



Graph Attention Transformer Network for Multi-Label Image Classification

JIN YUAN*, Southeast University, China

SHIKAI CHEN, Southeast University, China

YAO ZHANG, University of Chinese Academy of Sciences, China

ZHONGCHAO SHI, Lenovo Research, China

XIN GENG†, Southeast University, China

JIANPING FAN, Lenovo Research, China

YONG RUI†, Lenovo Research, China

Multi-label classification aims to recognize multiple objects or attributes from images. The key to solving this issue relies on effectively characterizing the inter-label correlations or dependencies, which bring the prevailing graph neural network (GNN). However, current methods often use the co-occurrence probability of labels based on the training set as the adjacency matrix to model this correlation, which is greatly limited by the dataset and affects the model's generalization ability. This paper proposes a Graph Attention Transformer Network (GATN), a general framework for multi-label image classification by mining rich and effective label correlation. First, we use the cosine similarity value of the pre-trained label word embedding as the initial correlation matrix, which can represent richer semantic information than the co-occurrence one. Subsequently, we propose the graph attention transformer layer to transfer this adjacency matrix to adapt to the current domain. Our extensive experiments have demonstrated that our proposed methods can achieve highly competitive performance on three datasets.

CCS Concepts: • **Computing methodologies** → **Object recognition**.

Additional Key Words and Phrases: Graph Neural Network; Transformer, Attention Mechanism, Multi-Label Classification

1 INTRODUCTION

Multi-label image classification is a basic visual task that aims to recognize multiple objects or attributes in an image. It has wide real-world applications, like human expression classification [4, 23, 58, 61], social tag recommendation [18, 31, 45, 46], and medical image-aided diagnosis [15]. The most critical challenge for multi-label image classification is to learn the inter-label relationships and their dependencies in various images.

Motivated by the powerful ability of graph neural networks (GNN) in processing relation information, current works [8, 33, 49] often exploited GNN to model the correlation between categories, thereby achieving excellent results for multi-label classification. As a core of GNN, the relation adjacency matrix represent the relational strength of different nodes (categories), which motivates us to solve multi-label classification by enhancing the

*This work is done when Jin Yuan is an intern at Lenovo Research.

†Corresponding authors.

Authors' addresses: Jin Yuan, Southeast University, Nanjing, China, yuanjin@seu.edu.cn; Shikai Chen, Southeast University, Nanjing, China, skchen@seu.edu.cn; Yao Zhang, University of Chinese Academy of Sciences, Beijing, China, zhangyao215@mailsucas.ac.cn; Zhongchao Shi, Lenovo Research, Beijing, China, shizc2@lenovo.com; Xin Geng, Southeast University, Nanjing, China, xgeng@seu.edu.cn; Jianping Fan, Lenovo Research, Beijing, China, jfan1@lenovo.com; Yong Rui, Lenovo Research, Beijing, China, yongrui@lenovo.com.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2022 Association for Computing Machinery.

1551-6857/2022/12-ART \$15.00

<https://doi.org/10.1145/3578518>

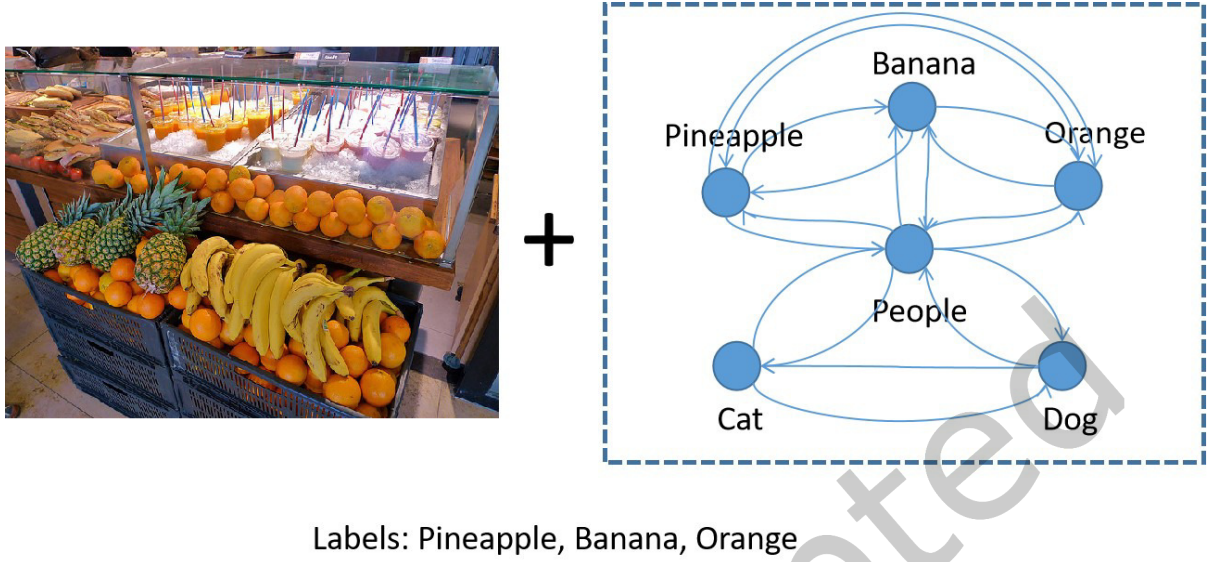


Fig. 1. This figure shows the input for GNN-based methods. The left part is the input image, and the part on the right is the topological diagram of the label relationship, usually represented by the adjacency matrix.

expressive ability of the relation adjacency matrix. We consider the issue faced by the adjacency matrix as two parts: (1) How to define the relationships between different categories. (2) How to transfer this knowledge into a specific task.

For these two problems, conventional methods [8, 33, 49] often adopted a fixed co-occurrences matrix as a relation adjacency matrix. However, relations include more than just appearing together, they also include similar appearances (e.g. *bear* and *teddy bear*), similar functions (e.g. *airplane* and *truck*) and etc, which are difficult to learn from image datasets without human annotation. On the other hand, a billion-level corpus contains a wealth of knowledge and information, which is often used for generating expressive word embedding. Therefore, the natural idea is to leverage pre-trained word embeddings [10, 34] to build an adjacency relation matrix instead of a co-occurrence one.

Given a relation matrix with rich information, how to adapt it to a specific task is another crucial issue. Inspired by the remarkable ability of the self-attention mechanism in processing word vector relationships, we find that the self-attention mechanism is essentially a graph-like structure, which is helpful in extracting the inter-label relationships. Based on this observation, we propose a novel graph attention transformer layer to transfer the correlation matrix by leveraging the attention block to mine its internal sub-graph structure. Generally, our proposed graph attention transformer network can effectively borrow information from the adjacency matrix and then adapt it to the specific task.

Our contributions are as follows:

- We propose a novel end-to-end GATN framework to enhance the representative ability of the adjacency matrix in the GNN-based model.
- We initialize the node adjacency matrix in the graph by the label node embedding, which has richer semantic information than the previous co-occurrence probability method.
- We evaluate our proposed model in three datasets, and the results demonstrate the highly competitive performance compared with the existing methods.

2 RELATED WORKS

2.1 Multi-label Classification

As a fundamental task in computer vision, multi-label classification has drawn much attention in past years. Some early methods solved this task by dividing the task into multiple binary classification tasks [27, 36, 38]. With the rise of convolutional neural networks (CNN), some works [14, 32, 54] used the CNN-based method to complete this task. These methods usually used binary cross-entropy and sigmoid function to learn image features, thereby neglecting the information between the labels. Therefore, some approaches adopted a CNN-RNN architecture [5, 25, 47, 55] to predict labels in order. Wang et al. [47] emphasized the importance of predicting simple labels first and then predicting other labels with the previous prediction result information. To break the limitation of predicting labels according to frequency, Chen et al. [5] exploited a long short term memory (LSTM) network to generate label order, making the prediction order learning parameters. Yazici et al. [55] proposed a predicted label alignment to avoid the further misclassification caused by the misalignment of the label order.

2.2 Graph Neural Network

The graph neural network has proven to be helpful in a variety of tasks [16, 35, 41, 42, 44, 57], like solving node classification in social networks or assisting other tasks to extract connections between nodes. As variants of graph neural networks, many networks have different advantages in processing graph structure data. Graph convolutional network (GCN) [2, 19] was introduced as a scalable method for semi-supervised learning on graph-structured data, and it can effectively extract the features between graph nodes. On the other hand, the graph transformer network [56] was designed to solve the problem of the heterogeneous graph by mining the structure of different sub-graphs to generate new meta-paths.

Due to the powerful ability of graph neural networks in dealing with the data relationship for graph-structured data, some works [8, 33, 49] solve multi-label classification based on image features and label topological structure. Chen et al. [8] exploited GCN to gain label features by inputting label word embedding and co-occurrence matrix as node representation and adjacency matrix, respectively. They achieved state-of-the-art performance at that time by blending the label features and image features learned by GCNs and brought the trend of studying multi-label image classification by graph neural network. Wang et al. [49] proposed a label graph superimposing framework to enhance the graph-based network further. To mine more sub-graph structures, Nguyen et al. [33] introduced a modular architecture with a graph transformer network by regarding the relation between categories as a heterogeneous graph.

2.3 Attention Mechanism

The attention mechanism was designed to mine the relevance between different entities. Due to its great success in relational understanding, the attention mechanism began to be widely used in Natural Language Processing (NLP) to understand the relationship between different words [1, 10, 13]. As a core of the transformer, the attention mechanism regained its brilliance in the Computer Vision (CV) [3, 11, 28, 43, 59] and its extraction of entity relationships was very competitive. For graph network, Graph Attention Network (GAT) [44, 48] combines attention mechanism and graph network node-wise by assigning different weights for its neighbors. However, GAT can only aggregate the weights of existing neighbors but can not transfer the relations to a specific task. Therefore, we try to design a network to apply the attention mechanism to the adjacency matrix and then complete the layer-wise aggregation through the Laplacian matrix by adaptively modifying the neighbor nodes.

2.4 Transformer

Recently, the prevailing transformer has achieved many state-of-the-art results in different tasks [37, 43]. As a combination of attention mechanism and multi-layer perception, the transformer firstly boosts the performance in

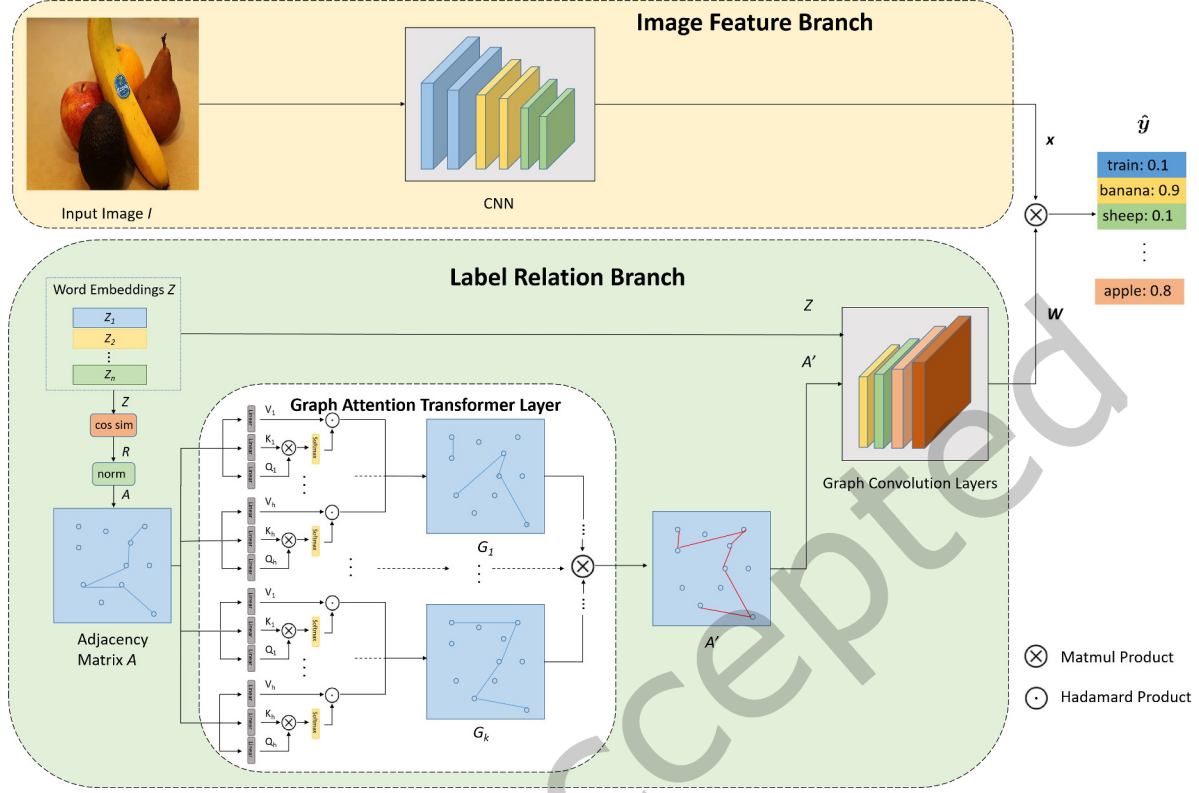


Fig. 2. The overview of our method. In the label relation branch, we first initialize the adjacency matrix by cosine similarity of the label node embedding. Then we adopt a graph attention transformer layer to mine sub-graph structures.

natural language processing and gradually extends to vision tasks [28–30]. In multi-label classification [20, 26, 39], Lanchantin et al. [20] leveraged transformer to mine the dependencies between features and labels. Tal et al. [39] adopted a transformer as the prediction head to replace the conventional fully-connected layer. Nevertheless, these methods inserted the transformer in the classifier, which has high overheads because of directly manipulating the extracted features. Unlike these methods, our proposed graph attention transformer layer is more efficient since we only operate on the adjacency matrix in the graph, which is only related to the number of categories.

3 METHOD

Our proposed framework has two branches. The image feature branch uses a common pre-trained model to extract image features, and the label relation branch aims to learn the relationship among labels. In the label relation branch, our proposed Graph Attention Transformer Network focuses on learning adequate label node correlation information.

Firstly, we propose a novel generation scheme of correlation matrix based on the distance of node embedding. Then we exploit the graph attention transformer layer to obtain the weight matrix for transferring the generated matrix. The overview is shown in Fig. 2.

3.1 Generation of Correlation Matrix

For n categories multi-label classification, the adjacency matrix $A \in \mathbb{R}^{n \times n}$ is expected to include semantic relational information of labels. In a graph neural network, the node representations $Z \in \mathbb{R}^{n \times d}$ usually comes from word embeddings trained in an unsupervised large-scale corpus, which contain rich semantic information. Therefore, we use node embeddings to construct the correlation matrix.

Let Z_i denote embedding vector of i th label of length d , and $R_{i,j}$ represent the relation value between i th label and j th label, we obtain $R_{i,j}$ by calculating the cosine similarity of the embedding vectors:

$$R_{i,j} = \frac{\sum_{k=1}^d Z_{i,k} \times Z_{j,k}}{\sqrt{\sum_{k=1}^d (Z_{i,k})^2} \times \sqrt{\sum_{k=1}^d (Z_{j,k})^2}}. \quad (1)$$

The obtained matrix R is symmetric, $R_{i,j}$ is always equal to $R_{j,i}$, which means that the relationship between any two categories is consistent.

To avoid some noisy edges and over-smoothing problems, we use binary and re-weighted strategies to filter and re-weight values [8]. τ aims to filter the low-value relation. p represents the weights of neighbor nodes and the node itself, and it is designed to adjust the ratio of neighbor aggregated information. When p approaches 0, neighbor information tends to be ignored, and when p approaches 1, the feature of the node itself tends to be ignored [8]. With threshold τ and weight parameter p , we get final correlation matrix A :

$$R'_{i,j} = \begin{cases} 0, & \text{if } R_{i,j} < \tau \\ 1, & \text{if } R_{i,j} \geq \tau \end{cases}, \quad (2)$$

$$A_{i,j} = \begin{cases} p / \sum_{i \neq j}^n R'_{i,j}, & \text{if } i \neq j \\ 1 - p, & \text{if } i = j \end{cases}. \quad (3)$$

3.2 Graph Attention Transformer Layer

We design a novel graph transformer layer by focusing on node representation. Thus, our proposed graph attention transformer layer can transform the adjacency matrix into a new graph structure and explore new multi-hop paths. Define each sub-graph selected from adjacency matrix A as G_j . Each G_j can be regarded as a transformation of A with some new structure information. On the other hand, the self-attention mechanism has a powerful ability in structural transformation, which has led a great success for transformers in computer vision. Therefore, we use self-attention branches to obtain different sub-graphs.

We use standard multi-head self-attention layer similar to [43]. For input correlation matrix A , we can get Q_i , K_i , $V_i \in \mathbb{R}^{n \times D_h}$ for i th head through different linear layers:

$$\begin{aligned} [Q_i, K_i, V_i] &= A[W_i^Q, W_i^K, W_i^V] \\ W_i^Q, W_i^K, W_i^V &\in \mathbb{R}^{n \times D_h}, \end{aligned} \quad (4)$$

where $D_h = \frac{n}{h}$ is the hidden size of Q_i , K_i and V_i and h is the head number. Then the attention result $\text{Attention}(Q_i, K_i, V_i) \in \mathbb{R}^{n \times D_h}$ for i th head can be written as:

$$\text{Attention}(Q_i, K_i, V_i) = \text{softmax}\left(\frac{Q_i K_i^T}{\sqrt{D_h}}\right) V_i. \quad (5)$$

For a h -head attention layer, we can get a sub-graph adjacency matrix $G_j \in \mathbb{R}^{n \times n}$:

$$G_j = \text{Concat}(\text{Attention}(Q_1, K_1, V_1), \dots, \text{Attention}(Q_h, K_h, V_h))W^O. \quad (6)$$

Since we want each G_j to learn a different sub-graph structure, each G_j uses attention layers of non-shared parameters. For k generated sub-graphs adjacency matrix G_1, G_2, \dots, G_k , we intend to integrate them into the transformed adjacency matrix A' . The most straightforward solution is to adopt matrix multiplication:

$$A' = \prod_{j=1}^k G_j. \quad (7)$$

In general, the graph attention transformer layer can be regarded as a transfer function used to adapt the adjacency matrix to any feature extraction layer:

$$A' = f(A). \quad (8)$$

3.3 Graph Attention Transformer Networks

With the transformed adjacency matrix A' and node embedding Z , we adopt graph convolutional network [19] to learn useful representations for label node. Let $H^{(l)}$ be the label node representations of the l th layer in GCNs, the forward propagation can be written as:

$$H^{(l+1)} = \sigma \left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)} \right), \quad (9)$$

where $\tilde{A} \in \mathbb{R}^{n \times n}$ is the transformed adjacency matrix A' with self-connections, \tilde{D} is the degree matrix of \tilde{A} and $W^{(l)}$ is a learnable weight matrix. The input of GCNs is the node embedding Z , and the output is the final feature $W \in \mathbb{R}^{n \times D}$ of label node, where the D represent the dimensionality of image extracted features. Since A' is learned from graph attention transformer layer, our final output can learn the characteristics of the node from a more diverse label node representation space. In general, our label relation branch contains the generated matrix, graph attention transformer layer and GCNs, which is shown in Fig. 2.

For the image feature branch, let I denote an image, and common CNN networks can be used to extract high-dimensional features of the image. Subsequently, global max-pooling will be adopted to obtain the one-dimensional feature $x \in \mathbb{R}^D$ of the image:

$$x = f_{\text{GMP}}(f_{\text{CNN}}(I; \theta_{\text{CNN}})), \quad (10)$$

where θ_{CNN} denotes the parameters of CNN.

Generally, our entire network include two branches. The image feature branch takes image I as input and output extracted feature x . The label relation branch inputs node embedding Z , and output learned label representation W . Thus, we can get predicted score $\hat{y} \in \mathbb{R}^n$:

$$\hat{y} = Wx. \quad (11)$$

Let $y \in \mathbb{R}^n$ denote the ground-truth label space with n class labels, and $y^i = \{0, 1\}$ represents whether the i th label appears or not. The optimization goal is to minimize the following loss:

$$\mathcal{L} = \sum_{i=1}^n y^i \log(\sigma(\hat{y}^i)) + (1 - y^i) \log(1 - \sigma(\hat{y}^i)), \quad (12)$$

where $\sigma(\cdot)$ usually use sigmoid function.

Table 1. Comparison with state-of-the-art models in VOC2007

Methods	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	motor	person	plant	sheep	sofa	train	tv	mAP
CNN-RNN [47]	96.7	83.1	94.2	92.8	61.2	82.1	89.1	94.2	64.2	83.6	70.0	92.4	91.7	84.2	93.7	59.8	93.2	75.3	99.7	78.6	84.0
ResNet-101 [17]	99.5	97.7	97.8	96.4	65.7	91.8	96.1	97.6	74.2	80.9	85.0	98.4	96.5	95.9	98.4	70.1	88.3	80.2	98.9	89.2	89.9
FeV+LV [54]	97.9	97.0	96.6	94.6	73.6	93.9	96.5	95.5	73.7	90.3	82.8	95.4	97.7	95.9	98.6	77.6	88.7	78.0	98.3	89.0	90.6
HCP [51]	98.6	97.1	98.0	95.6	75.3	94.7	95.8	97.3	73.1	90.2	80.0	97.3	96.1	94.9	96.3	78.3	94.7	76.2	97.9	91.5	90.9
RNN-Attention [50]	98.6	97.4	96.3	96.2	75.2	92.4	96.5	97.1	76.5	92.0	87.7	96.8	97.5	93.8	98.5	81.6	93.7	82.8	98.6	89.3	91.9
Atten-Reinforce [7]	98.6	97.1	97.1	95.5	75.6	92.8	96.8	97.3	78.3	92.2	87.6	96.9	96.5	93.6	98.5	81.6	93.1	83.2	98.5	89.3	92.0
VGG [40]	99.4	97.4	98.0	97.0	77.9	92.4	96.8	97.8	80.8	93.4	87.2	98.0	97.3	95.8	98.8	79.4	95.3	82.2	99.1	91.4	92.8
DLDL [14]	99.3	97.6	98.3	97.0	79.0	95.7	97.0	97.9	81.8	93.3	88.2	98.1	96.9	96.5	98.4	84.8	94.9	82.7	98.5	92.8	93.4
ML-GCN [8]	99.5	98.5	98.6	98.1	80.8	94.6	97.2	98.2	82.3	95.7	86.4	98.2	98.4	96.7	99.0	84.7	96.7	84.3	98.9	93.7	94.0
GAT [44]	99.8	96.9	99.2	97.9	80.1	96.4	97.5	98.6	81.5	97.1	88.6	98.4	99.0	97.6	98.9	85.1	98.6	87.4	98.8	95.5	94.6
MGTN [33]	99.5	98.3	98.8	97.7	86.3	97.2	98.2	99.0	83.0	96.7	90.4	97.5	98.9	97.6	99.1	87.5	97.9	84.6	98.6	95.2	95.1
GATN	99.8	98.9	99.1	98.8	89.5	97.6	97.6	99.3	87.3	98.4	90.7	99.1	99.2	98.5	99.1	88.7	98.4	90.8	99.1	96.3	96.3

Table 2. Comparison with state-of-the-art models in MS-COCO

Methods	mAP	CP	CR	CF1	OP	OR	OF1
SRN [60]	77.1	81.6	65.4	71.2	82.7	69.9	75.8
ResNet-101 [17]	77.3	80.2	66.7	72.8	83.9	70.8	76.8
GAT [44]	81.6	82.6	71.6	76.7	84.9	75.1	79.7
A-GCN [22]	83.1	84.7	72.3	78.0	85.6	75.5	80.3
ML-GCN [8]	83.4	83.0	73.7	78.0	82.9	76.6	79.6
KSSNET [49]	83.7	84.6	73.2	77.2	87.8	76.2	81.5
KGGR [6]	84.3	85.6	72.7	78.6	87.1	75.6	80.9
C-Tran [20]	85.1	86.3	74.3	79.9	87.7	76.5	81.7
MGTN [33]	87.4	86.2	79.5	82.7	87.0	81.3	84.1
ML-Decoder [39]	88.8	51.7	94.1	66.7	47.3	95.8	63.3
GATN	89.3	89.1	79.9	84.3	89.6	82.0	85.7

4 EXPERIMENTS

In this section, we firstly introduce datasets and compare our method with state-of-the-art models. Then we conduct experiments with different CNN feature extractors to study the node representation learning ability of our proposed method. Subsequently, we visualize the generated correlation matrix by building a relation graph, which vividly shows the difference between our generated matrix and co-occurrence matrix. Then, we study the adaptation ability of our proposed graph attention transformer layer. Finally, the ablation experiments aim to explore the importance of each module.

4.1 Datasets and Evaluation Metrics

We evaluate our method on three datasets: PASCAL VOC2007 [12], MS-COCO [24] and NUS-WIDE [9]. VOC2007 has 9,963 images in 20 classes, and each image is assigned with one or multiple labels. We use 5,011 images for training and 4,952 images for testing. MS-COCO is a challenging multi-label classification dataset which consists of 80 classes. In this dataset, the training set includes 82,081 images, and the testing set includes 40,137 images. NUS-WIDE is a much larger dataset with 269,648 images. Following [25, 55], we remove the unreliable images and labels and finally get 208,347 images of 81 categories.

Table 3. Comparison with state-of-the-art models in NUS-WIDE

Methods	mAP	CP	CR	CF1	OP	OR	OF1
CNN-RNN [47]	-	40.5	30.4	34.7	49.9	61.7	55.2
Li et al. [21]	-	44.2	49.3	46.6	53.9	68.7	60.4
SR CNN-RNN [25]	-	55.7	50.2	52.8	70.6	71.4	71.0
Chen et al. [5]	-	59.4	50.7	54.7	69.0	71.4	70.2
ML-GCN [8]	53.3	59.4	46.1	51.9	74.7	68.4	71.4
GAT [44]	56.4	60.8	50.6	55.2	76.2	67.4	71.5
MGTN [33]	60.0	62.9	54.9	58.6	75.2	69.6	72.3
GATN	64.6	68.4	55.4	61.2	76.7	69.8	73.1

Evaluation metrics: We use average precision (AP), mean average precision (mAP), the average overall precision (OP), recall (OR), F1 (OF1) and average per-class precision (CP), recall (CR), F1 (CF1) to measure our models. In general, mean average precision (**mAP**), the average overall F1 (**OF1**), and average per-class F1 (**CF1**) are more important for measuring model’s performance.

4.2 Comparisons with state-of-the-arts

Our proposed model has two GCN layers, and the output dimensionality is 1,024 and 2,048, respectively. The hidden size in the attention layer is set the same as the number of classes. The threshold τ in Eq(2) and the k in Eq(7) are empirically set as 0.2 and 2. We employ ResNeXt-101 32X16d as backbone [52] for image feature extraction with a semi-weakly supervised pre-trained model on ImageNet [53]. For a fair comparison with the previous method, we also use GloVe [34] model as our label embeddings. The whole model is trained with SGD with momentum set as 0.9 and learning rate set as 0.03, which has a 0.1 weight decay. The model is trained for 50 epochs with batch size 16.

For VOC2007 dataset, we compare our model with following state-of-the-art models: CNN-RNN [47], ResNet-101 [17], FeV+LV [54], HCP [51], RNN-Attention [50], Atten-Reinforce [7], VGG [40], DLDL [14], ML-GCN [8], GAT [44], and MGTN [33]. As shown in Table 1, our proposed model reach optimal performance. GATN outperform ResNet-101 6.4%, VGG 3.5% in mAP, while ResNet-101 and VGG are considered mainstream deep CNN models. For RNN-based methods, GATN shows 12.3% promotion of mAP over CNN-RNN, and it also holds a 4.4% and 4.3% higher mAP over two methods RNN-Attention and Atten-Reinforce, which also include attention mechanisms. In comparison, GNN-based methods have a better performance than RNN-based methods. Three GNN-based methods, ML-GCN, GAT, and MGTN, all have more than 94% mAP. For ML-GCN, GATN has a 2.3% improvement in mAP, which verifies the effectiveness of our proposed graph attention transformer layer to enhance relational learning. GAT is also based on the attention mechanism, which gives neighbors different weights. Different from GAT, our method try to find more accurate neighbors to establish a more accurate relation, which has a 1.7% promotion over GAT. For MGTN, our method also shows a clear advantage of 1.2% mAP over it. Besides, specific to each category, GATN achieves state-of-the-art performance on 16 categories in AP. Except for *car*, *bird*, *sheep*, and *train*, our model all reach the current best AP. Especially for *bottle*, *chair* and *sofa*, our method can exceed the previous state-of-the-art result by more than 3% AP.

In Table 2, we compare our model with following state-of-the-art models in MS-COCO dataset: SRN [60], ResNet-101 [17], KGGR [6], ML-GCN [8], A-GCN [22], KSSNET [49], C-Tran [20], GAT [44], MGTN [33], and ML-Decoder [39]. We can see that GATN can outperform other methods in almost all metrics. It has a 12.2% and 12.0% higher mAP, 13.1% and 11.5% higher CF1, 9.9% and 8.9% higher OF1 than SRN and ResNet-101, respectively.

Table 4. Comparison under different backbones

Backbone	model	mAP	CP	CR	CF1	OP	OR	OF1	Flops [G]
ResNet101	ML-GCN	83.0	85.1	72.0	78.0	85.8	75.4	80.3	31.37
	MGTN	83.1	84.0	72.8	78.0	85.4	75.8	80.3	62.74
	GATN	82.9	83.9	71.8	77.4	86.3	75.4	80.4	31.37
ResNeXt-50 32x4d SWSL	ML-GCN	83.4	83.0	73.7	78.0	82.9	76.6	79.6	17.06
	MGTN	86.9	86.3	77.4	81.6	87.5	79.4	83.2	34.12
	GATN	87.1	87.0	77.3	81.8	88.1	79.4	83.5	17.06
ResNeXt-101 32x4d SWSL	ML-GCN	79.4	79.8	69.3	74.2	81.2	74.4	77.7	32.08
	MGTN	87.4	85.1	78.6	81.7	84.9	81.0	82.9	64.17
	GATN	87.9	87.6	78.7	82.9	88.0	80.9	84.3	32.09
ResNeXt-101 32x8d SWSL	ML-GCN	80.2	80.2	70.1	74.8	82.6	74.5	78.4	65.94
	MGTN	87.0	85.8	78.4	81.9	86.7	80.6	83.5	131.88
	GATN	88.0	87.6	79.0	83.1	88.8	81.9	85.2	65.94
ResNeXt-101 32x16d SWSL	ML-GCN	80.9	79.6	72.0	75.6	82.6	75.9	79.1	145.1
	MGTN	87.4	86.2	79.5	82.7	87.0	81.3	84.1	290.19
	GATN	89.3	89.1	79.9	84.3	89.6	82.0	85.7	145.1

Compared with GNN-based methods, GATN shows a clear margin over GAT of 7.7%, 7.6%, and 6.0% in mAP, CF1, and OF1. GATN can also exceed ML-GCN 5.9%, 6.3%, 6.1% in mAP, CF1, OF1, and show an improvement of 1.9%, 1.6%, 1.6% in mAP, CF1, OF1 over MGTM, while the latter is regarded as the most powerful model currently. Moreover, C-Tran is a transformer-based method, and GATN also shows a significant advantage over it, which shows that applying the self-attention mechanism to graphs is a more effective way. We also compared our method with ML-Decoder [39] by using the same CNN backbone, and our method holds a 0.5% advantage over ML-Decoder, which further verifies the efficiency of the graph attention transformer network. In general, GATN achieves state-of-the-art performance in five metrics, including mAP, CP, CF1, OP, and OF1.

As shown in Table 3, we compare GATN with the competitive models including CNN-RNN [47], Li et al. [21], SR CNN-RNN [25], Chen et al. [5], ML-GCN [8], GAT [44], and MGTM [33]. Owing to a large amount of data in NUS-WIDE, all models performs worse than the former two datasets. In general, GATN holds a clear advantages over other competitors. It has a 4.6%, 2.6% and 0.8% higher mAP, CF1 and OF1 than the second-best MGTM [33]. The fact that GATN does not achieve optimal results on all metrics can be attributed to the choice of threshold. The performance of overall precision (OP), recall (OR), average per-class precision (CP), and recall (CR) are highly correlated with the selection of the threshold. The higher threshold will lead to a higher precision and a lower recall, which means the performance of OP and OR (CP and CR) is a trade-off. Although GATN are inferior to some models in CR and OR, it performs better on mAP, OF1, and CF1, verifying the overall superiority of GATN.

Furthermore, the comparison experiments with graph-based and high-performance state-of-the-art methods, including GAT [44], MGTM [33], and ML-Decoder [39] are under the same backbone ResNeXt-101 32x16d SWSL [52]. For other methods, we refer the best results from the original papers due to different feature dimensions and different frameworks (e.g., CNN-RNN [5, 25, 47], CNN-Transformer [20]). In the following section, we further analyse the effect of different backbones.

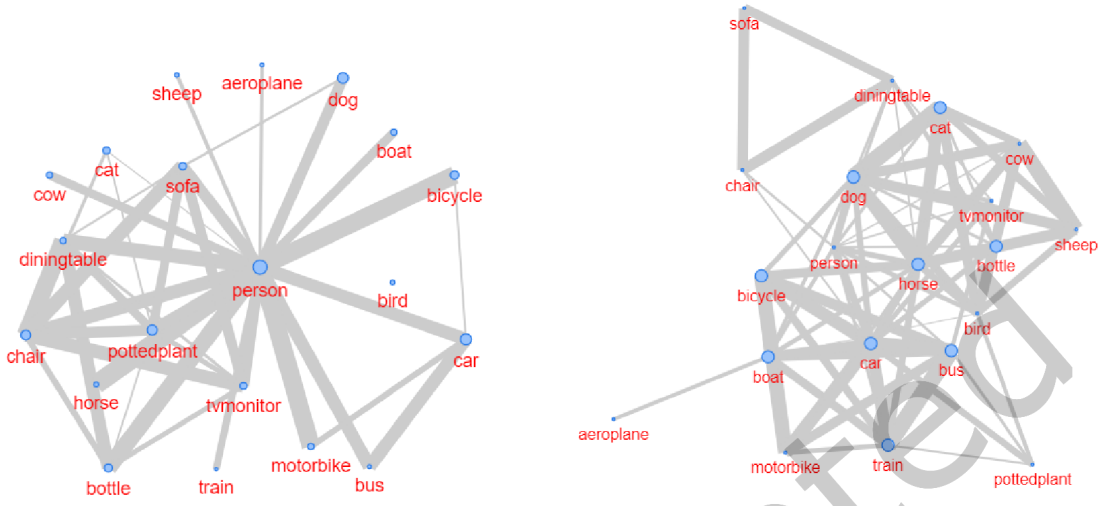


Fig. 3. Visualization of the adjacency matrix on the VOC2007 dataset. Each blue circle is a label node whose size is determined by the sum of its relation values. The gray line represents the relationship, and the thicker line means the stronger relationship. The left picture is the co-occurrence matrix and the right picture is our generated correlation matrix.

4.3 Model Representation Ability Experiments

GNN-based methods rely on the extraction of image features and label correlation features. To further explore the importance of label correlation features and the label correlation representation ability of different GNN models, we compare our model with two GNN-based models (ML-GCN and MGTN) of excellent performance under different backbones.

As shown in Table 4, for ResNet-101, these three models have similar performance, including mAP around 83.0, CF1 around 78.0, and OF1 around 80.3, which shows that the performance of the model at this time is limited by the image extractor. With the increase of the representation ability of image feature backbone models, ML-GCN has a slight improvement with ResNeXt-50 32x4d SWSL of 0.4% mAP. At this time, MGTN and GATN both have a greater improvement. MGTN has 3.8%, 3.6% and 2.9% improvements on mAP, OF1, and CF1, while GATN gains 4.2%, 4.4% and 3.1% in these three metrics. For this backbone, ML-GCN is limited by its label correlation feature learning branch, which also illustrates that the core of the multi-label task is to learn more powerful relational representations. On the contrary, MGTN and GATN both get better results with the increase of image feature expression ability, which can be attributed to their powerful label correlation features.

For ResNeXt-101 models, ML-GCN has some decline in all metrics, so we will not discuss it later. On the other hand, MGTN and GATN both get better results by using ResNeXt-101. However, GATN continues to improve its performance as the backbone model's ability become more powerful, it has a 0.8% increase of mAP, 1.1% increase of CF1, 0.8% increase of OF1 with ResNeXt-101 32x4d SWSL compared with ResNeXt-50, and it also obtains 2.2% mAP increment, 2.5% CF1 increment and 2.2 % OF1 increment with ResNeXt-101 32x16d SWSL compared with ResNeXt-50. Different from GATN, MGTN has little difference in performance on different ResNeXt-101, it only has a 1.0% and 1.2% rise in CF1 and OF1 when comparing its result on ResNeXt-101 32x4d SWSL and ResNeXt-101

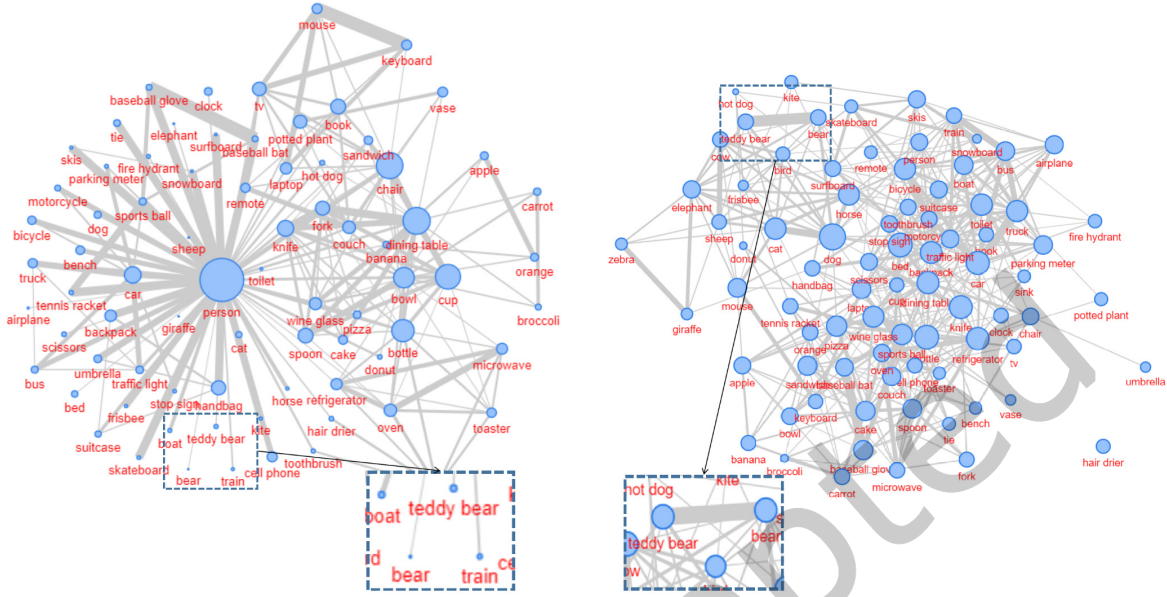


Fig. 4. Visualization of the adjacency matrix on the MS-COCO dataset. Each blue circle is a label node whose size is determined by the sum of its relation values. The gray line represents the relationship, and the thicker line means the stronger relationship. The left picture is the co-occurrence matrix and the right picture is our generated correlation matrix. Specifically, we zoomed in *bear* and *teddy bear* for a comparison.

32x16d SWSL. These results demonstrate that GATN has a stronger representation ability in label correlation learning compared to MGTN.

We also compared the computational cost in the last column. Compared with ML-GCN, GATN does not increase much computational cost, which is reasonable since GATN only adds a small multi-head attention based transformer block after the adjacency matrix. The adjacency matrix $A \in \mathbb{R}^{n \times n}$ is only related to the number of categories n (in MS-COCO, n is 80), so the computational resources consumed by our proposed method are negligible compared to those required by the feature extractor. Moreover, the computation cost of MGTN is almost double that of the ML-GCN and GATN. These results can be attributed to their modular strategy of using two backbone models with different graph correlation matrices.

4.4 Visualization of the Generated Correlation Matrix

To further demonstrate the effect of our generated correlation matrix, we conduct a visualization experiment for the VOC2007 and MS-COCO datasets in this section.

As shown in Fig. 3 and Fig. 4, we compare our generated correlation matrix with the co-occurrence matrix. Firstly, we set node size as the sum of all its relations. Thus, the larger the node, the more the number of its relationship with other categories. Subsequently, the thickness of the edge is defined as the strength of the relationship, and the thick edge indicates a strong relationship. To avoid visual disturbance caused by too many edges, we filter the edges with a threshold.

For VOC2007, the center of co-occurrence topological graph is *person* since images containing people have a large proportion in this dataset. Therefore, models with a co-occurrence matrix will only focus on the relation

between *person* and other objects, thereby ignoring the potential relations between other categories. On the other hand, our proposed distance-based adjacency matrix is more reasonable by discovering adequate semantic relations. It shows some functional relations like *car*, *train* and *bus*. Besides, it does not miss the potential co-occurring relationship including *sofa*, *diningtable* and *chair*. Combined with Table 1, we can find that the categories with more improvement have complex relationships with other categories in the co-occurrence matrix, like bottle (5 edges), chair (6 edges), and sofa (6 edges). For some categories with simple relationships, our method has a limited promotion, e.g., sheep (1 edge), aeroplane(1 edge), train (1 edge), and bird (1 edge). These results further illustrate that our method improves performance by enhancing relational learning.

For MS-COCO, *person* also occupies a dominant position in the co-occurrence one due to its extremely high frequency in the training set. On the contrary, our generated correlation matrix has a more balanced relationship strength. Specifically, we can observe that *teddy bear* and *bear* have a strong relation in our generated matrix since they have a similar appearance. However, the two rarely appear simultaneously in natural images, which leads to the co-occurrence matrix being likely to ignore this relation. Moreover, our generated correlation matrix still captures the functional relation between *airplane*, *bus* and *train*, which is difficult to obtain in co-occurrence matrix since different transports rarely appear in a same image. For animals, our generated matrix find the relation between *elephant*, *zebra* and *giraffe*, which is hard for co-occurrence matrix to catch. Besides, some other interesting relations are also helpful for model training, like *kite* and *bird*, *skateboard* and *surfboard*, *knife* and *scissors*.

In general, the correlation matrix we generated can capture high-level semantic information, which is more helpful for the model to understand high-level semantic knowledge.

4.5 Adaptation Ability Study

To study the adaptation ability of our proposed graph attention transformer layer, we output the adjacency matrix before and after our graph attention transformer layer for comparison.

We randomly select 10 categories in the VOC2007 dataset and show results in Fig. 5. We can find the relation between *dog* and *chair* become stronger, which can be attributed to the VOC2007 dataset containing a large number of photos of dogs at home. Samely, the relation between *person* and *aeroplane* is lower than that between *person* and *bus*, which is reasonable since people usually take buses more often than airplanes. Due to the photos of airplanes and people in the VOC2007 dataset being relatively high, the relationship between *airplane* and *person* in the transformed matrix is becoming stronger. Moreover, *train* and *bus* are deeply related in semantic information because they are both vehicles. In this dataset, the two often appear in isolation, which makes the relationship between the *train* and *bus* in the transformed matrix weaker.

4.6 Ablation Studies

To verify the effectiveness of each module of our proposed method, we conduct ablation studies. Since our method is based on ML-GCN, we name the original ML-GCN as ML-GCN (CO-OCC), the version that only uses our generated correlation matrix as ML-GCN (CORR), the version that only uses graph attention transformer layer as GATN (CO-OCC), and our final version is named GATN (CORR). All experiments are conducted in the MS-COCO dataset.

As shown in Table 5, both ML-GCN (CORR) and GATN (CO-OCC) show advantages over ML-GCN (CO-OCC). ML-GCN (CORR) has a 0.9%, 1.1% and 1.6% improvements on mAP, CF1 and OF1, while GATN (CO-OCC) has 5.2%, 5.2% and 3.6% improvements on mAP, CF1 and OF1. Generally, our proposed graph attention transformer layer is more helpful for performance, which is reasonable since the generated matrix indeed changes the initial graph adjacency matrix without changing the adaptation ability of the model. Moreover, the GATN (CORR)

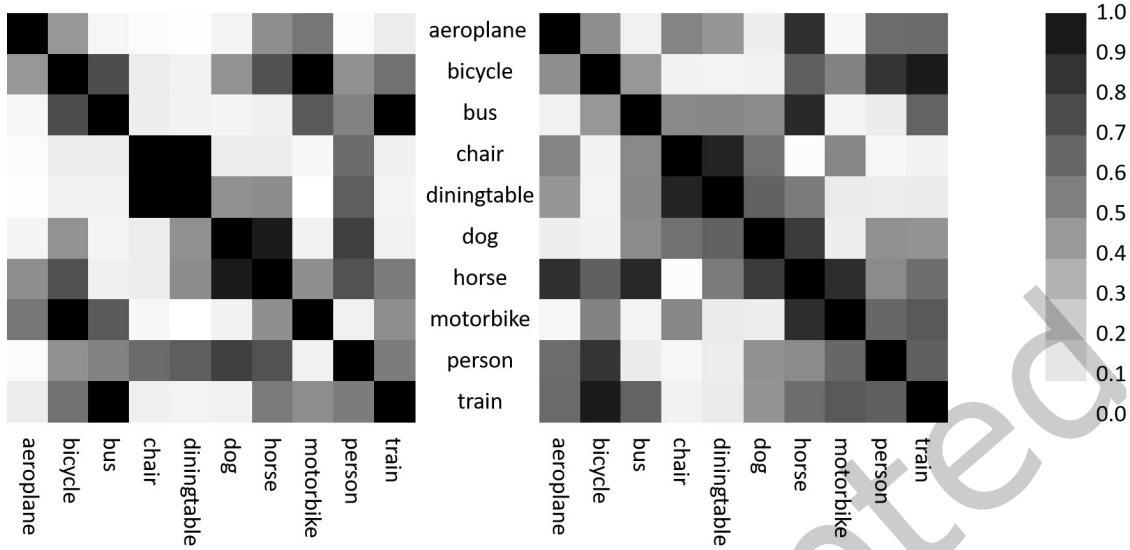


Fig. 5. The left picture is the initialized correlation matrix generated by the distance-based method, and the right is the transformed matrix after the graph attention transformer layer. The darker the color, the stronger the relationship between the two categories.

Table 5. Ablation studies

Methods	mAP	CP	CR	CF1	OP	OR	OF1
ML-GCN (CO-OCC)	83.4	83.0	73.7	78.0	82.9	76.6	79.6
ML-GCN (CORR)	84.3	86.9	72.6	79.1	87.4	75.9	81.2
GATN (CO-OCC)	88.6	85.6	80.9	83.2	84.0	82.4	83.2
GATN (CORR)	89.3	89.1	79.9	84.3	89.6	82.0	85.7

also has a 0.7%, 1.1% and 2.5% in mAP, CF1 and OF1 compared to GATN (CO-OCC), further explaining that the generated correlation matrix is also helpful to the result.

5 CONCLUSION

In this paper, we propose Graph Attention Transformer Networks to complete multi-label image classification. Our proposed method generates a correlation matrix through a distance-based algorithm and exploits a novel structure including several self-attention branches to mine the multi-hop paths in the correlation matrix. By transforming the correlation matrix into multiple sub-graphs, we can improve the expressive ability of the label relation, thereby improving the accuracy of multi-label classification tasks. A large number of experiments on three datasets prove the effectiveness of our method. We believe that our method has powerful abilities. It only needs to input the node representation in the network, and then it can extract the different transformations of relationships among nodes. In the future, we will continue to explore the performance of this network in other tasks, such as node classification and relationship extraction.

REFERENCES

- [1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *ICLR 2015 : International Conference on Learning Representations 2015*.
- [2] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. 2014. Spectral Networks and Locally Connected Networks on Graphs. In *ICLR 2014 : International Conference on Learning Representations (ICLR) 2014*.
- [3] Mark Chen, Alec Radford, Rewon Child, Jeffrey K Wu, Heewoo Jun, David Luan, and Ilya Sutskever. 2020. Generative Pretraining From Pixels. In *ICML 2020: 37th International Conference on Machine Learning*, Vol. 1. 1691–1703.
- [4] Shikai Chen, Jianfeng Wang, Yuedong Chen, Zhongchao Shi, Xin Geng, and Yong Rui. 2020. Label Distribution Learning on Auxiliary Label Space Graphs for Facial Expression Recognition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 13984–13993.
- [5] Shang-Fu Chen, Yi-Chen Chen, Chih-Kuan Yeh, and Yu-Chiang Frank Wang. 2017. Order-Free RNN with Visual Attention for Multi-Label Classification. In *AAAI*. 6714–6721.
- [6] Tianshui Chen, Liang Lin, Xiaolu Hui, Riquan Chen, and Hefeng Wu. 2020. Knowledge-Guided Multi-Label Few-Shot Learning for General Image Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020), 1–1.
- [7] Tianshui Chen, Zhouxia Wang, Guanbin Li, and Liang Lin. 2017. Recurrent Attentional Reinforcement Learning for Multi-label Image Recognition. In *AAAI*. 6730–6737.
- [8] Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, and Yanwen Guo. 2019. Multi-Label Recognition With Graph Convolutional Networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 5177–5186.
- [9] Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yantao Zheng. 2009. NUS-WIDE: a real-world web image database from National University of Singapore. In *Proceedings of the ACM International Conference on Image and Video Retrieval*. 48.
- [10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina N. Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 4171–4186.
- [11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR 2021: The Ninth International Conference on Learning Representations*.
- [12] Mark Everingham, Luc Gool, Christopher K. Williams, John Winn, and Andrew Zisserman. 2010. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision* 88, 2 (2010), 303–338.
- [13] Andrea Galassi, Marco Lippi, and Paolo Torrioni. 2020. Attention in natural language processing. *IEEE Transactions on Neural Networks and Learning Systems* 32, 10 (2020), 4291–4308.
- [14] Bin-Bin Gao, Chao Xing, Chen-Wei Xie, Jianxin Wu, and Xin Geng. 2017. Deep Label Distribution Learning With Label Ambiguity. *IEEE Transactions on Image Processing* 26, 6 (2017), 2825–2838.
- [15] Zongyuan Ge, Dwarikanath Mahapatra, Suman Sedai, Rahil Garnavi, and Rajib Chakravorty. 2018. Chest X-rays Classification: A Multi-Label and Fine-Grained Problem. *arXiv preprint arXiv:1807.07247* (2018).
- [16] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable Feature Learning for Networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Vol. 2016. 855–864.
- [17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 770–778.
- [18] Qinghua Huang, Bisheng Chen, Jingdong Wang, and Tao Mei. 2014. Personalized video recommendation through graph propagation. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 10, 4 (2014), 1–17.
- [19] Thomas N. Kipf and Max Welling. 2016. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR (Poster)*.
- [20] Jack Lanchantin, Tianlu Wang, Vicente Ordonez, and Yanjun Qi. 2020. General Multi-label Image Classification with Transformers. *arXiv preprint arXiv:2011.14027* (2020).
- [21] Liang Li, Shuhui Wang, Shuqiang Jiang, and Qingming Huang. 2018. Attentive Recurrent Neural Network for Weak-supervised Multi-label Image Classification. In *Proceedings of the 26th ACM international conference on Multimedia*. 1092–1100.
- [22] Qing Li, Xiaojiang Peng, Yu Qiao, and Qiang Peng. 2019. Learning Category Correlations for Multi-label Image Recognition with Graph Networks. *arXiv preprint arXiv:1909.13005* (2019).
- [23] Yining Li, Chen Huang, Chen Change Loy, and Xiaoou Tang. 2016. Human Attribute Recognition by Deep Hierarchical Contexts. In *European Conference on Computer Vision*. 684–700.
- [24] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. In *European Conference on Computer Vision*. 740–755.
- [25] Feng Liu, Tao Xiang, Timothy M. Hospedales, Wankou Yang, and Changyin Sun. 2017. Semantic Regularisation for Recurrent Image Annotation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 4160–4168.

- [26] Shilong Liu, Lei Zhang, Xiao Yang, Hang Su, and Jun Zhu. 2021. Query2label: A simple transformer way to multi-label classification. *arXiv preprint arXiv:2107.10834* (2021).
- [27] Weiwei Liu and Ivor W. Tsang. 2015. On the optimality of classifier chain for multi-label classification. In *NIPS'15 Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, Vol. 28. 712–720.
- [28] Yang Liu, Yao Zhang, Yixin Wang, Feng Hou, Jin Yuan, Jiang Tian, Yang Zhang, Zhongchao Shi, Jianping Fan, and Zhiqiang He. 2021. A survey of visual transformers. *arXiv preprint arXiv:2111.06091* (2021).
- [29] Xin Man, Deqiang Ouyang, Xiangpeng Li, Jingkuan Song, and Jie Shao. 2022. Scenario-Aware Recurrent Transformer for Goal-Directed Video Captioning. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 18, 4 (2022), 1–17.
- [30] Nicola Messina, Giuseppe Amato, Andrea Esuli, Fabrizio Falchi, Claudio Gennaro, and Stéphane Marchand-Maillet. 2021. Fine-grained visual textual alignment for cross-modal retrieval using transformer encoders. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 17, 4 (2021), 1–23.
- [31] Jinseok Nam, Young-Bum Kim, Eneldo Loza Mencia, Sunghyun Park, Ruhi Sarikaya, and Johannes Fürnkranz. 2019. Learning Context-dependent Label Permutations for Multi-label Classification. In *International Conference on Machine Learning*. 4733–4742.
- [32] Nipun D. Nath, Theodora Chaspari, and Amir H. Behzadan. 2019. Single- and multi-label classification of construction objects using deep transfer learning methods. *Journal of Information Technology in Construction* 24, 28 (2019), 511–526.
- [33] Hoang D. Nguyen, Xuan-Son Vu, and Duc-Trong Le. 2021. Modular Graph Transformer Networks for Multi-Label Image Classification. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI '21)*. AAAI.
- [34] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 1532–1543.
- [35] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. DeepWalk: online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 701–710.
- [36] José Ramón Quevedo, Oscar Luaces, and Antonio Bahamonde. 2012. Multilabel classifiers with a probabilistic thresholding strategy. *Pattern Recognition* 45, 2 (2012), 876–883.
- [37] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. (2018).
- [38] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. 2011. Classifier chains for multi-label classification. *Machine Learning* 85, 3 (2011), 333–359.
- [39] Tal Ridnik, Gilad Sharir, Avi Ben-Cohen, Emanuel Ben-Baruch, and Asaf Noy. 2021. ML-Decoder: Scalable and Versatile Classification Head. *arXiv preprint arXiv:2111.12933* (2021).
- [40] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR 2015 : International Conference on Learning Representations 2015*.
- [41] Yaguang Song, Xiaoshan Yang, and Changsheng Xu. 2022. Self-supervised Calorie-aware Heterogeneous Graph Networks for Food Recommendation. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* (2022).
- [42] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. LINE: Large-scale Information Network Embedding. In *Proceedings of the 24th International Conference on World Wide Web*. 1067–1077.
- [43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Vol. 30. 5998–6008.
- [44] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.
- [45] Xuan-Son Vu, Duc-Trong Le, Christoffer Edlund, Lili Jiang, and Hoang D. Nguyen. 2020. Privacy-Preserving Visual Content Tagging using Graph Transformer Networks. In *Proceedings of the 28th ACM International Conference on Multimedia*. 2299–2307.
- [46] Haidong Wang, Xuan He, Zhiyong Li*, Jin Yuan*, and Shutao Li. 2022. JDAN: Joint Detection and Association Network for Real-Time Online Multi-Object Tracking. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* (2022).
- [47] Jiang Wang, Yi Yang, Junhua Mao, Zhiheng Huang, Chang Huang, and Wei Xu. 2016. CNN-RNN: A Unified Framework for Multi-label Image Classification. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2285–2294.
- [48] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. 2019. Heterogeneous graph attention network. In *The world wide web conference*. 2022–2032.
- [49] Ya Wang, Dongliang He, Fu Li, Xiang Long, Zhichao Zhou, Jinwen Ma, and Shilei Wen. 2020. Multi-Label Classification with Label Graph Superimposing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 12265–12272.
- [50] Zhouxia Wang, Tianshui Chen, Guanbin Li, Ruijia Xu, and Liang Lin. 2017. Multi-label Image Recognition by Recurrently Discovering Attentional Regions. In *2017 IEEE International Conference on Computer Vision (ICCV)*. 464–472.
- [51] Yunchao Wei, Wei Xia, Min Lin, Junshi Huang, Bingbing Ni, Jian Dong, Yao Zhao, and Shuicheng Yan. 2016. HCP: A Flexible CNN Framework for Multi-Label Image Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38, 9 (2016), 1901–1907.
- [52] Saining Xie, Ross Girshick, Piotr Dollar, Zhuowen Tu, and Kaiming He. 2017. Aggregated Residual Transformations for Deep Neural Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 5987–5995.

- [53] I. Zeki Yalniz, Hervé Jégou, Kan Chen, Manohar Paluri, and Dhruv Mahajan. 2019. Billion-scale semi-supervised learning for image classification. *arXiv preprint arXiv:1905.00546* (2019).
- [54] Hao Yang, Joey Tianyi Zhou, Yu Zhang, Bin-Bin Gao, Jianxin Wu, and Jianfei Cai. 2016. Exploit Bounding Box Annotations for Multi-Label Object Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 280–288.
- [55] Vacit Oguz Yazici, Abel Gonzalez-Garcia, Arnau Ramisa, Bartłomiej Twardowski, and Joost van de Weijer. 2020. Orderless Recurrent Models for Multi-Label Classification. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 13440–13449.
- [56] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J. Kim. 2019. Graph Transformer Networks. In *33rd Annual Conference on Neural Information Processing Systems, NeurIPS 2019*, Vol. 32. 11960–11970.
- [57] Yong Zhang, Yingwei Pan, Ting Yao, Rui Huang, Tao Mei, and Chang-Wen Chen. 2022. Boosting Scene Graph Generation with Visual Relation Saliency. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* (2022).
- [58] Kaili Zhao, Wen-Sheng Chu, and Honggang Zhang. 2016. Deep Region and Multi-label Learning for Facial Action Unit Detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 3391–3399.
- [59] Wei Zhou, Zhiwu Xia, Peng Dou, Tao Su, and Haifeng Hu. 2022. Double Attention based on Graph Attention Network for Image Multi-Label Classification. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* (2022).
- [60] Feng Zhu, Hongsheng Li, Wanli Ouyang, Nenghai Yu, and Xiaogang Wang. 2017. Learning Spatial Regularization with Image-Level Supervisions for Multi-label Image Classification. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2027–2036.
- [61] Ni Zhuang, Yan Yan, Si Chen, Hanzhi Wang, and Chunhua Shen. 2018. Multi-label learning based deep transfer neural network for facial attribute classification. *Pattern Recognition* 80 (2018), 225–240.