.7
ResNet-101	77.3	80.2	66.7	72.8	83.9	70.8	76.8	84.1	59.4	69.7	89.1	62.8	73.6
ML-GCN (DP)	83.0	85.1	72.0	78.0	85.8	75.4	80.3	89.2	64.1	74.6	90.5	66.5	76.7
A-GCN (DP)	83.1	84.7	72.3	78.0	85.6	75.5	80.3	89.0	64.2	74.6	90.5	66.3	76.6
F-GCN	83.2	85.4	72.4	78.3	86.0	75.7	80.5	**89.3**	64.3	74.7	90.5	66.6	76.7
STMG (DP)	**84.3**	**85.8**	**72.7**	**78.7**	**86.7**	**76.8**	**81.5**	**89.3**	**64.8**	**75.1**	**90.8**	**67.4**	**77.4**

Bold represents the highest value in the corresponding evaluation metric

which illustrates that STMG exceeds these CNN baseline methods by at least 1.1% mAP on MS-COCO by using Swin transformer to effectively learn the global image features.

Results on FLICKR25K In this part, we further compare STMG with ML-GCN [5], A-GCN [19] and F-GCN [33] on FLIKCR25K. In addition, we also compare the classification results with the state-of-the-art methods including ADD-GCN [35], VACIT [13] and SS-GRL [4]. We present the experimental results in Table 2. As we see, STMG achieves a higher mAP value than other candidates by 6.5–7.0%. In addition, STMG also achieves obvious superiority on all the precision, recall and F1-scores. Similarly, the main reasons lie in two parts. One is that Swin transformer can more effectively extract the global image features compared to CNN baseline methods on single-label images. We improve it to adapt to our multi-label image recognition task and successfully learn the accurate image representations. The other is that our STMG can adaptively capture the label dependencies according to the label embeddings, which further helps Swin transformer to generate better classification results. Especially, we list the representative metric (i.e., mAP) between STMG, ML-GCN, A-GCN and F-GCN in Fig. 6b, which illustrates that STMG exceeds these three baseline methods by at least 6.5% mAP on FLICKR25K by using Swin transformer to effectively learn the global image features.

4.3.3 Ablation studies

In this section, we conduct ablation studies to explore how different settings will influence the performance of our model, including the word vector methods for label embeddings, the dimension of d_L (see Eq. 5), different GCN layers, and weighted factor λ (see Eq. 11) in the loss function L_{total}.

Word vector methods for label embeddings In this part, we utilize different popular word vector methods including Bert [9], GloVe [24] and GoogleNews [22] to generate the label embedding for each label word. We record the mAP change on MS-COCO and FLICKR25K in Table 3. As we see, although Bert obtains higher mAP values on these two datasets than others, different word vector methods will not

Table 2 Performance comparisons on FLICKR25K. Note that DP denotes the dot product operation

Method	All							Top-3					
	mAP	CP	CR	CF1	OP	OR	OF1	CP	CR	CF1	OP	OR	OF1
ML-GCN (DP)	79.9	79.2	67.9	73.1	83.5	74.8	78.9	85.1	49.3	62.4	88.6	57.1	69.4
A-GCN (DP)	80.0	79.1	68.0	73.1	83.3	75.1	79.0	85.1	49.4	62.5	88.7	57.2	69.5
F-GCN	80.1	79.1	68.0	73.1	83.4	75.2	79.1	85.1	49.5	62.6	88.8	57.2	69.6
ADD-GCN	80.1	79.2	68.1	73.2	83.4	75.2	79.1	85.2	49.5	62.6	88.8	57.3	69.7
VACIT	80.4	83.6	64.9	72.5	86.8	73.3	79.5	85.7	48.9	62.3	89.2	57.6	70.0
SS-GRL	80.1	79.3	68.0	73.2	83.6	75.3	79.2	85.3	49.6	62.7	89.0	57.4	69.8
STMG	**86.9**	**84.4**	**75.7**	**79.8**	**87.5**	**82.0**	**84.7**	**91.6**	**51.9**	**66.2**	**92.7**	**61.8**	**74.1**

Bold represents the highest value in the corresponding evaluation metric

Table 3 The change of mAP using different word vector methods for label embeddings

Word vector methods	mAP	
	MS-COCO	FLICKR25K
Bert	**84.3**	**86.9**
Glove	84.2	86.7
GoogleNews	84.2	**86.9**

Bold represents the highest value in the corresponding evaluation metric

affect the model performance too much. Nevertheless, Bert is a more powerful word embedding method trained on large text corpus, which can better maintain the semantic topology between objects, and our STMG greatly benefits from this scheme to better learn the label co-occurrence embeddings. Therefore, we adopt Bert to generate the label embedding for each label word.

The dimension of d_L As mentioned in Eq. (5), we use two convolution layers to first generate the intermediate $C \times d_L$ matrices, then adopt the dot product operation to fuse these two matrices to obtain the correlation matrix. In this part, we first vary the dimension of d_L, then observe the change of performance on MS-COCO and FLICKR25K. As shown in Fig. 7, we respectively set $d_L = 128, 256, 512, 1024, 2048$ to calculate the mAP on these two datasets. As we see, the performance will continue to rise when we increase d_L from 128 to 1024, but it will drop when we increase d_L from 1024 to 2048. In fact, d_L acts as a bridge between 768-dimensional label embeddings and 1536-dimensional label co-occurrence embeddings. The possible reason is that $d_L = 1024$ can make a better and smooth transition from 768-dimensional label embeddings to 1536-dimensional label co-occurrence embeddings, while other values of d_L may lead to slight instability in the GCN training. We believe $d_L = 1024$ is a better dimension to learn relationship between different label embeddings to generate a more effective correlation matrix.

Table 4 The change of mAP using different number of GCN layers

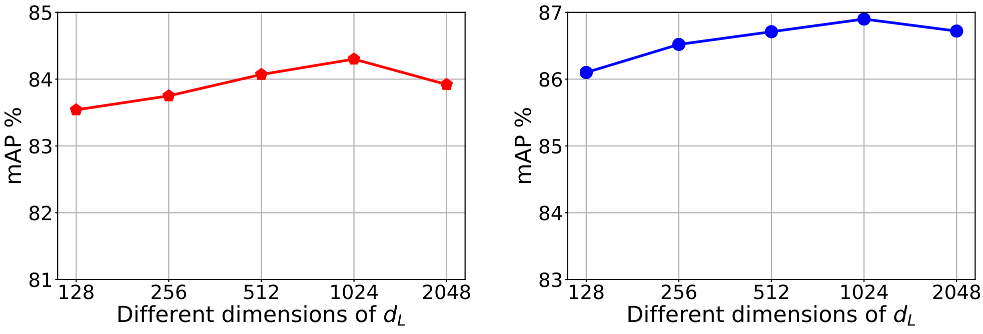
Number of GCN layers	mAP	
	MS-COCO	FLICKR25K
2	**84.3**	**86.9**
3	83.8	86.3
4	83.4	85.8

Bold represents the highest value in the corresponding evaluation metric

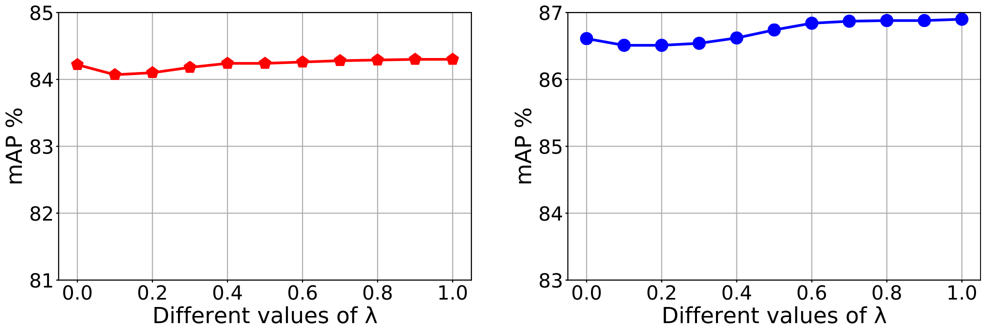
Different GCN layers In this part, we respectively design GCN with two, three and four layers to conduct the experiments and record the change of performance on MS-COCO and FLICKR25K. As shown in Table 4, when using two GCN layers, we can achieve the highest mAP on MS-COCO and FLICKR25K. However, when increasing the layers of GCN, the performance is gradually decreasing on all datasets. This is because the features of nodes may become indistinguishable in the propagation process with multiple accumulated layers, which will reduce the image recognition effect. Therefore, in the experiments, we choose two GCN layers to complete the label graph learning process.

Weighted factor λ in the loss function L_{total} When training our model, we combine the multi-label loss and L2-norm loss to update the network. The weighted factor λ indicates the contribution of L_A in the whole loss L_{total}. In this part, we vary λ from 0 to 1 and record the change of mAP on MS-COCO and FLICKR25K in Fig. 8. As we see, when we set $\lambda = 1$, the performance on these two datasets will achieve the highest values. We believe this regularization helps optimize the model via avoiding the over-smoothing problem in GCN. Therefore, we choose $\lambda = 1$ as the default setting of STMG.

Fig. 7 The change of mAP using different dimensions of d_L



(a) MS-COCO **(b)** FLICKR25K



(a) MS-COCO **(b)** FLICKR25K

Fig. 8 The change of mAP using different values of λ

5 Conclusion

In this paper, we propose STMG that combines transformer and GCN to extract the image features and learn the label dependencies for multi-label image recognition. STMG consists of an image representation learning module and a label co-occurrence embedding module. Firstly, in the image representation learning module, to avoid computing the similarity between each two patches, we adopt Swin transformer instead of ViT to generate the image feature for each input image. Secondly, in the label co-occurrence embedding module, we design a two-layer GCN to adaptively capture the label dependencies to output the label co-occurrence embeddings. At last, STMG fuses the image feature and label co-occurrence embeddings to produce the image classification results with the commonly-used multi-label classification loss function and a L2-norm loss function. We conduct extensive experiments on two multi-label image datasets including MS-COCO and FLICKR25K. Experimental results demonstrate STMG can achieve better performance including the convergence efficiency and image classification results compared with the state-of-the-art multi-label image recognition methods.

Acknowledgements Thanks for the support of the National Natural Science Foundation of China No.61871139 and the International Science and Technology Cooperation Projects of Guangdong Province No.2020A0505100060.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Ba LJ, Kiros JR, Hinton GE (2016) Layer normalization. CoRR **abs/1607.06450**
2. Carion N, Massa F, Synnaeve G, Usunier N, Kirillov A, Zagoruyko S (2020) End-to-end object detection with transformers. In: Vedaldi A, Bischof H, Brox T, Frahm J (eds) Computer vision - ECCV 2020 - 16th European conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I. Lecture Notes in Computer Science, vol 12346, Springer, pp. 213–229
3. Chen S, Chen Y, Yeh C, Wang YF (2018) Order-free RNN with visual attention for multi-label classification. In: McIlraith SA, Weinberger KQ (eds) Proceedings of the thirty-second AAAI conference on artificial intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th aaai symposium on educational advances in artificial intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2–7, 2018. AAAI Press, pp 6714–6721
4. Chen T, Xu M, Hui X, Wu H, Lin L (2019) Learning semantic-specific graph representation for multi-label image recognition. In: 2019 IEEE/CVF international conference on computer vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019. IEEE, pp 522–531
5. Chen Z, Wei X, Wang P, Guo Y (2019) Multi-label image recognition with graph convolutional networks. In: ieee conference on computer vision and pattern recognition, CVPR 2019, Long Beach, CA, USA, June 16–20, 2019. Computer Vision Foundation/IEEE, pp 5177–5186
6. Defferrard M, Bresson X, Vandergheynst P (2016) Convolutional neural networks on graphs with fast localized spectral filtering. In: D.D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, R. Garnett (eds.) Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain, pp. 3837–3845
7. Deng J, Dong W, Socher R, Li L, Li K, Li F (2009) Imagenet: a large-scale hierarchical image database. In: 2009 IEEE computer society conference on computer vision and pattern recognition (CVPR 2009), 20–25 June 2009, Miami, Florida, USA. IEEE Computer Society, pp 248–255
8. Devlin J, Chang M, Lee K, Toutanova K (2018) BERT: pre-training of deep bidirectional transformers for language understanding. CoRR abs/1810.04805
9. Devlin J, Chang M, Lee K, Toutanova K (2019) BERT: pre-training of deep bidirectional transformers for language understanding. In: Burstein J, Doran C, Solorio T (eds) Proceedings of the 2019 conference of the north american chapter of the association for computational linguistics: human language technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2–7, 2019, vol 1 (Long and Short Papers). pp 4171–4186. Association for Computational Linguistics
10. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S, Uszkoreit J, Houlsby N (2020) An image is worth 16x16 words: transformers for image recognition at scale. CoRR abs/2010.11929

11. Ge W, Yang S, Yu Y (2018) Multi-evidence filtering and fusion for multi-label classification, object detection and semantic segmentation based on weakly supervised learning. In: 2018 IEEE conference on computer vision and pattern recognition, CVPR 2018, Salt Lake City, UT, USA, June 18–22, 2018. IEEE Computer Society, pp 1277–1286
12. Girdhar R, Carreira J, Doersch C, Zisserman A (2019) Video action transformer network. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16–20, 2019. Computer Vision Foundation/IEEE, pp 244–253
13. Guo H, Zheng K, Fan X, Yu H, Wang S (2019) Visual attention consistency under image transforms for multi-label image classification. In: IEEE conference on computer vision and pattern recognition, CVPR 2019, Long Beach, CA, USA, June 16–20, 2019. Computer Vision Foundation/IEEE, pp 729–739
14. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: 2016 IEEE conference on computer vision and pattern recognition, CVPR 2016, Las Vegas, NV, USA, June 27–30, 2016. IEEE Computer Society, pp 770–778
15. He T, Jin X (2019) Image emotion distribution learning with graph convolutional networks. In: El-Saddik A, Bimbo, AD, Zhang Z, Hauptmann AG, Candan KS, Bertini M, Xie L, Wei X (eds) Proceedings of the 2019 on international conference on multimedia retrieval, ICMR 2019, Ottawa, ON, Canada, June 10–13, 2019. ACM, pp 382–390
16. Huiskes MJ, Lew MS (2008) The MIR flickr retrieval evaluation. In: Lew MS, Bimbo AD, Bakker EM (eds) Proceedings of the 1st ACM SIGMM international conference on multimedia information retrieval, MIR 2008, Vancouver, British Columbia, Canada, October 30–31, 2008. ACM, pp 39–43
17. Kipf TN, Welling M (2017) Semi-supervised classification with graph convolutional networks. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24–26, 2017, Conference Track Proceedings. OpenReview.net
18. Lee C, Fang W, Yeh C, Wang YF (2018) Multi-label zero-shot learning with structured knowledge graphs. In: 2018 IEEE conference on computer vision and pattern recognition, CVPR 2018, Salt Lake City, UT, USA, June 18–22, 2018. IEEE Computer Society, pp 1576–1585
19. Li Q, Peng X, Qiao Y, Peng Q (2020) Learning label correlations for multi-label image recognition with graph networks. Pattern Recognit Lett 138:378–384
20. Lin T. Maire M, Belongie SJ, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL: Microsoft COCO: common objects in context. In: Fleet DJ, Pajdla T, Schiele B, Tuytelaars T (eds) Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V. Lecture Notes in Computer Science, vol 8693. Springer, pp 740–755 (2014)
21. Liu Z, Lin Y, Cao Y, Hu H, Wei Y, Zhang Z, Lin S, Guo B (2021) Swin transformer: hierarchical vision transformer using shifted windows. CoRR abs/2103.14030
22. Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. In: Bengio Y, LeCun Y (eds) 1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2–4, 2013, Workshop Track Proceedings
23. Neimark D, Bar O, Zohar M, Asselmann D (2021) Video transformer network. CoRR abs/2102.00719
24. Pennington J, Socher R, Manning CD (2014) Glove: Global vectors for word representation. In: Moschitti A, Pang B, Daelemans W (eds) Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25–29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL. ACL, pp 1532–1543
25. Sun C, Shrivastava A, Singh S, Gupta A (2017) Revisiting unreasonable effectiveness of data in deep learning era. In: IEEE international conference on computer vision, ICCV 2017, Venice, Italy, October 22-–29, 2017. IEEE Computer Society, pp 843–852
26. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In: 2016 IEEE conference on computer vision and pattern recognition, CVPR 2016, Las Vegas, NV, USA, June 27–30, 2016, IEEE Computer Society, pp 2818–2826
27. Tan H, Bansal M (2019) LXMERT: learning cross-modality encoder representations from transformers. In: Inui K, Jiang J, Ng V, Wan X (eds) Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3–7, 2019. Association for Computational Linguistics, pp. 5099–5110
28. Touvron H, Cord M, Douze M, Massa F, Sablayrolles A, Jégou H (2020) Training data-efficient image transformers and distillation through attention. CoRR abs/2012.12877
29. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. In: Guyon I, von Luxburg U, Bengio S, Wallach HM, Fergus R, Vishwanathan SVN, Garnett R (eds) Advances in neural information processing systems 30: annual conference on neural information processing systems 2017, December 4–9, 2017, Long Beach, CA, USA. pp 5998–6008
30. Wang J, Yang Y, Mao J, Huang Z, Huang C, Xu W (2016) CNN-RNN: A unified framework for multi-label image classification. In: 2016 IEEE conference on computer vision and pattern recognition, CVPR 2016, Las Vegas, NV, USA, June 27–30, 2016. IEEE Computer Society, pp 2285–2294
31. Wang X, Ye Y, Gupta A (2018) Zero-shot recognition via semantic embeddings and knowledge graphs. In: 2018 IEEE conference on computer vision and pattern recognition, CVPR 2018, Salt Lake City, UT, USA, June 18–22, 2018. IEEE Computer Society, pp 6857–6866
32. Wang Y, Song J, Zhou K, Liu Y (2021) Unsupervised deep hashing with node representation for image retrieval. Pattern Recognit 112:107785
33. Wang Y, Xie Y, Liu Y, Zhou K, Li X (2020) Fast graph convolution network-based multi-label image recognition via cross-modal fusion. In: d'Aquin M, Dietze S, Hauff C, Curry E, Cudré-Mauroux P (eds) CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19–23, 2020. ACM, pp 1575–1584
34. Wang Z, Chen T, Li G, Xu R, Lin L (2017) Multi-label image recognition by recurrently discovering attentional regions. In: IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22–29, 2017. IEEE Computer Society, pp. 464–472
35. Ye J, He J, Peng X, Wu W, Qiao Y (2020) Attention-driven dynamic graph convolutional network for multi-label image recognition. In: Vedaldi A, Bischof H, Brox T, Frahm J (eds) Computer vision - ECCV 2020 - 16th European cnference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI. Lecture Notes in Computer Science, vol 12366. Springer, pp 649–665
36. Zheng S, Lu J, Zhao H, Zhu X, Luo Z, Wang Y, Fu Y, Feng J, Xiang T, Torr PHS, Zhang L (2020) Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. CoRR abs/2012.15840
37. Zhu F, Li H, Ouyang W, Yu N, Wang X (2017) Learning spatial regularization with image-level supervisions for multi-label image classification. In: 2017 IEEE conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21–26, 2017. IEEE Computer Society, pp. 2027–2036