Knowledge-Guided Multi-Label Few-Shot Learning for General Image Recognition

Tianshui Chen, Liang Lin, Riquan Chen, Xiaolu Hui, and Hefeng Wu

Abstract—Recognizing multiple labels of an image is a practical yet challenging task, and remarkable progress has been achieved by searching for semantic regions and exploiting label dependencies. However, current works utilize RNN/LSTM to implicitly capture sequential region/label dependencies, which cannot fully explore mutual interactions among the semantic regions/labels and do not explicitly integrate label co-occurrences. In addition, these works require large amounts of training samples for each category, and they are unable to generalize to novel categories with limited samples. To address these issues, we propose a knowledge-guided graph routing (KGGR) framework, which unifies prior knowledge of statistical label correlations with deep neural networks. The framework exploits prior knowledge to guide adaptive information propagation among different categories to facilitate multi-label analysis and reduce the dependency of training samples. Specifically, it first builds a structured knowledge graph to correlate different labels based on statistical label co-occurrence. Then, it introduces the label semantics to guide learning semantic-specific features to initialize the graph, and it exploits a graph propagation network to explore graph node interactions, enabling learning contextualized image feature representations. Moreover, we initialize each graph node with the classifier weights for the corresponding label and apply another propagation network to transfer node messages through the graph. In this way, it can facilitate exploiting the information of correlated labels to help train better classifiers, especially for labels with limited training samples. We conduct extensive experiments on the traditional multi-label image recognition (MLR) and multi-label few-shot learning (ML-FSL) tasks and show that our KGGR framework outperforms the current state-of-the-art methods by sizable margins on the public benchmarks.

Index Terms—Image Recognition, Multi-Label Learning, Few-Shot Learning, Knowledge Graph, Graph Reasoning

◆

1 INTRODUCTION

REAL-WORLD images generally contain objects belonging to multiple diverse categories; thus, recognizing the multiple object categories in images is a more fundamental and practical task compared with single-object image recognition in computer vision and multimedia fields. Recently, researchers have concentrated on developing a series of algorithms for multi-label image analysis [1], [2], [3], [4], [5] that underpin many critical applications, such as content-based image retrieval and recommendation systems [6], [7]. Despite these achievements, identifying the existence of multiple semantic categories requires not only mining diverse semantic object regions but also capturing the interplay among these regions and their semantics, making multi-label image analysis a challenging and unsolved task.

The current MLR approaches mainly employ object localization techniques [2], [8], [9] or adopt visual attentional mechanisms [10], [11] to adaptively discover regions with meaningful semantic categories. However, object localization techniques [12], [13], [14], [15] need to search numerous category-agnostic and superfluous region proposals, which

cannot be unified into deep neural networks for end-to-end optimization. Moreover, visual attentional mechanisms cannot accurately locate semantic object regions due to a lack of explicit supervision and guidance. More recent works have further introduced RNN/LSTM models [16], [17], [18] to identify the contextual dependencies among located regions and thus implicitly capture label dependencies. However, these algorithms merely model sequential region/label dependencies; they cannot fully exploit these properties because a direct correlation or dependency exists between each region/label pair. In addition, they do not explicitly integrate statistical category correlations that provide direct and key guidance to aid multi-label image analysis.

The current methods depend on deep convolution networks [19], [20] for learning image features; such networks require large numbers of training samples for each category and are unable to generalize to novel categories with limited samples. To address this issue, researchers have recently developed a series of few-shot learning algorithms that can learn novel categories after being trained on a set of base categories with sufficient training samples. These algorithms utilize the meta-learning paradigm [21], [22] to help learn the novel categories by distilling the knowledge of the categories with sufficient training samples or use sample synthesis technologies [23] to generate more diverse samples for the novel categories. Although these techniques have achieved impressive progress, they focus mainly on single-label scenarios rather than on more general multi-label cases.

Objects in visual scenes commonly have strong correlations. For example, desks tend to co-occur with chairs, while

computers usually co-exist with keyboards. These correlations can provide extra guidance for capturing the interplay among different categories and thus facilitate multi-label analysis, especially for complex and few-shot scenarios. In this work, we show that semantic correlations can be explicitly represented by a structured knowledge graph and that such interplay can be effectively captured by information propagation through the graph. To this end, we propose a knowledge-guided graph routing (KGGR) framework that captures the interplay of features from different semantic regions to help learn more powerful contextualized image representation and guides the propagation of semantic information through different categories to help train the classifiers.

The framework builds on two graph propagation networks that perform message propagation on feature and semantic spaces. Specifically, it first constructs a graph based on statistical label co-occurrences to correlate different categories. For the feature space, we designed a semantically guided attention module that applies category semantics to guide learning category-related image features that focus more on the corresponding semantic regions. By initializing the graph node with the feature vector of the corresponding category, we introduce a graph propagation network to propagate features through the graph to capture feature interactions and learn contextualized feature representations. Regarding the semantic space, we treat each node as the classifier weight of each category and utilize another graph propagation network to transfer node messages throughout the graph, allowing each category to derive information from its correlated categories to improve classifier training.

A preliminary version of this work was presented as a conference paper [24]. In this version, we inherit the idea of explicitly integrating statistical category correlations with deep graph propagation networks and strengthen the framework from several aspects as follows. First, we propagate information in both the feature and semantic spaces, which guides the learning of more powerful contextualized features and simultaneously regularizes learning classifier weights. Second, we generalize the proposed framework to the more challenging multi-label few-shot learning task and demonstrate its superiority on this task. Finally, we conduct more extensive experiments and analyses on several widely used benchmarks and demonstrate the effectiveness of the proposed framework while verifying the contribution of each component.

In summary, the contributions of this work can be summarized as follows: 1) We propose a novel knowledge-guided graph routing (KGGR) framework that explores category interplay in both feature and semantic spaces under the explicit guidance of statistical category co-occurrence correlations. This approach can help learn more powerful contextualized features and simultaneously regularize learning classifier weights. 2) We propose a simple yet effective semantically guided attention mechanism that exploits category semantics to learn semantic-aware features that focus more on semantic regions of corresponding categories. 3) We apply the KGGR framework to address both multi-label image recognition and multi-label few-shot learning tasks and conduct experiments on various benchmarks, including PASCAL VOC 2007 & 2012 [25], Microsoft-COCO

[26], and Visual Genome with larger-scale categories [27], and demonstrate that our framework exhibits substantial performance improvements on both tasks.

2 RELATED WORKS

In this section, we review the related works following three main research streams: multi-label image recognition, few-shot learning, and knowledge representation learning.

2.1 Multi-label Image Recognition

Recent progress on multi-label image classification relies on a combination of object localization and deep learning techniques [2], [8]. Generally, such works introduced object proposals [13] that were assumed to include all possible foreground objects in the image; then, they aggregated the features extracted from all these proposals to incorporate local information. Although these methods have achieved notable performance improvement, the region candidate localization step usually incurred redundant computation costs and prevented the models from being applied to end-to-end training approaches with deep neural networks. Zhang et al. [28] further utilized a learning-based region proposal network and integrated it with deep neural networks. Although this method could be jointly optimized, it required additional bounding box annotations to train the proposal generation component. To solve this issue, some other works [10], [11] resorted to attention mechanisms to locate the informative regions. These methods could be trained with image-level annotations in an end-to-end manner. For example, Wang et al. [11] introduced a spatial transformer to adaptively search for semantic regions and then aggregated the features from these regions to identify multiple labels. However, due to a lack of supervision and guidance, these methods could merely locate the regions roughly.

Modeling label dependencies can help capture label co-occurrence, which is also a key aspect of multi-label recognition. To achieve this, a series of works have introduced graphic models such as the conditional random field [29], dependency network [30], or co-occurrence matrix [31] to capture pairwise label correlations. Recently, Wang et al. [17] formulated a CNN-RNN framework that implicitly utilized semantic redundancy and co-occurrence dependency to facilitate effective multi-label classification. Some works [18], [28] further capitalized on the proposal generation/visual attention mechanism to search local discriminative regions and used RNN/LSTM [16] models to explicitly model label dependencies. For example, Chen et al. [18] proposed a recurrent attention reinforcement learning framework to iteratively discover a sequence of attentional and informative regions to model long-term dependencies among them to help capture semantic label co-occurrences. Furthermore, Chen et al. [32] directly mapped the GloVe vector to the corresponding object classifier via a graph convolutional network. Different from all these methods, we correlate all label pairs in the form of a structured graph and introduce two parallel graph neural networks to simultaneously explore interactions among category-specific features to learn contextualized feature representation and to explore

information propagation from correlated categories to learn classifier weights. Besides, the semantically guided attention mechanism that introduces category semantic to guide learning category-specific features is also original to current works.

2.2 Few-shot learning

Few-shot learning aims to understand novel concepts from only a few examples, mimicking the human cognitive abilities. Learning robust feature representations and training classifiers to recognize novel categories is the essential image recognition issue in the few-shot scenario. To address this task, the existing works are mainly characterized by three types of helpful techniques: metric learning, meta-learning and data synthesis, which are often combined. Metric learning-based methods [21], [22], [33], [34] are intended to learn an appropriate embedding space in which same-category samples are close while different-category samples are distant. A matching network using an attention mechanism and memory unit was introduced in [33] to compare the test and support samples. Snell et al. [34] proposed a prototypical network that adopts the mean of the embedded sample features in novel categories as a class prototype and then recognizes test samples using nearest-neighbor classifiers. In [21], a relational network was proposed to learn a transferable deep metric. Some recent works have further adaptively generated the classifier weights of novel categories from feature embedding [35], [36]. Meta-learning, also known as learning to learn, seeks to transfer some "meta knowledge" from previously learned tasks to achieve rapid learning of new tasks. Meta-learning is popular among few-shot learning approaches [21], [37], [38], [39]. Finn et al. [38] proposed a model-agnostic meta-learning approach to initialize the network well and make fine-tuning the model easier for novel categories with few samples. In [39], transferable optimization strategies are learned using an LSTM-based meta-learner to achieve effective model training in the few-shot regime. Data synthesis tries to generate new samples from the few given training samples to augment model training in the few-shot task, motivated by the human ability to generate new object examples from known primitives. In addition to simple data augmentation tricks such as horizontal flipping, scaling and positional shifts, more sophisticated approaches have been introduced [23], [40], [41], [42]. In [40], a delta-encoder model was designed that learned to extract transferable intraclass deformations and synthesize samples for novel categories. Alfassy et al. [42] proposed a label-set operations network to synthesize samples with multiple labels to address the new and challenging multi-label few-shot task and was the first to introduce a benchmark. In this work, we generalize our KGGR framework to this task and demonstrate its superior ability to learn robust features and classifiers from limited samples.

2.3 Knowledge Representation Learning

Deep neural networks have made impressive breakthroughs in recent years and are capable of learning powerful representations from raw training data [9], [19], [43], [44].

However, these models also have a bottleneck. One potential reason is that they lack the notable human capability of exploiting prior knowledge, which arises from knowledge accumulation and reasoning. Thus, the ability to incorporate prior knowledge into deep representation learning has become a new trend, and promising progress has been made in computer vision for tasks ranging from object detection/recognition [45], [46], [47], [48], [49] to visual navigation/reasoning [50], [51], [52], [53]. In [45], Marino et al. constructed a knowledge graph to correlate category relationships and introduced a graph search neural network to incorporate the knowledge graph into end-to-end representation learning for image classification. Tang et al. [47] incorporated object similarity knowledge from the visual and semantic domains during the transfer learning process to help improve semisupervised object detection. To address the zero-shot recognition task, Wang et al. [54] proposed building on a graph convolutional network to utilize both semantic embeddings and categorical relationships for classifier prediction. A hierarchical semantic embedding framework was presented in [48] to incorporate hierarchal category knowledge into a deep neural network to enhance fine-grained representation learning and recognition. In [51], an end-to-end trainable graph reasoning model was introduced to explicitly model the interplay of knowledge of people and contextual objects to perform reasoning regarding social relationships. Li et al. [55] introduced hierarchical category knowledge via semantic clustering [56] and used it to regularize the learning of transferable visual features for large-scale few-shot image classification. Yang et al. [53] studied incorporating semantic priors to address the semantic navigation task, in which the prior knowledge was fused into a deep reinforcement learning framework by graph convolutional networks. In [57], a knowledge integration network was designed to encode human and scene knowledge for action recognition. Inspired by these works, we propose a new knowledge-guided graph routing (KGGR) framework that incorporates prior knowledge to propagate both feature and semantic space information to effectively boost multi-label analysis performance.

3 KGGR FRAMEWORK

In this section, we provide an overall introduction to the proposed KGGR framework. The framework builds on a graph that encodes prior knowledge of category correlations. Then, it introduces two graph propagation networks that propagate information in both the feature and semantic spaces. For feature propagation, the framework first feeds the input image into a fully convolutional network to generate the feature maps and incorporates category semantics to guide learning semantic-specific feature vectors that focus more on the semantic regions for each category. Then, it further initializes the graph nodes with the corresponding feature vectors and applies a graph propagation network to explore feature interplay to learn contextualized features. For semantic propagation, each node is initialized with the classifier weights of the corresponding category. Then, another graph propagation network is learned to adaptively propagate node messages through the graph to explore node interactions and transfer the classifier's information of the
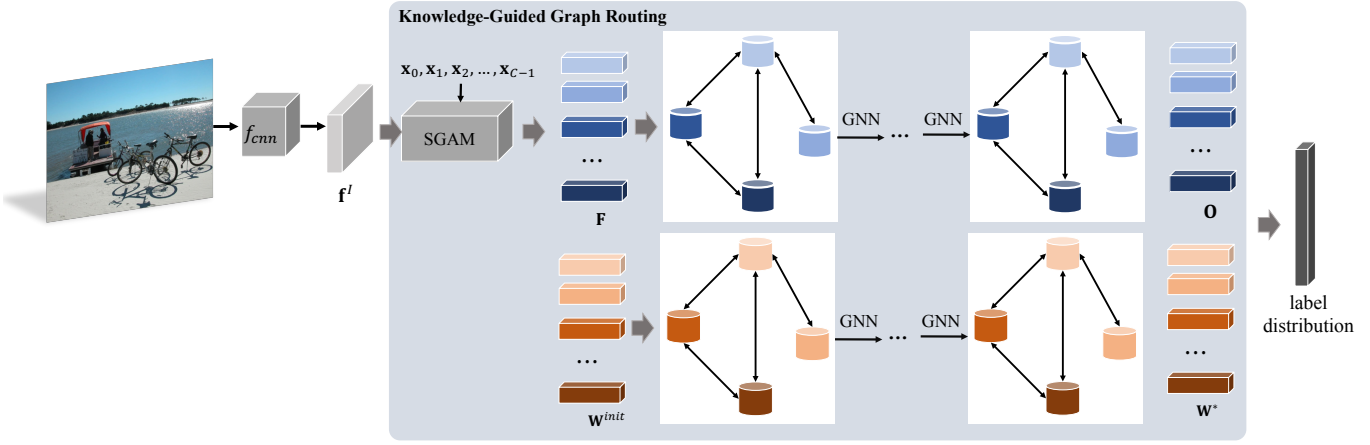
Fig. 1. Illustration of our knowledge-guided graph routing framework. It first uses a CNN to extract image features, and then introduces two graph propagation networks to transfer message on both feature and semantic spaces to learn contextualized features and simultaneously regularize learning classifier weights.

correlated categories to help train the classifiers. The overall pipeline of the proposed KGGR framework is presented in Figure 1.

3.1 Knowledge Graph Construction

We first introduce the knowledge graph $\mathcal{G} = \{\mathbf{V}, \mathbf{A}\}$, in which the nodes represent categories and the edges represent co-occurrences between corresponding categories. Specifically, suppose that the dataset covers C categories, $\mathbf{V}$ can be represented as $\{v_0, v_1, \ldots, v_{C-1}\}$ where element v_c denotes category c and $\mathbf{A}$ can be represented as $\{a_{00}, a_{01}, \ldots, a_{0(C-1)}, \ldots, a_{(C-1)(C-1)}\}$ where element $a_{cc'}$ denotes the probability that an object belongs to category c' when in the presence of an object belonging to category c. We compute the probabilities between all the category pairs using the annotations from the samples in the training set; thus, we do not introduce any additional annotation.

3.2 Graph Feature Propagation

Here, we describe the feature propagation module, which first decouples the image into C category-specific feature vectors via a semantically guided attention mechanism and then propagates these vectors throughout the graph to learn contextualized features for each category via a knowledge-embedded propagation mechanism.

3.2.1 Semantically guided attention mechanism

In this part, we introduce the semantically guided attention mechanism, which adopts the category semantic to guide the learning of semantic-specific feature representations. Specifically, given an input image I, this mechanism first extracts feature maps $\mathbf{f}^I \in \mathcal{R}^{W \times H \times N}$, where W, H, and N are the width, height and channel numbers of the feature maps, respectively, formulated as

$$\mathbf{f}^I = f_{cnn}(I), \tag{1}$$

where $f_{cnn}(\cdot)$ is a feature extractor implemented by a fully convolutional network. For each category c, the framework

extracts a d_s-dimensional semantic-embedding vector using the pretrained GloVe [58] model

$$\mathbf{x}_c = f_g(w_c), \tag{2}$$

where w_c is the semantic word representing category c. Then, we incorporate the semantic vector $\mathbf{x}_c$ to guide the focus to the semantic-aware regions and thus learn a feature vector corresponding to this category. More specifically, for each location (w, h), we first fuse the corresponding image features $\mathbf{f}^I_{wh}$ and $\mathbf{x}_c$ using a low-rank bilinear pooling method [59]

$$\tilde{\mathbf{f}}^I_{c,wh} = \mathbf{P}^T \left(\tanh \left((\mathbf{U}^T \mathbf{f}^I_{wh}) \odot (\mathbf{V}^T \mathbf{x}_c) \right) \right) + \mathbf{b}, \tag{3}$$

where $\tanh(\cdot)$ is the hyperbolic tangent function, and $\mathbf{U} \in \mathcal{R}^{N \times d_1}$, $\mathbf{V} \in \mathcal{R}^{d_s \times d_1}$, $\mathbf{P} \in \mathcal{R}^{d_1 \times d_2}$, and $\mathbf{b} \in \mathcal{R}^{d_2}$ are the learnable parameters, $\odot$ represents elementwise multiplication, and d_1 and d_2 are the dimensions of the joint embeddings and the output features, respectively. Then, we compute an attentional coefficient under the guidance of $\mathbf{x}_c$ as follows:

$$\tilde{a}_{c,wh} = f_a(\tilde{\mathbf{f}}^I_{c,wh}), \tag{4}$$

which indicates the importance of location (w, h). $f_a(\cdot)$ is an attentional function implemented by a fully connected network. The process is repeated for all locations. To allow the coefficients to be easily compared across different samples, we normalize them over all the locations using a softmax function:

$$a_{c,wh} = \frac{\exp(\tilde{a}_{c,wh})}{\sum_{w',h'} \exp(\tilde{a}_{c,w'h'})}. \tag{5}$$

Finally, we perform weighted average pooling over all the locations to obtain a feature vector

$$\mathbf{f}_c = \sum_{w,h} a_{c,wh} \mathbf{f}_{c,wh} \tag{6}$$

that encodes the information related to category c. We repeat the process for all categories and obtain all the category-related feature vectors $\{\mathbf{f}_0, \mathbf{f}_1, \ldots, \mathbf{f}_{C-1}\}$. In this way, we can learn the feature $\mathbf{f}_c$, which focuses more on the semantic

regions of each category c and primarily encodes the features for each category. We also present some examples for quantitative analysis in Fig. 3.

3.2.2 Knowledge-embedded feature propagation

After obtaining the feature vectors corresponding to all the categories, we initialize the nodes of $\mathcal{G}$ with the feature vector of the corresponding category and introduce a graph neural network to propagate messages through the graph to explore their interactions and learn a contextualized representation.

Inspired by the gated graph neural networks method [60], we adopt a gated recurrent update mechanism to propagate messages through the graph and learn the contextualized node-level features. Specifically, each node $v_c \in \mathbf{V}$ has a hidden state $\mathbf{h}_c^t$ at timestep t. In this work, because each node corresponds to a specific category and our model aims to explore the interactions among semantic-specific features, we initialize the hidden state at $t = 0$ with the feature vector that relates to the corresponding category, formulated as

$$\mathbf{h}_c^0 = \mathbf{f}_c. \tag{7}$$

At timestep t, the framework aggregates messages from its neighbor nodes as follows:

$$\mathbf{a}_c^t = \left[\sum_{c'} (a_{cc'}) \mathbf{h}_c^{t-1}, \sum_{c'} (a_{c'c}) \mathbf{h}_c^{t-1} \right]. \tag{8}$$

In this way, the framework encourages message propagation when node c' is highly correlated with node c; otherwise, it suppresses propagation. Therefore, this approach propagates messages through the graph and explores node interactions under the prior knowledge guidance of statistical label co-occurrences. Then, the framework updates the hidden state based on the aggregated feature vector $\mathbf{a}_c^t$ and its hidden state at the previous timestep $\mathbf{h}_c^{t-1}$ via a gated mechanism similar to the gated recurrent unit, formulated as

$$
\begin{aligned}
\mathbf{z}_c^t &= \sigma(\mathbf{W}^z \mathbf{a}_c^t + \mathbf{U}^z \mathbf{h}_c^{t-1}) \\
\mathbf{r}_c^t &= \sigma(\mathbf{W}^r \mathbf{a}_c^t + \mathbf{U}^r \mathbf{h}_c^{t-1}) \\
\widetilde{\mathbf{h}_c^t} &= \tanh\left(\mathbf{W}\mathbf{a}_c^t + \mathbf{U}(\mathbf{r}_c^t \odot \mathbf{h}_c^{t-1})\right) \\
\mathbf{h}_c^t &= (1 - \mathbf{z}_c^t) \odot \mathbf{h}_c^{t-1} + \mathbf{z}_c^t \odot \widetilde{\mathbf{h}_c^t},
\end{aligned}
\tag{9}
$$

where $\sigma(\cdot)$ is the logistic sigmoid function, $\tanh(\cdot)$ is the hyperbolic tangent function, and $\odot$ is elementwise multiplication. In this way, each node aggregates messages from the other nodes and simultaneously propagates their information through the graph, enabling interactions among all the feature vectors corresponding to all categories. The process is repeated T_f times; the generated final hidden states are $\{\mathbf{h}_0^{T_f}, \mathbf{h}_1^{T_f}, \ldots, \mathbf{h}_{C-1}^{T_f}\}$. Here, the hidden state of each node $\mathbf{h}_c^{T_f}$ not only encodes features from category c but also carries contextualized messages from other categories. Finally, we concatenate $\mathbf{h}_c^{T_f}$ and the input feature vector $\mathbf{h}_c^0$ to obtain the feature vector for each category

$$\mathbf{o}_c = f_o(\mathbf{h}_c^{T_f}, \mathbf{h}_c^0) \tag{10}$$

where $f_o(\cdot)$ is an output function that maps the concatenation of $\mathbf{h}_c^T$ and $\mathbf{h}_c^0$ into an output vector $\mathbf{o}_c$.

3.3 Graph Semantic Propagation

A classifier weight can be regarded as the prototype representation of the corresponding category. Transferring the prototype representations among similar categories can substantially enhance classifier training, especially for categories with few training samples. To achieve this, we integrate the graph $\mathcal{G}$ to guide the prototype representation propagation among all the categories to fully exploit the information of correlated categories and improve classifier training. Specifically, we initialize each graph node with the classifier weight of the corresponding category and iteratively propagate the node messages though the graph. In each iteration t, node v_c has a hidden state $\mathbf{h}_c^t$. The hidden state $\mathbf{h}_c^0$ at iteration $t = 0$ is set to the initial classifier weight $\mathbf{w}_c^{init}$, formulated as follows:

$$\mathbf{h}_c^0 = \mathbf{w}_c^{init}, \tag{11}$$

where $\mathbf{w}_c^{init}$ is randomly initialized before training. At each iteration t, each node k aggregates the messages from its correlated nodes such that the parameter vectors of those nodes can help refine the node's own parameter vector, formulated as follows:

$$\mathbf{a}_c^t = [\sum_{c'=1}^{K} a_{kk'} \mathbf{h}_c^{t-1}, \sum_{c'=1}^{K} a_{c'c} \mathbf{h}_c^{t-1}]. \tag{12}$$

In this way, a high correlation between nodes k and k' encourages message propagation from k' to k; otherwise, it suppresses the propagation. Then, the framework uses these aggregated feature vectors and the hidden state of the previous iteration as input to update the hidden state via a gated mechanism as Eq. 9. In this way, each node can aggregate more message from the nodes of correlated categories to help update its classifiers. The iteration is repeated T_s times; eventually, the final hidden states $\{\mathbf{h}_0^{T_s}, \mathbf{h}_1^{T_s}, \ldots, \mathbf{h}_{C-1}^{T_s}\}$ are generated. Finally, we also utilize a simple output network to predict the classifier weight

$$\mathbf{w}_c^* = w_o(\mathbf{h}_c^{T_s}, \mathbf{h}_c^0). \tag{13}$$

3.4 Network Architecture

Following existing multi-label image classification works [10], we implement the feature extractor $f_{cnn}(\cdot)$ based on the widely used ResNet-101 [19]. Specifically, we replace the last average pooling layer with another average pooling layer with a size of 2×2 and a stride of 2. The other layers remain unchanged in this implementation. For the low rank bilinear pooling operation, N, d_s, d_1, and d_2 are set to 2,048, 300, 1,024, and 1,024, respectively. Thus, $f_a(\cdot)$ is implemented by a 1,024-to-1 fully connected layer that maps the 1,024-feature vector to a single attentional coefficient.

The two gate graph networks share the same architecture. Specifically, the hidden state dimension is set to 2,048, while the iteration numbers T_f and T_s are set to 3. The output vector dimension $\mathbf{o}_c$ is also set to 2,048. Thus, the output networks $f_o(\cdot)$ and $w_o(\cdot)$ are implemented by a 4,096-to-2,048 fully connected layer followed by the hyperbolic tangent function.

3.5 From MLR to ML-FSL

As we obtain the feature vector $\mathbf{f}_c$ and classifier $\mathbf{w}_c^*$ for each category c, we multiply them to obtain the score:

$$s_c = \mathbf{w}_c^{*\top} \mathbf{f}_c. \tag{14}$$

We conduct this multiplication process for all the categories and obtain a score vector $\mathbf{s} = \{s_0, s_1, \ldots, s_{C-1}\}$. Next, we apply the framework to multi-label image recognition and adapt it to a more challenging task, i.e., multi-label few shot learning.

3.5.1 Multi-label Image Recognition

Given a dataset that contains M training samples $\{I_i, y_i\}_{i=0}^{M-1}$, in which I_i is the i-th image and $y_i = \{y_{i0}, y_{i1}, \ldots, y_{i(C-1)}\}$ is the corresponding annotation. y_{ic} is assigned a 1 if the sample is annotated with category c and 0 otherwise. Given an image I_i, we can obtain a predicted score vector $\mathbf{s}_i = \{s_{i0}, s_{i1}, \ldots, s_{i(C-1)}\}$ and compute the corresponding probability vector $\mathbf{p}_i = \{p_{i0}, p_{i1}, \ldots, p_{i(C-1)}\}$ via a sigmoid function

$$p_{ic} = \sigma(s_{ic}). \tag{15}$$

We adopt cross entropy as the objective loss function

$$\mathcal{L} = \sum_{i=0}^{N-1} \sum_{c=0}^{C-1} \left(y_{ic} \log p_{ic} + (1 - y_{ic}) \log(1 - p_{ic}) \right). \tag{16}$$

We train the proposed framework with the loss $\mathcal{L}$ in an end-to-end fashion. Specifically, we first apply the ResNet-101 parameters pretrained on the ImageNet dataset [61] to initialize the parameters of the corresponding layers in f_{cnn}; we initialize the parameters of other layers randomly. Because the lower layer parameters pretrained on the ImageNet dataset generalize well across different datasets, we fix the parameters of the previous 92 convolutional layers in $f_{cnn}(\cdot)$ and jointly optimize all the other layers. The framework is trained with the Adam algorithm [62] with a batch size of 4 and momentums of 0.999 and 0.9. The learning rate is initialized to 10^{-5} and divided by 10 when the error plateaus. During training, the input image is resized to 640×640, and we randomly choose a number from $\{640, 576, 512, 448, 384, 320\}$ as the width and height to crop random patches. Finally, the cropped patches are further resized to 576×576. During testing, we simply resize the input image to 640×640 and perform a center crop with a size of 576×576 for evaluation.

3.5.2 Multi-Label Few-Shot Learning

For multi-label few-shot learning, the dataset contains a set of C_b base categories with sufficient training samples and a set of C_n novel categories with limited training samples (e.g., 1, 2, or 5). Because the training samples for the base and novel categories are extremely unbalanced, it inevitably leads to poor performances if using the loss as Equation (16) for training, especially on the novel categories. Thus, we follow the previous work [63] to adopt a two-stage process to train the proposed framework.
Stage 1. In the first stage, we train the framework using the training samples of the base categories. Here, we have C_b categories, and construct the adjacent matrix $\mathbf{A} = \mathbf{A}_b \in$ $\mathcal{R}^{C_b \times C_b}$ based on the co-occurrences among these C_b categories (following the process described in Sec. 3.1). Given a sample from the C_b categories, we can compute the score vector $\mathbf{s}_i = \{s_{i0}, s_{i1}, \ldots, s_{i(C_b-1)}\}$ and the corresponding probability vector $\mathbf{p}_i = \{p_{i0}, p_{i1}, \ldots, p_{i(C_b-1)}\}$. Then, we define the loss similarly to Eq. 16, expressed as

$$\mathcal{L}_b = \sum_{i=0}^{N_b-1} \sum_{c=0}^{C_b-1} \left(y_{ic} \log p_{ic} + (1 - y_{ic}) \log(1 - p_{ic}) \right). \tag{17}$$

where N_b is the number of training samples from the base set. Similarly, the parameters of the backbone network are initialized with the parameters pretrained on the ImageNet dataset, while those of the other layers are randomly initialized. We also fix the previous convolutional layers in the backbone networks and jointly train all the other layers using the Adam algorithm with a batch size of 4 and momentums of 0.999 and 0.9. We set the initial learning rate to 10^{-5} and divided it by 10 after 12 epochs. The framework is trained with 20 epochs in total. During training, the augmentation strategy is the same as that adopted for multi-label recognition (described in Sec. 3.5.1).
Stage 2. In the second stage, we fix the parameters of the backbone network and the attention modules for feature extraction and train the two graph neural networks using the novel sets. Because the training samples are limited during this stage, we cannot compute the statistical co-occurrence to obtain the adjacent matrix. To address this issue, we consider semantic similarity to compute the label correlations. Specifically, we use Glove [58] to extract semantic vector for each category, compute the cosine distance between each category pair to obtain their semantic similarities, and construct the adjacent matrix $\mathbf{A} = \mathbf{A}_n \in \mathcal{R}^{C_n \times C_n}$. In addition, we introduce an additional regularization term on the classifier weights and thus the loss function can be formulated as

$$\mathcal{L}_b = \sum_{i=0}^{N_n-1} \sum_{c=0}^{C_n-1} \left(y_{ic} \log p_{ic} + (1 - y_{ic}) \log(1 - p_{ic}) \right)$$
$$+ \gamma \sum_{c=0}^{C_n-1} ||\mathbf{w}_c^*||_2^2, \tag{18}$$

where N_n is the number of samples from the novel set and γ is a balance parameter that is set to 0.001. We train the model using the Adam algorithm with the same batch size, momentums, and data augmentation strategy as in stage 1. Because we merely train the two graph neural networks while fixing other layers, we set a large learning rate 10^{-4} and train the framework for merely 500 iterations.

4 EXPERIMENTS

In this section, we conduct extensive experiments on various benchmarks to evaluate the performance of the proposed framework over existing state-of-the-art methods and perform ablative studies to further analyze the actual contribution of each component.

4.1 Evaluation Metrics

For multi-label image recognition, we follow current state-of-the-art works [2], [32] by adopting the average precision

Methods	mAP	Top 3						All					
		CP	CR	CF1	OP	OR	OF1	CP	CR	CF1	OP	OR	OF1
WARP [64]	-	59.3	52.5	55.7	59.8	61.4	60.7	-	-	-	-	-	-
CNN-RNN [17]	-	66.0	55.6	60.4	69.2	66.4	67.8	-	-	-	-	-	-
RLSD [28]	-	67.6	57.2	62.0	70.1	63.4	66.5	-	-	-	-	-	-
RARL [18]	-	78.8	57.2	66.2	84.0	61.6	71.1						
RDAR [11]	73.4	79.1	58.7	67.4	84.0	63.0	72.0	-	-	-	-	-	-
KD-WSD [65]	74.6	-	-	66.8	-	-	72.7	-	-	69.2	-	-	74.0
ResNet-SRN-att [10]	76.1	85.8	57.5	66.3	88.1	61.1	72.1	81.2	63.3	70.0	84.1	67.7	75.0
ResNet-SRN [10]	77.1	85.2	58.8	67.4	87.4	62.5	72.9	81.6	65.4	71.2	82.7	69.9	75.8
ML-GCN [66]	83.1	89.2	64.1	74.6	90.5	66.5	76.7	85.1	72.0	78.0	85.8	75.4	80.3
Ours	84.3	89.4	64.6	75.0	91.3	66.6	77.0	85.6	72.7	78.6	87.1	75.6	80.9

TABLE 1
Comparison of the mAP, CP, CR, CF1 and OP, OR, OF1 (in %) scores of our framework and other state-of-the-art methods under settings of all and top-3 labels on the Microsoft COCO dataset. A hyphen "-" denotes that no corresponding result was provided.

(AP) on each category and the mean average precision (mAP) over all categories. For more comprehensive comparisons, we further follow the current works [1], [10] to present the precision, recall, and F1-measure values. Here, we assign the labels with the top-3 highest scores for each image and compare them with the ground truth labels. Concretely, we adopt overall precision, recall, and F1-measure (OP, OR, and OF1) and per-class precision, recall, and F1-measure (CP, CR, and CF1), which are defined as shown below:

$$\text{OP} = \frac{\sum_i N_i^c}{\sum_i N_i^p}, \qquad \text{CP} = \frac{1}{C}\sum_i \frac{N_i^c}{N_i^p},$$

$$\text{OR} = \frac{\sum_i N_i^c}{\sum_i N_i^g}, \qquad \text{CR} = \frac{1}{C}\sum_i \frac{N_i^c}{N_i^g}, \qquad (19)$$

$$\text{OF1} = \frac{2 \times \text{OP} \times \text{OR}}{\text{OP} + \text{OR}}, \quad \text{CF1} = \frac{2 \times \text{CP} \times \text{CR}}{\text{CP} + \text{CR}},$$

where C is the number of labels, N_i^c is the number of images for which the i-th label is correctly predicted, N_i^p is the number of images for which the i-th label is predicted, and N_i^g is the number of ground truth images for the i-th label. The above metrics require a fixed number of labels, but the label numbers of different images generally vary. Thus, we further present the OP, OR, OF1 and the CP, CR, CF1 metrics using the idea that a label is predicted as positive when its estimated probability is greater than 0.5 [10]. Among these metrics, mAP, OF1, and CF1 are the most important and provide a more comprehensive evaluation.

4.2 Datasets

Pascal VOC 2007 & 2012 [25] are the datasets most widely used to evaluate the multi-label image classification task, and most of the existing works report their results on these datasets. Therefore, we also conducted experiments on these two datasets for evaluation purposes. Specifically, both datasets include 20 common categories. Pascal VOC 2007 contains a training and validation (trainval) set with 5,011 images and a test set with 4,952 images, while VOC 2012 includes 11,540 images as the trainval set and 10,991 images as the test set. For fair comparison purposes, the proposed framework and existing competitors were all trained on the trainval set and evaluated on the test set. However, because the number of categories in these two datasets is limited, we use these datasets merely to evaluate the multi-label image recognition task; we do not use them to evaluate the multi-label few-shot learning task.

Microsoft COCO [26] was originally constructed for object detection and segmentation but has recently been adopted to evaluate multi-label image classification. This dataset contains 122,218 images and covers 80 common categories. The dataset is further divided into a training set with 82,081 images and a validation set with 40,137 images. Because no ground-truth annotations are available for the test set, we trained our method and all the compared methods on the training set and evaluated them on the validation set.

For multi-label few-shot learning, we follow the previous work [63] to split the 80 categories into 64 base categories and 16 novel categories. Specifically, the novel categories are *bicycle, boat, stop sign, bird, backpack, frisbee, snowboard, surfboard, cup, fork, spoon, broccoli, chair, keyboard, microwave,* and *vase*. To ensure fair comparisons with current works, we utilize the trainval samples of the base categories as the base set and randomly select K (K=1,5) trainval samples from the novel categories as the novel set. We evaluate the test samples of the novel categories.

Visual Genome [27] is a dataset that contains 108,249 images and covers 80,138 categories. Because most categories have very few samples, we consider only the 500 most frequent categories, resulting in a Visual Genome 500 (VG-500) subset. We randomly selected 10,000 images as the test set and the remaining 98,249 images as the training set. Compared with other existing benchmarks, this one covers more categories, i.e., 500 categories as opposed to 20 in Pascal VOC [25] and 80 in Microsoft-COCO [26]; thus, it can be used to evaluate multi-label image recognition performance with larger-scale categories.

For multi-label few-shot learning, we split VG-500 into 400 base categories and 100 novel categories. Similar to Microsoft COCO, we used all training samples of the base categories as the base set and selected K (K=1,5) training samples of the novel categories as the novel set. Then, we evaluate the models on the test samples of the novel categories.

4.3 Results on Multi-Label Image Recognition

In this subsection, we provide the comparison of the proposed framework with current state-of-the-art methods on the multi-label image recognition task.

4.3.1 Comparison on Microsoft COCO

We first present the evaluation results on the Microsoft COCO [26] dataset. To fairly compare the OP, OR, OF1 and

Methods	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
CNN-RNN [17]	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
RMIC [67]	97.1	91.3	94.2	57.1	86.7	90.7	93.1	63.3	83.3	76.4	92.8	94.4	91.6	95.1	92.3	59.7	86.0	69.5	96.4	79.0	84.5
VGG16+SVM [20]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	89.3
VGG19+SVM [20]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	89.3
RLSD [28]	96.4	92.7	93.8	94.1	71.2	92.5	94.2	95.7	74.3	90.0	74.2	95.4	96.2	92.1	97.9	66.9	93.5	73.7	97.5	87.6	88.5
HCP [2]	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
FeV+LV [8]	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
RDAR [11]	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
RARL [18]	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
RCP [68]	99.3	97.6	98.0	96.4	79.3	93.8	96.6	97.1	78.0	88.7	87.1	97.1	96.3	95.4	99.1	82.1	93.6	82.2	98.4	92.8	92.5
Ours	99.8	97.1	98.4	98.0	84.2	95.1	96.9	98.4	78.6	94.9	87.0	98.1	97.7	97.4	98.7	82.4	97.1	82.5	98.7	92.0	93.6
Ours (pre)	99.3	98.6	97.9	98.4	86.2	97.0	98.0	99.2	82.6	98.3	87.5	99.0	98.9	97.4	99.1	86.9	98.2	84.1	99.0	95.0	95.0
VGG16&19+SVM [20]	98.9	95.0	96.8	95.4	69.7	90.4	93.5	96.0	74.2	86.6	87.8	96.0	96.3	93.1	97.2	70.0	92.1	80.3	98.1	87.0	89.7
FeV+LV (fusion) [8]	98.2	96.9	97.1	95.8	74.3	94.2	96.7	96.7	76.7	90.5	88.0	96.9	97.7	95.9	98.6	78.5	93.6	82.4	98.4	90.4	92.0

TABLE 2

Comparison of the AP and mAP (in %) results of our framework and those of state-of-the-art methods on the PASCAL VOC 2007 dataset. The upper part presents the results of single models and the lower part presents those that aggregate multiple models. "Ours" and "Ours (pre)" denote our framework without and with pretraining on the COCO dataset, respectively. The best and second-best results are highlighted in red and blue, respectively. A hyphen "-" denotes that no corresponding result was provided. Best viewed in color.

Methods	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
RMIC [67]	98.0	85.5	92.6	88.7	64.0	86.8	82.0	94.9	72.7	83.1	73.4	95.2	91.7	90.8	95.5	58.3	87.6	70.6	93.8	83.0	84.4
VGG16+SVM [20]	99.0	88.8	95.9	93.8	73.1	92.1	85.1	97.8	79.5	91.1	83.3	97.2	96.3	94.5	96.9	63.1	93.4	75.0	97.1	87.1	89.0
VGG19+SVM [20]	99.1	88.7	95.7	93.9	73.1	92.1	84.8	97.7	79.1	90.7	83.2	97.3	96.2	94.3	96.9	63.4	93.2	74.6	97.3	87.9	89.0
HCP [2]	99.1	92.8	97.4	94.4	79.9	93.6	89.8	98.2	78.2	94.9	79.8	97.8	97.0	93.8	96.4	74.3	94.7	71.9	96.7	88.6	90.5
FeV+LV [8]	98.4	92.8	93.4	90.7	74.9	93.2	90.2	96.1	78.2	89.8	80.6	95.7	96.1	95.3	97.5	73.1	91.2	75.4	97.0	88.2	89.4
RCP [68]	99.3	92.2	97.5	94.9	82.3	94.1	92.4	98.5	83.8	93.5	83.1	98.1	97.3	96.0	98.8	77.7	95.1	79.4	97.7	92.4	92.2
Ours	99.6	95.1	97.5	96.9	83.1	94.2	96.2	98.9	86.5	97.0	84.9	98.9	98.9	96.6	98.8	84.0	98.3	84.0	99.6	94.0	93.9
Ours (pre)	99.6	96.8	97.9	96.7	87.3	96.5	96.2	99.1	87.9	97.7	86.8	99.3	99.3	97.5	99.1	85.39	98.8	84.9	99.6	94.4	95.0
VGG16&19+SVM [20]	99.1	89.1	96.0	94.1	74.1	92.2	85.3	97.9	79.9	92.0	83.7	97.5	96.5	94.7	97.1	63.7	93.6	75.2	97.4	87.8	89.3
FeV+LV (fusion) [8]	98.9	93.1	96.0	94.1	76.4	93.5	90.8	97.9	80.2	92.1	82.4	97.2	96.8	95.7	98.1	73.9	93.6	76.8	97.5	89.0	90.7
HCP+AGS [2], [69]	99.8	94.8	97.7	95.4	81.3	96.0	94.5	98.9	88.5	94.1	86.0	98.1	98.3	97.3	97.3	76.1	93.9	84.2	98.2	92.7	93.2
RCP+AGS [68], [69]	99.8	94.5	98.1	96.1	85.5	96.1	95.5	99.0	90.2	95.0	87.8	98.7	98.4	97.5	99.0	80.1	95.9	86.5	98.8	94.6	94.3
Ours (pre & fusion)	99.8	97.3	98.4	97.1	87.9	97.3	96.5	99.3	89.4	97.8	88.7	99.4	99.4	97.9	99.2	86.3	98.8	86.3	99.7	95.2	95.6

TABLE 3

Comparison of the AP and mAP scores (in %) of our model and those of state-of-the-art methods on the PASCAL VOC 2012 dataset. The upper part presents the results of single models and the lower part presents those that aggregate multiple models. "Ours" and "Ours (pre)" denote our framework without and with pretraining on the COCO dataset. "Ours (pre & fusion)" denotes the results when fusing our two scale results. The best and second-best results are highlighted in red and blue, respectively. Best viewed in color.

CP, CR, CF1 metrics with the top-3 constraint, we follow the existing method [17] to exclude labels with probabilities below a threshold (0.5 in our experiments). The comparison results are presented in Table 1. As shown, the current best-performing methods are ResNet-SRN and ML-GCN, in which ResNet-SRN builds on ResNet-101 and applies an attention mechanism to model the label relations, while ML-GCN further uses a graph convolutional network to capture the label dependencies and achieves a mAP of 83.1%. In contrast to these methods, our framework models label dependencies in both the feature and label spaces, leading to notable performance improvements on all the metrics. Specifically, it achieves a mAP of 84.3%, improving on the results of the previous best methods by 1.2%.

4.3.2 Comparisons on Pascal VOC 2007 and 2012

Here, we present the AP of each category and the mAP over all categories on the Pascal VOC 2007 dataset in Table 2. Most of the existing state-of-the-art methods focus on locating informative regions (e.g., proposal candidates [2], [8], [28], attentive regions [11], or random regions [68]) to aggregate local discriminative features to facilitate recognizing multiple labels in the given image. For example, RCP achieves a mAP of 92.5%, which is the best result to date. In contrast, our framework incorporates category semantics to better learn semantic-specific features and explores their interactions under the explicit guidance of statistical label

dependencies, which further improves the mAP to 93.6%. In addition, when our framework is pretrained on the COCO dataset, it achieves an even better performance, i.e., 95.0%, as shown in Table 2. Note that the existing methods aggregate multiple models [20] or fuse the results with other methods [8] to improve the overall performance. For example, FeV+LV (fusion) aggregates its results with those of VGG16&19+SVM, which improves the mAP from 90.6% to 92.0%. Although our results are generated by a single model, it still outperforms all these aggregated results.

We also compare the performances on the Pascal VOC 2012 dataset, as depicted in Table 3. VOC 2012 is more challenging and larger in size, but our framework still achieves the best performance compared with the state-of-the-art competitors. Specifically, it obtains mAP scores of 93.9% and 95.0% without and with pretraining on the COCO dataset, respectively, with improvements over the previous best method by 1.7% and 2.8%, respectively. Similarly, the existing methods also aggregate the results of multiple models to boost performance. To ensure a fair comparison, we trained another model with an input of 448×448. Specifically, during training, we resized the input image to 512×512, and randomly choose a number from 512, 448, 384, 320, 256 as the width and height to randomly crop patches; then, we further resized the cropped patches to 448×448. We denote the original model as "scale 640" and this modified model as "scale 512". The two models are
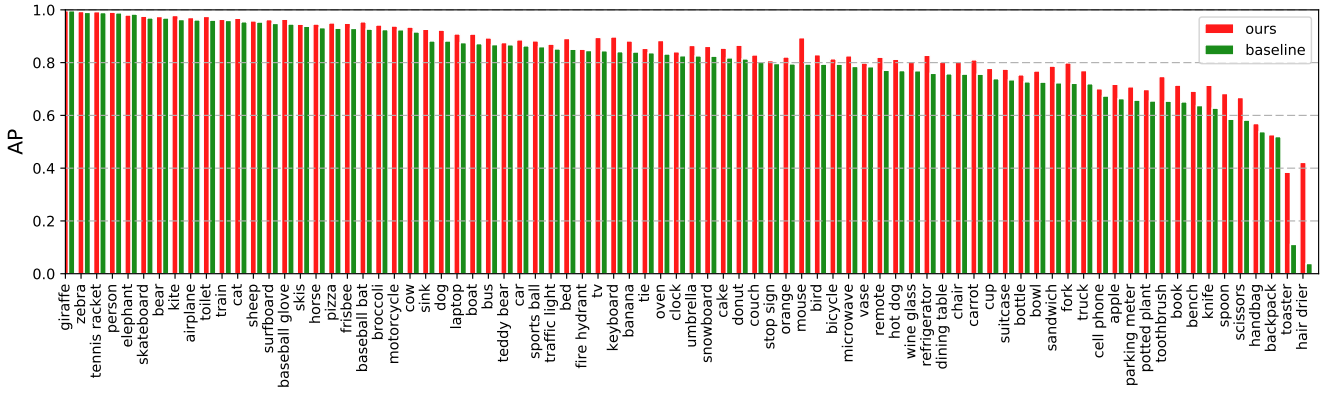
Fig. 2. The AP (in %) of each category of our proposed framework and the ResNet-101 baseline on the Microsoft COCO dataset.

Methods	mAP	Top 3						All					
		CP	CR	CF1	OP	OR	OF1	CP	CR	CF1	OP	OR	OF1
ResNet-101 [19]	30.9	39.1	25.6	31.0	61.4	35.9	45.4	39.2	11.7	18.0	75.1	16.3	26.8
ML-GCN [66]	32.6	42.8	20.2	27.5	66.9	31.5	42.8	39.4	10.6	16.8	77.1	16.4	27.1
Ours	**37.4**	**47.4**	**24.7**	**32.5**	**66.9**	**36.5**	**47.2**	**48.7**	**12.1**	**19.4**	**78.6**	**17.1**	**28.1**

TABLE 4

Comparison of the mAP, CP, CR, CF1 and OP, OR, OF1 (in %) scores of our framework and other state-of-the-art methods under settings of all and top-3 labels on the VG-500 dataset.

both pretrained on the COCO dataset and retrained on the VOC 2012 dataset. Then, we perform ten crop evaluations (the four corner crops and the center crop as well as their horizontally flipped versions) for each scale and aggregate the results from the two scales. As shown in the lower part of Table 3, our framework boosts the mAP to 95.6%, outperforming all the existing methods with single and multiple models.

4.3.3 Comparison on Visual Genome 500

The Visual Genome 500 is a new dataset that can be used to evaluate multi-label image recognition with larger-scale categories; thus, no existing methods have reported their results on this dataset. To demonstrate the effectiveness of our proposed framework on this dataset, we implemented a ResNet-101 baseline network and trained it using the same process as was used for our model previously. Because ML-GCN [32] is the best-performing method on the Microsoft-COCO dataset, we further follow its released code to train the model on VG-500 for comparison. All the methods were trained on the training set and evaluated on the test set. The comparison results are presented in Table 4. Our framework performs considerably better than the existing state-of-the-art and the ResNet-101 baseline methods on all metrics. Specifically, it achieves the mAP, top-3 CF1 and OF1, and top-all CF1 and OF1 of 37.4%, 32.5%, 47.2%, 19.4%, 28.1%, improving the existing best method ML-GCN by 4.8%, 1.5%, 1.8%, 1.4%, 1.0%, respectively. This comparison suggests that our framework also performs better for recognizing large-scale categories.

4.4 Results on Multi-Label Few-Shot Learning

In this subsection, we present comparisons of our proposed framework with current methods on the multi-label few-shot learning task.

Methods	1-shot	5-shot
Baseline w/ Mixup Aug [70]	40.2	54.0
LaSO (GoogleNet-v3) [63]	45.3	58.1
Ours (GoogleNet-v3)	49.4	61.0
Ours (ResNet-101)	52.3	63.5

TABLE 5

Comparison of the mAP (in %) on 1-shot and 5-shot settings on the Microsoft COCO dataset. We present the results of both 1-shot and 5-shot settings.

4.4.1 Comparison on Microsoft COCO

The previous best-performing method for multi-label few-shot learning is LaSO [63], and its results are mainly presented on Microsoft COCO. In this section, we compare our proposed method with this work and the baseline method that uses the mixUp [70] augmentation technique to directly train on the small novel set. The results are presented in Table 5. As shown, LaSO achieves mAPs of 45.3% and 58.1% on the 1-shot and 5-shot settings, respectively. In contrast, our method incorporates prior knowledge of category correlation to guide feature and semantic propagation among the different categories, leading to superior performance. Specifically, our method's mAP scores on the 1-shot and 5-shot settings are 52.3% and 63.5%, outperforming LaSO by 7.0% and 5.4%, respectively. Note that we use ResNet-101 [19] as the backbone, while LaSO uses GoogleNet-v3 [71], [72]. To ensure a fair comparison, we further replaced our backbone network with GoogleNet-v3, keeping the other components and the training process unchanged. Nevertheless, our framework still outperforms LaSO, achieving mAP scores of 49.4% and 61.0% on the 1-shot and 5-shot settings, respectively.

4.4.2 Comparison on Visual Genome 500

Microsoft-COCO contains only 80 categories, but current works [73], [74] for few-shot learning may contain thou-
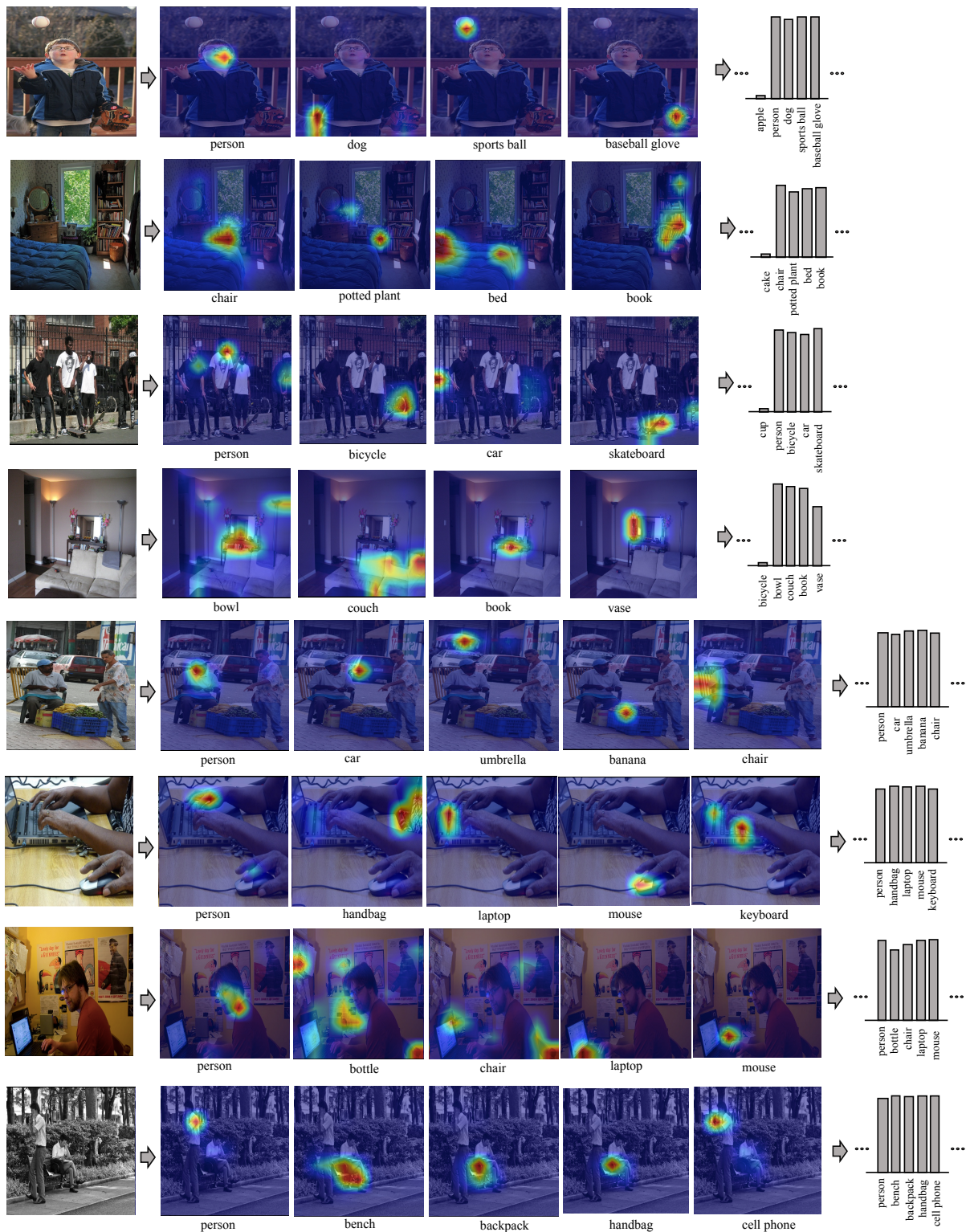
Fig. 3. Several examples of input images (left), semantic feature maps corresponding to categories with the top-k ($k = 4, 5$) highest confidences (middle), and the predicted label distribution (right).

sands of categories, which is more realistic. To remedy this issue, we utilize Visual Genome 500 as a new evaluation benchmark. Because LaSO is the previous best-performing method, we followed the code [1] released by the authors to train the LaSO model on the VG-500 dataset. For a fair comparison, we used ResNet-101 as the backbone network

1. https://github.com/leokarlin/LaSO

for both methods. The results are listed in Table 6. Our framework clearly outperform the LaSO by a sizable margin. Specifically, our framework achieves mAPs of 20.7% and 26.1% on the 1-shot and 5-shot settings, outperforming LaSO by 4.1% and 4.3%, respectively.

Methods	1-shot	5-shot
LaSO [63]	16.6	21.8
Ours	20.7	26.1

TABLE 6
Comparison of the mAP (in %) on 1-shot and 5-shot settings on the VG-500 dataset. We present the results of 1-shot and 5-shot settings.

4.5 Ablative study

The proposed framework builds on ResNet-101 [19]; thus, we first compare it with this baseline to analyze the contributions of knowledge-guided graph routing (KGGR). Specifically, we simply replace the last fully connected layer of ResNet-101 with a 2,048-to-C fully connected layer and use C sigmoid functions to predict the probability of each category. The training and test settings are the same as those described in Section 3.5.1. We conducted the experiments on the Microsoft-COCO dataset and present the results in Table 7. As shown, the mAP drops from 84.3% to 80.3%. To more deeply analyze the performance comparisons, we further present the AP of each category in Figure 2, which shows that the AP improvement by our framework is more evident for the categories that are more difficult to recognize (i.e., the categories for which the baseline obtains lower AP). For example, for categories such as giraffe and zebra, the baseline obtains a very high AP; thus, our framework achieves only slight improvements. In contrast, for more difficult categories such as toaster and hair drier, our framework improves the AP by a sizable margin—27.3% and 38.3% improvements for toaster and hair drier, respectively.

We also present a comparison with the baseline ResNet-101 on the multi-label few-shot learning task in Table 8. As shown, our method achieves evident improvements over the baseline (i.e., improving the mAP by 8.8% and 7.0% on the 1-shot and 5-shot settings, respectively).

Methods	mAP
ResNet-101 [19]	80.3
Ours w/o KEFP	80.9
Ours w/o SGA	82.2
Ours w/o GFP	80.6
Ours w/o GSP	83.8
Ours	84.3

TABLE 7
Comparison of mAP (in %) of our framework (Ours), our framework without the graph feature propagation module (Ours w/o GFP), our framework without the graph semantic propagation module (Ours w/o GSP), our framework without semantically guided attention (Ours w/o SGA), our framework without knowledge-embedded feature propagation (Ours w/o KEFP), and the baseline ResNet-101 from the multi-label image recognition task on the Microsoft-COCO dataset.

The foregoing comparisons verify the contribution of the proposed KGGR as a whole. Actually, the KGGR contains two graph propagation modules that conduct information

Methods	1-shot	5-shot
ResNet-101 [19]	43.5	56.5
Ours w/o GFP	44.1	56.0
Ours w/o GSP	50.4	61.0
Ours	52.3	63.5

TABLE 8
Comparison of mAP (in %) of our framework (Ours), our framework without the graph feature propagation module (Ours w/o GFP), our framework without the graph semantic propagation module (Ours w/o GSP), and the baseline ResNet-101 from the multi-label few-shot task on the Microsoft-COCO dataset.
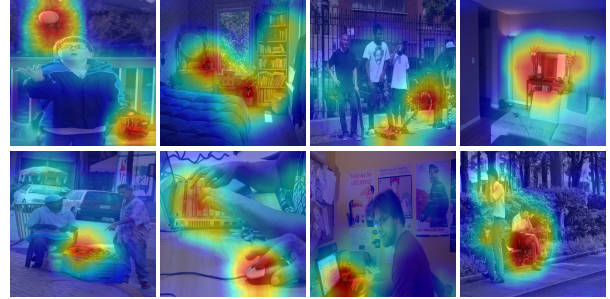

Fig. 4. Several examples of the feature maps generated by the baseline ResNet-101. The samples are the same as those in Figure 3 for direct comparisons.

interactions to learn features and classifiers. In the following, we further conduct ablation experiments to analyze the true contribution of each module.

4.5.1 Contribution of graph feature propagation

Graph feature propagation can help learn the contextualized feature vectors for each category, and it is the key module in this work. Here, we first remove this module and directly use $\mathbf{f}$ for classification. We find that the resulting model suffers from an obvious performance drop (i.e., the mAP decreased from 84.3% to 80.6%). Moreover, this module uses a semantically guided attention mechanism that decouples the image into C category-specific feature vectors and a knowledge-embedded feature propagation that explores feature interactions to learn contextualized features. We also analyze these two components in the following. 1) To analyze the semantically guided attention mechanism, we remove it and directly use $\mathbf{f}$ to initialize the graph nodes. As shown in Table 8, this model achieve a mAP of 80.9%. It performs slightly better than the baseline method, because it does not incur any additional information but does increase the model complexity. As discussed above, this component can learn semantic-specific feature maps that focus on corresponding semantic regions via semantically guided attention mechanism. We present some examples in Figure 3. From left to right in each row are the input images, the semantic maps corresponding to categories with the top $k(k = 4, 5)$ highest confidences, and the predicted label distribution. These results show that our semantically guided attention mechanism is able to highlight the semantic regions well when the objects of the corresponding categories exist. Taking the first sample as an example, it contains person, dog, sport ball, and baseball glove objects, which highlight the corresponding regions of these semantic objects, respectively. Similar phenomena can

be observed for the other examples. To clearly verify that it is the semantically guided attention mechanism that brings about this appealing characteristic, we further visualize the feature maps generated by the baseline ResNet-101. As shown in Figure 4, it tends to highlight the most salient regions, but neglect some regions that are less salient but equally important, e.g., the dog in the first example and the chair in the seventh example. 2) To validate the knowledge-embedded feature propagation component, we remove the graph propagation network; thus, $\mathbf{o}_c = \mathbf{f}_c$. As shown in Table 7, the mAP is 82.2%, a decrease in the mAP of 2.1%.

We also analyze the effect of this module on the multi-label few-shot task and present the result with and without the graph feature propagation component in Table 8. The mAP falls from 52.3% and 63.5% to 44.1% and 56.0% on the 1-shot and 5-shot settings, respectively.

4.5.2 Contribution of graph semantic propagation

Graph semantic propagation helps transfer the information of correlated categories to better learn the classifiers. To validate its contribution, we remove this component and use C fully connected layers that directly take the corresponding contextualized feature vector as input to predict the probability of the corresponding category. As shown in Table 7, the mAP decreases from 84.3% to 83.8% on Microsoft COCO. We found a similar phenomenon for the multi-label few-shot task, where the mAP decreases from 52.3% and 63.5% to 50.4% and 61.0% on the 1-shot and 5-shot settings, respectively.

4.5.3 Analysis of different losses

In this work, we learn a contextualized feature vector for each category and adopt the logistic regressor to predict the existence probability for each category. We follow recent multi-label works [32], [41] to train the models with the cross-entropy loss. Earlier works [2], [18] has also used the euclidean loss between the output score and normalized ground truth to train the model. In this part, we conduct an experiment that follows works [2], [18] to replace the loss with Euclidean loss to analyze the effect of different losses. As shown in Table 9, the model trained with the entropy loss achieves much better performance.

Methods	mAP
Ours EU	79.2
Ours CE	84.3

TABLE 9
Comparison of mAP (in %) of our framework with the cross entropy (Ours CE) and euclidean (Ours EU) losses on the Microsoft-COCO dataset.

5 CONCLUSION

In this work, we explore integrating prior knowledge of label correlations into deep neural networks to guide learning both feature and classifier representations. To achieve this end, we propose a novel knowledge-guided graph routing (KGGR) framework that consists of two graph propagation mechanisms. The first propagation mechanism introduces category semantic to guide learning semantic-specific features and exploit a graph n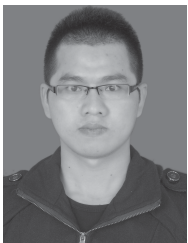eural network to explore feature interaction to learn contextualized feature representation for each category. The second propagation mechanism exploits another graph neural network that takes initial classifier weight as input and transfers classifier information through different categories to help better learn the classifiers. We apply the proposed framework to both multi-label image recognition and multi-label few-shot learning tasks on the Microsoft-COCO, Pascal VOC 2007 & 2012, and Visual Genome datasets, and demonstrate its effectiveness over all existing leading methods on both two tasks.

REFERENCES

[1] Y. Li, Y. Song, and J. Luo, "Improving pairwise ranking for multi-label image classification," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 1837–1845.

[2] Y. Wei, W. Xia, M. Lin, J. Huang, B. Ni, J. Dong, Y. Zhao, and S. Yan, "Hcp: A flexible cnn framework for multi-label image classification," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 9, pp. 1901–1907, 2016.

[3] R. Cabral, F. De la Torre, J. P. Costeira, and A. Bernardino, "Matrix completion for weakly-supervised multi-label image classification," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 1, pp. 121–135, 2014.

[4] H. O. Song, R. Girshick, S. Zickler, C. Geyer, P. Felzenszwalb, and T. Darrell, "Generalized sparselet models for real-time multiclass object recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 5, pp. 1001–1012, 2014.

[5] M. Lapin, M. Hein, and B. Schiele, "Analysis and optimization of loss functions for multiclass, top-k, and multilabel classification," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 7, pp. 1533–1554, 2017.

[6] T.-S. Chua, H.-K. Pung, G.-J. Lu, and H.-S. Jong, "A concept-based image retrieval system," in *Proceedings of the International Conference on System Sciences, 1994.*, vol. 3. IEEE, 1994, pp. 590–598.

[7] X. Yang, Y. Li, and J. Luo, "Pinterest board recommendation for twitter users," in *Proceedings of the ACM international conference on Multimedia*. ACM, 2015, pp. 963–966.

[8] H. Yang, J. Tianyi Zhou, Y. Zhang, B.-B. Gao, J. Wu, and J. Cai, "Exploit bounding box annotations for multi-label object recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 280–288.

[9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," pp. 1137–1149, 2017.

[10] F. Zhu, H. Li, W. Ouyang, N. Yu, and X. Wang, "Learning spatial regularization with image-level supervisions for multi-label image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5513–5522.

[11] Z. Wang, T. Chen, G. Li, R. Xu, and L. Lin, "Multi-label image recognition by recurrently discovering attentional regions," in *Computer Vision (ICCV), 2017 IEEE International Conference on*. IEEE, 2017, pp. 464–472.

[12] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013.

[13] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *European Conference on Computer Vision*. Springer, 2014, pp. 391–405.

[14] Z. Zhang, Y. Liu, X. Chen, Y. Zhu, M.-M. Cheng, V. Saligrama, and P. H. Torr, "Sequential optimization for efficient high-quality object proposal generation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 5, pp. 1209–1223, 2017.

[15] J. Pont-Tuset, P. Arbelaez, J. T. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping for image segmentation and object proposal generation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 1, pp. 128–140, 2016.

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[17] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu, "Cnn-rnn: A unified framework for multi-label image classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2285–2294.

[18] T. Chen, Z. Wang, G. Li, and L. Lin, "Recurrent attentional reinforcement learning for multi-label image recognition," in *Proc. of AAAI Conference on Artificial Intelligence*, 2018, pp. 6730–6737.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, Y. Bengio and Y. LeCun, Eds., 2015.

[21] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 1199–1208.

[22] J. Kim, T.-H. Oh, S. Lee, F. Pan, and I. S. Kweon, "Variational prototyping-encoder: One-shot learning with prototypical images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 9462–9470.

[23] B. Hariharan and R. Girshick, "Low-shot visual recognition by shrinking and hallucinating features," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2017, pp. 3037–3046.

[24] T. Chen, M. Xu, X. Hui, H. Wu, and L. Lin, "Learning semantic-specific graph representation for multi-label image recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 522–531.

[25] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[26] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European Conference on Computer Vision*. Springer, 2014, pp. 740–755.

[27] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma *et al.*, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *International Journal of Computer Vision*, vol. 123, no. 1, pp. 32–73, 2017.

[28] J. Zhang, Q. Wu, C. Shen, J. Zhang, and J. Lu, "Multi-label image classification with regional latent semantic dependencies," *IEEE Transactions on Multimedia*, 2018.

[29] N. Ghamrawi and A. McCallum, "Collective multi-label classification," in *Proceedings of the 14th ACM international conference on Information and knowledge management*. ACM, 2005, pp. 195–200.

[30] Y. Guo and S. Gu, "Multi-label classification using conditional dependency networks," in *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, 2011, p. 1300.

[31] X. Xue, W. Zhang, J. Zhang, B. Wu, J. Fan, and Y. Lu, "Correlative multi-label multi-instance image annotation," in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 651–658.

[32] Z.-M. Chen, X.-S. Wei, P. Wang, and Y. Guo, "Multi-label image recognition with graph convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5177–5186.

[33] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra *et al.*, "Matching networks for one shot learning," in *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2016, pp. 3630–3638.

[34] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 4077–4087.

[35] S. Qiao, C. Liu, W. Shen, and A. L. Yuille, "Few-shot image recognition by predicting parameters from activations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7229–7238.

[36] H. Qi, M. Brown, and D. G. Lowe, "Low-shot learning with imprinted weights," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 5822–5830.

[37] A. Ravichandran, R. Bhotika, and S. Soatto, "Few-shot learning with embedded class models and shot-free meta training," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 331–339.

[38] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017, pp. 1126–1135.

[39] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, 201.

[40] E. Schwartz, L. Karlinsky, J. Shtok, S. Harary, M. Marder, A. Kumar, R. S. Feris, R. Giryes, and A. M. Bronstein, "Delta-encoder: an effective sample synthesis method for few-shot object recognition," in *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2018, pp. 2850–2860.

[41] P. Tokmakov, Y.-X. Wang, and M. Hebert, "Learning compositional representations for few-shot recognition," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 6372–6381.

[42] A. Alfassy, L. Karlinsky, A. Aides, J. Shtok, S. Harary, R. S. Feris, R. Giryes, and A. M. Bronstein, "Laso: Label-set operations networks for multi-label few-shot learning," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 6548–6557.

[43] T. Chen, L. Lin, L. Liu, X. Luo, and X. Li, "Disc: Deep image saliency computing via progressive representation learning," *IEEE transactions on neural networks and learning systems*, vol. 27, no. 6, pp. 1135–1149, 2016.

[44] T. Chen, L. Lin, X. Wu, N. Xiao, and X. Luo, "Learning to segment object candidates via recursive neural networks," *IEEE Transactions on Image Processing*, vol. 27, no. 12, pp. 5827–5839, 2018.

[45] K. Marino, R. Salakhutdinov, and A. Gupta, "The more you know: Using knowledge graphs for image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2673–2681.

[46] C. Jiang, H. Xu, X. Liang, and L. Lin, "Hybrid knowledge routed modules for large-scale object detection," in *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2018, pp. 1552–1563.

[47] Y. Tang, J. Wang, X. Wang, B. Gao, E. Dellandréa, R. J. Gaizauskas, and L. Chen, "Visual and semantic knowledge transfer for large scale semi-supervised object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 3045–3058, 2018.

[48] T. Chen, W. Wu, Y. Gao, L. Dong, X. Luo, and L. Lin, "Fine-grained representation learning and recognition by exploiting hierarchical semantic embedding," in *Proceedings of ACM Conference on Multimedia*, 2018, pp. 2023–2031.

[49] Y. Xie, T. Chen, T. Pu, H. Wu, and L. Lin, "Adversarial graph representation adaptation for cross-domain facial expression recognition," in *Proceedings of the 28th ACM international conference on Multimedia*, 2020.

[50] T. Chen, W. Yu, R. Chen, and L. Lin, "Knowledge-embedded routing network for scene graph generation," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 6163–6171.

[51] Z. Wang, T. Chen, J. Ren, W. Yu, H. Cheng, and L. Lin, "Deep reasoning with knowledge graph for social relationship understanding," in *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 2018, pp. 2021–2018.

[52] T. Chen, R. Chen, L. Nie, X. Luo, X. Liu, and L. Lin, "Neural task planning with AND-OR graph representations," *IEEE Transactions on Multimedia*, vol. 21, no. 4, pp. 1022–1034, 2018.

[53] W. Yang, X. Wang, A. Farhadi, A. Gupta, and R. Mottaghi, "Visual semantic navigation using scene priors," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2019.

[54] X. Wang, Y. Ye, and A. Gupta, "Zero-shot recognition via semantic embeddings and knowledge graphs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 6857–6866.

[55] A. Li, T. Luo, Z. Lu, T. Xiang, and L. Wang, "Large-scale few-shot learning: Knowledge transfer with class hierarchy," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 7212–7220.

[56] X. Yang, W. Yu, R. Wang, G. Zhang, and F. Nie, "Fast spectral clustering learning with hierarchical bipartite graph for large-scale data," *Pattern Recognition Letters*, vol. 130, pp. 345–352, 2020.

[57] S. Zhang, S. Guo, L. Wang, W. Huang, and M. R. Scott, "Knowledge integration networks for action recognition," in *Proceedings of AAAI Conference on Artificial Intelligence*, 2020.

[58] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[59] J.-H. Kim, K.-W. On, W. Lim, J. Kim, J.-W. Ha, and B.-T. Zhang, "Hadamard product for low-rank bilinear pooling," *arXiv preprint arXiv:1610.04325*, 2016.

[60] Y. Li, D. Tarlow, M. Brockschmidt, and R. S. Zemel, "Gated graph sequence neural networks," in *International Conference on Learning Representations*, 2016.

[61] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 248–255.

[62] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015.

[63] A. Alfassy, L. Karlinsky, A. Aides, J. Shtok, S. Harary, R. Feris, R. Giryes, and A. M. Bronstein, "Laso: Label-set operations networks for multi-label few-shot learning," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[64] Y. Gong, Y. Jia, T. Leung, A. Toshev, and S. Ioffe, "Deep convolutional ranking for multilabel image annotation," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2014.

[65] Y. Liu, L. Sheng, J. Shao, J. Yan, S. Xiang, and C. Pan, "Multi-label image classification via knowledge distillation from weakly-supervised detection," in *ACM MM*, 2018.

[66] Z. Chen, X. Wei, P. Wang, and Y. Guo, "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*, pp. 5177–5186.

[67] S. He, C. Xu, T. Guo, C. Xu, and D. Tao, "Reinforced multi-label image classification by exploring curriculum," in *AAAI*, 2018.

[68] M. Wang, C. Luo, R. Hong, J. Tang, and J. Feng, "Beyond object proposals: Random crop pooling for multi-label image recognition," *IEEE Transactions on Image Processing*, vol. 25, no. 12, pp. 5678–5688, 2016.

[69] J. Dong, W. Xia, Q. Chen, J. Feng, Z. Huang, and S. Yan, "Subcategory-aware object classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 827–834.

[70] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," 2017.

[71] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[72] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[73] H. Zhang, J. Zhang, and P. Koniusz, "Few-shot learning via saliency-guided hallucination of samples," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2770–2779.

[74] R. Chen, T. Chen, X. Hui, H. Wu, G. Li, and L. Lin, "Knowledge graph transfer network for few-shot recognition," in *AAAI Conference on Artificial Intelligence*, 2020.

Liang Lin (M'09, SM'15) is a Full Professor of computer science at Sun Yat-sen University. He has authored or co-authored more than 200 papers in top-tier academic journals and conferences with more than 12,000 citations. He is an associate editor of IEEE Trans. Human-Machine Systems and IET Computer Vision. He served as Area Chairs for numerous conferences such as CVPR, ICCV, and IJCAI. He is the recipient of numerous awards and honors including Wu Wen-Jun Artificial Intelligence Award, CISG Science and Technology Award, ICCV Best Paper Nomination in 2019, Annual Best Paper Award by Pattern Recognition (Elsevier) in 2018, Best Paper Dimond Award in IEEE ICME 2017, Google Faculty Award in 2012, and Hong Kong Scholars Award in 2014. He is a Fellow of IET.

Riquan Chen received a B.E. degree from the School of Mathematics, Sun Yat-sen University, Guangzhou, China, in 2017, where he is currently pursuing his master's degree in computer science at the School of Data and Computer Science. His current research interests include computer vision and machine learning.

Xiaolu Hui received her B.E. degree from the School of Computing, Northwestern Polytechnical University, Shaanxi, China, in 2018. She is currently pursuing a masters degree in computer science with the School of Data and Computer Science at Sun Yat-sen University. Her current research interests include computer vision and machine learning.

Tianshui Chen received a Ph.D. degree in computer science at the School of Data and Computer Science Sun Yat-sen University, Guangzhou, China, in 2018. Before that, he received a B.E. degree from the School of Information and Science Technology. He is currently an Associate Research Director at DMAI Co., Ltd. His current research interests include computer vision and machine learning. He has authored and coauthored more than 20 papers published in top-tier academic journals and conferences. He has served as a reviewer for numerous academic journals and conferences, including TPAMI, TIP, TMM, TNNLS, CVPR, ICCV, ECCV, AAAI, and IJCAI. He was the recipient of the Best Paper Diamond Award at IEEE ICME 2017.

Hefeng Wu received a B.S. degree in computer science and technology and a Ph.D. in computer application technology from Sun Yat-sen University, China, in 2008 and 2013, respectively. He is currently a full research scientist with the School of Data and Computer Science, Sun Yat-sen University, China. His research interests include computer vision, multimedia, and machine learning.