Knowledge Graph Transfer Network for Few-Shot Recognition

Riquan Chen,1 Tianshui Chen,1,2 Xiaolu Hui,1 Hefeng Wu,1∗ Guanbin Li,1 Liang Lin1,2
1Sun Yat-sen University, 2DarkMatter AI Research
sysucrq@gmail.com, tianshuichen@gmail.com, huixlu@mail2.sys.edu.cn, wuhefeng@gmail.com
liguanbin@mail.sysu.edu.cn, linlng@mail.sysu.edu.cn

Abstract

Few-shot learning aims to learn novel categories from very few samples given some base categories with sufficient training samples. The main challenge of this task is the novel categories are prone to dominated by color, texture, shape of the object or background context (namely specificity), which are distinct for the given few training samples but not common for the corresponding categories (see Figure 1). Fortunately, we find that transferring information of the correlated based categories can help learn the novel concepts and thus avoid the novel concept being dominated by the specificity. Besides, incorporating semantic correlations among different categories can effectively regularize this information transfer. In this work, we represent the semantic correlations in the form of structured knowledge graph and integrate this graph into deep neural networks to promote few-shot learning by a novel Knowledge Graph Transfer Network (KGTN). Specifically, by initializing each node with the classifier weight of the corresponding category, a propagation mechanism is learned to adaptively propagate node message through the graph to explore node interaction and transfer classifier information of the base categories to those of the novel ones. Extensive experiments on the ImageNet dataset show significant performance improvement compared with current leading competitors. Furthermore, we construct an ImageNet-6K dataset that covers larger scale categories, i.e, 6,000 categories, and experiments on this dataset further demonstrate the effectiveness of our proposed model.

Introduction

Recently, deep convolutional neural networks (ConvNet) (Krizhevsky, Sutskever, and Hinton 2012; He et al. 2016) have obtained remarkable success in various visual recognition tasks. To fully train a deep ConvNet recognition system, it is requested that each category has thousands of training samples. If the system needs to recognize some novel categories, we need to collect large amounts of training samples for these categories to avoid being overfitting. In contrast, human can effortlessly learn to recognize novel categories with few samples by exploiting prior knowledge accumulated from daily life. Mimicking this ability to learn novel categories from very few samples (also known as few-shot learning) is a crucial yet practical task in the computer vision community.

Formally, we consider a general and practical scenario for few-shot learning, in which there exist a set of base categories with sufficient training samples and a set of novel categories with very limited samples (e.g., no more than ten samples in this work). Unlike previous works that merely focus on the novel categories, we aim to develop a recog-
The contributions are summarized into three folds: 1) We propose to integrate category correlations as prior knowledge to regularize and guide transferring information of classifier weights. 2) We introduce a graph update mechanism that propagates the node message through the graph to iteratively update the classifier weights. 3) We conduct experiments on the widely used ImageNet based few-shot dataset and demonstrate the superior performance of our proposed framework over existing state-of-the-art methods. To evaluate the performance on larger scale categories, we construct an ImageNet-6k dataset that covers 6,000 categories. Experiments conducted on this benchmark shows that our model still outperforms current leading methods.

Related Works

Few-shot learning. Previous works (Finn, Abbeel, and Levine 2017; Li et al. 2017; Vinyals et al. 2016; Garcia and Bruna 2017) follow a simple N-way K-shot setting, where there are N categories with K training samples for each category, and N is no more than 20. These works often adopt learning-to-learn paradigm that distills knowledge learned from training categories to help learn novel concepts. For example, (Vinyals et al. 2016; Snell, Swersky, and Zemel 2017; Sung et al. 2018; Kim et al. 2019) learn an embedding and metric function from the base categories to well recognize samples in the novel categories. Most of these works evaluate their algorithms on some small-scale datasets, e.g., miniImageNet with 64 base categories, 16 validation categories, 20 novel categories. Recently, some works (Hariharan and Girshick 2017; Chen et al. 2019c; Hui, Chen, and Chen 2019) switch to a more general and practical setting, where the algorithms aim to recognize hundreds of novel concepts with very limited samples given a set of base categories with sufficient training samples. To address this few-shot learning scenario, (Hariharan and Girshick 2017) learn a transformation function to hallucinate additional samples for novel categories. (Chen et al. 2019c) further consider to enlarge the size of dataset and learn a deformation network to deform images by fusing a pair of images. (Qiao et al. 2018b) explore the relation of training feature and classifier weight and adapt a neural network to obtain classifier weights directly from the training features. To evaluate these algorithms, researchers construct a larger-scale dataset that covers 1,000 categories (Hariharan and Girshick 2017). As this is a more practical scenario, we focus on this setting in our work.

Knowledge Embedded Visual Reasoning. Recently, lots of works attempt to incorporate prior knowledge with deep representation learning for various visual reasoning tasks, ranging from visual recognition/detection (Wang, Ye, and Gupta 2018; Chen et al. 2018c; Lee et al. 2018; Chen et al. 2019a; Chen et al. 2018b; Jiang et al. 2018) to visual relationship...
reasoning (Chen et al. 2019b; Wang et al. 2018b) and navigation/planning (Yang et al. 2019; Chen et al. 2018a). As a pioneering work, (Marino, Salakhutdinov, and Gupta 2017) build a knowledge graph to correlate object categories and learn graph representation to enhance image representation learning. (Chen et al. 2019a) incorporate a similar knowledge graph to guide feature encoding and interaction to further facilitate multi-label image classification. (Wang, Ye, and Gupta 2018) apply graph convolutional network (Kipf and Welling 2016) to explore semantic convolutional and direct map semantic vectors to classifier parameters. To further explore high-level tasks, (Chen et al. 2019b) consider the correlations between specific object pairs and their corresponding relationships to regularize scene graph generation and thus alleviate the effect of the uneven distribution issue. (Chen et al. 2018a) build And-Or Graphs (Zhu, Mumford, and others 2007) to describe tasks, which helps regularize atomic action sequence generation to reduce the dependency on annotated samples. In few-shot learning scenario, the latest work (Li et al. 2019) construct category hierarchy by semantic cluster and regularize prediction at each hierarchy via a hierarchical output net. Different from these works, we formulate classifier weight as a prototype representation of the corresponding category, and introduce a graph propagation mechanism to transfer prototype representation of base categories to guide learning novel concepts under the explicit guidance of prior semantic correlation.

Methodology

We first revisit the few-shot problem formulation in the conventional deep ConvNet paradigm with some formalized interpretation on why such paradigm easily fails in the few-shot scenarios, and then present our model in detail.

Problem Formulation Revisited

A few-shot recognition system should be able to recognize novel categories with limited few samples. Recent deep ConvNet models have achieved remarkable success in image recognition, but they are also notorious for requiring a large number of training samples for each category and are not fit for the few-shot scenarios. In this work, we amend the conventional paradigm to adapt deep ConvNet models to such scenarios.

A typical deep ConvNet-based image recognition model consists of the feature extractor and classifiers, which are jointly learned for optimal performance. Given an image sample \( x \), the recognition model predicts its label \( \hat{y} \) as

\[
\hat{y} = \arg \max_k p(y = k|x) = \frac{\exp(f_k(x))}{\sum_K \exp(f_i(x))}
\]

is the softmax function. Here \( K \) is the number of categories, and \( x \) is the \( d \)-dimensional representation feature of \( x \) output by the feature extractor \( \phi(-) \), i.e., \( x = \phi(x) \). The linear classifier \( f_k(x) = w_k^T x + b_k \) computes the confidence of the sample \( x \) belonging to category \( k \), and is implemented with a fully-connected layer. \( w_k \) denotes the classifier weight and \( b_k \) is the bias term. It can be easily inferred that \( \arg \max_k p(y = k|x) \) is equivalent to \( \arg \max_k f_k(x) \).

We reformulate \( f_k(x) \) as follows:

\[
f_k(x) = w_k^T x + b_k = -\frac{1}{2}||w_k - x||^2_2 + \frac{1}{2}||w_k||^2_2 + \frac{1}{2}||x||^2_2 + b_k
\]

By introducing the constraints \( b_k = 0 \) and \( ||w_k||_2 = ||w_j||_2 \), \( \forall i, j \), the classifier \( f_k(x) \) can be viewed as a similarity measure between \( x \) and \( w_k \), and we have

\[
\hat{y} = \arg \max_k f_k(x) = \arg \min_k ||w_k - x||^2_2
\]

Therefore, the weight \( w_k \) can be viewed as the prototype representation of category \( k \), and the sample \( x \) is predicted as category \( k \) if its feature \( x \) has the maximum similarity (or minimum distance) with prototype \( w_k \). In this perspective, we can implement \( f_k(x) \) with different similarity metrics.

Relaxing the above constraints to general deep ConvNet models, the prototype representation perspective of classifier weight is reasonable to some extent. Thus, when these models are applied in the few-shot scenarios, the learned prototype will be guided to reflect the specificity of the few training samples (as visualized in Figure 2) and cannot capture the commonality of the corresponding category.

To tackle this problem, we amend the conventional deep ConvNet paradigm by incorporating category correlations to transfer prototype presentations among similar categories and thus substantially enhance the prototypes of novel categories. Specifically, the category correlations are modeled as a knowledge graph and integrated into the deep ConvNet to build our Knowledge Graph Transfer Network model, which will be detailed below.

Knowledge Graph Transfer Network

The overall framework of our Knowledge Graph Transfer Network (KGTN) is illustrated in Figure 3, which consists of three modules: Feature Extraction, Knowledge Graph Transfer, and Prediction. The key design is the knowledge graph transfer module where we incorporate a graph neural network to explore the knowledge transfer of the prototypes (classifier weights) by the guidance of semantic correlations on top of the ConvNet.

Knowledge Graph Transfer Module. We model the classifier weight into a graph, in which nodes refer to the classifier weight and edges represent their semantic correlations. Then, we incorporate a Gated Graph Neural Network (GGNN) to update and propagate the message between nodes.

Given a dataset that covers \( K = K_{base} + K_{novel} \) categories (\( K_{base} \) and \( K_{novel} \) denote the number of base and novel categories), we use a graph \( G = \{\mathbf{V}, \mathbf{A}\} \) to encode the correlations among all categories, in which nodes refer to the categories and edges represent their correlations. Specifically, \( \mathbf{V} \) is represented as \( \{v_1, v_2, \ldots, v_{K_{base}}, \ldots, v_K\} \) where node \( v_k \) corresponds to category \( k \), and \( \mathbf{A} \) is an adjacency matrix, in which \( a_{ij} \) denotes the correlation weight between categories \( i \) and \( j \).

At each iteration \( t \), node \( v_k \) has a hidden state \( h_k^t \), and the hidden state \( h_k^0 \) at iteration \( t = 0 \) is set as the initial classifier weight \( w_k^{init} \), formulated as

\[
h_k^0 = w_k^{init},
\]
in which \(w_k^{init}\) is learnable parameters and randomly initialized before training. At each iteration \(t\), each node \(k\) aggregates message from its correlated node such that the parameter vectors of these nodes can help refine its parameter vector, formulated as

\[
a_k^t = \left[ \sum_{k'=1}^{K} a_{kk'} h_{k'}^{t-1}, \sum_{k'=1}^{K} a_{kk'} h_{k'}^{t-1} \right].
\]  

In this way, a high correlation between nodes \(k\) and \(k'\) encourages message propagation from \(k'\) to \(k\), and it suppresses the propagation otherwise. Then, the framework takes this aggregated feature vector and the hidden state of the previous iteration as input to update the corresponding hidden state by a gating mechanism

\[
\begin{align*}
\bar{z}_k^t &= \sigma(W^z a_k^t + U h_k^{t-1}) \\
\bar{r}_k^t &= \sigma(W^r a_k^t + U h_k^{t-1}) \\
\tilde{h}_k^t &= \tanh(W a_k^t + U (\bar{r}_k^t \odot h_k^{t-1})) \\
h_k^t &= (1 - \bar{z}_k^t) \odot h_k^{t-1} + \bar{z}_k^t \odot \tilde{h}_k^t
\end{align*}
\]  

where \(\sigma(\cdot)\) and \(\tanh(\cdot)\) are the logistic sigmoid and hyperbolic tangent functions, and \(\odot\) is the element-wise multiplication operation. In this way, the model tends to adopt the more correlated message to update parameters of the current node. The propagation is repeated by \(T\) iterations, and we can obtain the final hidden states, i.e., \(\{h_0^t, h_1^t, \ldots, h_K^t\}\). Finally, we utilize a simple output network to predict the classifier weight

\[
w_k^* = o(h_K^T, h_0^0).
\]  

**Prediction with different similarity metrics.** As stated in the problem formulation, the classifier \(f_k(x)\) can be implemented as similarity metric. Here, we consider three similarity metrics for evaluation: inner product, cosine similarity, and Pearson correlation coefficient.

**Inner product:**

\[
f_k(x) = x \cdot w_k^*
\]  

**Cosine similarity:**

\[
f_k(x) = \frac{x \cdot w_k^*}{\|x\|_2 \cdot \|w_k^*\|_2}
\]  

**Pearson correlation coefficient:**

\[
f_k(x) = \frac{(x - \bar{x}) \cdot (w_k^* - w_k^*)}{\|x - \bar{x}\|_2 \cdot \|w_k^* - w_k^*\|_2}
\]  

where \(\bar{x}\) is a \(d\)-dimensional vector with all elements being the same value computed by averaging all elements in \(x\), as is likewise for \(w_k^*\).

The above-defined similarity metric will be put into the softmax function as denoted by Equation (1) to obtain the final prediction. For cosine similarity and Pearson correlation metrics, the values output by the softmax may be extremely small, since \(\|f_k(x)\|_2 \leq 1\) in such situations. So, similar to (Gidaris and Komodakis 2018, Q. Brown, and Lowe 2018), we multiply a learnable scale parameter \(s\), i.e., putting \(s \cdot f_k(x)\) into the softmax for these two metrics.

**Optimization**

Similar to (Hariharan and Girshick 2017), we adopt a two-stage training procedure to train the proposed model.

**Stage 1:** At the first stage, we train the feature extractor \(\phi(\cdot)\) based on the base set \(D_{base}\). Given an image sample \(x_i\) with label \(y_i\), we first compute its probability distribution \(p_i = \{p_{i1}, p_{i2}, \ldots, p_{ik_{base}}\}\) with \(p_{ik} = p(y = k|x_i)\), and then define the cross-entropy loss as our objective function. To make the learned features easily generalize to the novel categories, we further introduce Squared Gradient Magnitude loss proposed in (Hariharan and Girshick 2017) to regularize representation learning. Thus, the objective function at
this stage can be defined as
\[
\mathcal{L}_1 = \mathcal{L}_c + \lambda \mathcal{L}_s
\]
where
\[
\mathcal{L}_c = -\frac{1}{N_{base}} \sum_{i=1}^{N_{base}} \sum_{k=1}^{K_{base}} 1(k = y_i) \log p_{i}^{k}
\]
\[
\mathcal{L}_s = \frac{1}{N_{base}} \sum_{i=1}^{N_{base}} \sum_{k=1}^{K_{base}} (p_{i}^{k} - 1(k = y_i))^2 ||x_i||^2
\]
where \(N_{base}\) is the number of training samples in all base categories, \(1(k = y_i)\) is an indicator that equals 1 when \(k = y_i\) is true and 0 otherwise, \(\lambda\) is a parameter to balance two loss terms and it is set as 0.005 by following [Hariharan and Girshick 2017]. At this stage, the model is trained using the SGD algorithm with a batch size of 256, momentum of 0.9, and weight decay of 0.0005. The learning rate is initialized as 0.1 and is divided by 10 for every 30 epochs.

**Stage 2:** We fix the parameters of the feature extractor and use both the base and novel set to train the other components, including \(w^{\text{init}}\), the parameters of GGNN and scale factor \(s\). Similarly, we obtain its probability vector \(p_i = \{p_i^1, p_i^2, \ldots, p_i^K\}\) and use the cross-entropy loss as the objective function. To handle class imbalance, we sample the novel and base samples by 1:1 ratio in each batch. Besides, for inner product, we introduce an additional regularization term on the classifier weights, thus the overall objective function can be defined as
\[
\mathcal{L}_2 = -\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} 1(k = y_i) \log p_{i}^{k} + \eta \sum_{k=1}^{K} ||w_{k}^s||^2
\]
where \(N\) is the number of all training samples, \(\eta\) is a parameter to balance two loss terms and it is set as 0.001 empirically. In this stage, we train the model using SGD algorithm with a batch size of 1,000, momentum of 0.9, weight decay of 0.0001, and learning rate of 0.01.

**Experiment**

**Graph Construction**

The knowledge graph encodes the correlations among different categories. It can be constructed according to different prior knowledge. Here, we introduce two kinds of knowledge, i.e., semantic similarity and category hierarchy.

**Semantic similarity.** Semantic word of a specific category well carries its semantic information, and the semantic distance of two categories encodes their correlations. In other words, two categories are of high correlation if their semantic distance is small, and are of low correlation otherwise. Thus, we first exploit this property to construct the graph. Specifically, given two categories \(i\) and \(j\) with semantic words \(w_i\) and \(w_j\), we first extract their semantic feature vector \(f_i^w\) and \(f_j^w\) using the GloVe model [Pennington, Socher, and Manning, 2014] and compute their Euclidean distance \(d_{ij}\). Then, we apply a monotone decreasing function \(a_{ij} = \lambda d_{ij} - \min d_{ij} \forall j \neq i\) (we set \(\lambda = 0.4\)) to map the distance to the correlation coefficient \(a_{ij}\). Besides, we also consider the self-message of each node so that we set \(a_{ii} = 1\).

**Category hierarchy.** Category hierarchy encodes category correlations via different levels of concept abstraction. Generally, the distance from one category to another indicates their correlation, where a small distance indicates a high correlation while a large distance indicates a low correlation. In this work, we also exploit this knowledge based on the WordNet [Miller 1995] to construct the graph. Concretely, given two categories \(i\) and \(j\), we compute the shortest path from node \(i\) to \(j\) as the distance \(d_{ij}\) and we also apply a similar monotone decreasing function to map the distance to the correlation coefficient \(a_{ij}\).

**Datasets**

Unlike previous few-shot benchmark with low-resolution images and few novel categories, we consider the more realistic dataset with large scale base and novel categories: ImageNet Few-Shot (ImageNet-FS) dataset. To further verify our approach, we construct a more challenging dataset ImageNet-6K, which covers 5,000 novel categories.

**ImageNet-FS.** In this work, we first evaluate our proposed model on the widely used ImageNet-FS dataset. The dataset covers 1,000 categories from ILSVRC2012 and is divided into 389 base categories and 611 novel categories where 193 base categories and 300 novel categories are used for cross-validation and the remaining 196 categories and 311 novel categories are used for testing. Each base category has about 1,280 training images and 50 test images.

**ImageNet-6K.** To evaluate our proposed method on larger scale categories, we further construct a new benchmark ImageNet-6K that covers 6,000 categories. It contains 1,000 categories from ImageNet 2012 dataset with all the labeled training samples as the base categories, and we further select 5,000 categories from the ImageNet 2011 as novel categories. Concretely, 2,500 categories are used for validation and the rest 2,500 categories for final evaluation. Each category has 50 test samples.

For \(k\)-shot learning, only \(k\) labeled images of the novel categories are used. We follow previous works [Hariharan and Girshick 2017] [Wang et al. 2018a] to repeat the process by 5 times and report the average accuracy. Here, we set \(k\) as 1, 2, 5, 10 on ImageNet-FS and ImageNet-6K.

**Evaluation Metrics.** We follow previous works [Wang et al. 2018a] to evaluate our proposed model on the top-5 accuracy of the novel categories (Acc of novel) and all (base + novel) categories (Acc of all).

**Comparison with State-of-the-Art**

**Performance on ImageNet-FS dataset.** We present the results of the above metrics on 1, 2, 5, 10 shot learning on the ImageNet-FS in Table T. As shown, our proposed model outperforms all existing methods by a sizable margin. Take the “novel” metric as example, our model achieves the accuracies of 62.1%, 70.9%, 78.4%, and 82.3%, outperforming current best results by 2.0%, 1.3%, 1.0%, and 0.3%, on 1, 2, 5, 10 shot learning, respectively.

Notably, compared with KTCH which also introduces word embedding as the external knowledge, we construct semantic correlation based on word embedding to transferring information between classifier weight and obtain an im-
Table 1: Top5-accuracy in “novel” and “all” metrics of our model and current state-of-the-art methods on the ImageNet-FS and ImageNet-6K datasets. For fair comparison, all the methods use ResNet-50 for feature extraction. Some methods train an additional generator to hallucinate extra training samples for the novel categories (w/ G). The best and second best results are highlighted in bold and italic, respectively. “-” denotes the corresponding result is not provided.

| Dataset   | Method                                      | Novel 1 | Novel 2 | Novel 5 | Novel 10 | All   1 | All   2 | All   5 | All   10 |
|-----------|---------------------------------------------|---------|---------|---------|----------|-------|--------|--------|--------|
| ImageNet-FS | MN (Vanysals et al. 2016)                  | 53.5    | 63.5    | 72.7    | 77.4     | 64.9  | 71.0   | 77.0   | 80.2   |
|           | PN (Snell, Swersky, and Zemel 2017)        | 49.6    | 64.0    | 74.4    | 78.1     | 61.4  | 71.4   | 78.0   | 80.0   |
|           | SGM (Hariharan and Girshick 2017)          | 54.3    | 67.0    | 77.4    | 81.9     | 60.7  | 71.6   | 80.2   | 83.6   |
|           | SGM w/ G (Hariharan and Girshick 2017)     | 52.9    | 64.9    | 77.3    | 82.0     | 63.9  | 71.9   | 80.2   | 83.6   |
|           | AWG (Gidaris and Komodakis 2018)           | 53.9    | 65.5    | 75.9    | 80.3     | 65.1  | 72.3   | 79.1   | 82.1   |
|           | PMN (Wang et al. 2018a)                    | 53.3    | 65.2    | 75.9    | 80.1     | 64.8  | 72.1   | 78.8   | 81.7   |
|           | PMN w/ G (Wang et al. 2018a)               | 54.7    | 66.8    | 77.4    | 81.4     | 65.7  | 73.5   | 80.2   | 82.8   |
|           | LSD (Douze et al. 2018)                    | 57.7    | 66.9    | 73.8    | 77.6     | –     | –      | –      | –      |
|           | KTCH (Li et al. 2019)                      | 58.1    | 67.3    | 77.6    | 81.8     | –     | –      | –      | –      |
|           | IDeMeNet (Chen et al. 2019c)               | 60.1    | 69.6    | 77.4    | 80.2     | –     | –      | –      | –      |
|           | Ours (CosSim)                              | 61.4    | 70.4    | 78.4    | 82.2     | 67.7  | 74.7   | 80.9   | 83.6   |
|           | Ours (PearsonCorr)                         | 61.5    | 70.6    | 78.5    | 82.3     | 67.5  | 74.4   | 80.7   | 83.5   |
|           | Ours (InnerProduct)                        | 62.1    | 70.9    | 78.4    | 82.3     | 68.3  | 75.2   | 80.8   | 82.5   |
| ImageNet-6K | SGM (Hariharan and Girshick 2017)          | 26.7    | 35.0    | 44.7    | 51.5     | 37.7  | 44.8   | 53.1   | 58.5   |
|           | SGM w/ G (Hariharan and Girshick 2017)     | 26.3    | 35.5    | 46.2    | 52.0     | 39.4  | 46.4   | 54.4   | 58.8   |
|           | AWG (Gidaris and Komodakis 2018)           | 27.6    | 35.9    | 45.0    | 49.4     | 39.3  | 45.2   | 52.1   | 55.4   |
|           | Ours (CosSim)                              | 30.5    | 37.5    | 46.0    | 51.7     | 41.0  | 46.8   | 53.8   | 58.5   |
|           | Ours (PearsonCorr)                         | 30.4    | 37.2    | 45.9    | 51.8     | 41.1  | 46.8   | 53.7   | 58.5   |
|           | Ours (InnerProduct)                        | 29.5    | 36.8    | 46.1    | 52.1     | 40.8  | 46.9   | 54.4   | 58.7   |

Performance on ImageNet-6K dataset. All the foregoing comparisons focus on recognizing about 500 categories. In fact, there are much more categories in a real-world setting. To evaluate the proposed model on larger scale categories, we further conduct experiments on the ImageNet-6K dataset. As SGM (Hariharan and Girshick 2017) and AWG (Gidaris and Komodakis 2018) are current leading methods, we use the released codes to implement these methods on this dataset for comparison. Following (Hariharan and Girshick 2017), we train models with and without generating samples for the novel categories. The comparison results are presented in Table 1. Even though ImageNet-6K is more challenging and much larger in category size, our proposed model still outperforms existing methods. Specifically, it obtains the “novel” accuracy of 30.5%, 37.5%, an improvement of 2.9%, 1.6% on the 1-shot and 2-shot learning compared with existing methods. This comparison suggests the proposed method is scalable in category size.

Ablative study

Analysis of Knowledge Embedding. The core of our proposed model is the prior knowledge graph that correlates base and novel categories to help regularize parameter propagation. As discussed above, our model follows SGM (Hariharan and Girshick 2017) to use ResNet-50 as feature extractor and also use identical loss function for optimization, thus SGM can be regarded as the baseline of our model. In this part, we emphasize the comparison with SGM to evaluate the effectiveness of knowledge embedding model. As shown in Table 1, our framework significantly outperforms SGM, with accuracy improvement of 7.8%, 7.6% on two metrics in 1-shot setting.

To further analyze the effect of knowledge embedding, we further replace the category correlations with other non-informative form to demonstrate its benefit. Specifically, we consider the following two variants: 1) Uniform graph in which all the correlation values are uniformly set as 1/2 and 2) Random graph in which all the correlation values are randomly selected from a uniform distribution. The comparison results are presented in Table 2. We find that these two variants perform comparably with each other as both incur no additional information. Note that they achieve slightly better results than the baseline SGM. One possible reason is that knowledge propagation can help to better learn the classifier weights of novel categories. Still, our model with prior knowledge embedding significantly outperforms both two variants on all metrics for 1, 2, 5, 10 shot learning. These comparisons clearly demonstrate the benefit of knowledge embedding.

Analysis on Different Kinds of Knowledge. The correlations between categories can be constructed based on different kinds of knowledge, e.g., semantic similarity and category hierarchy in the paper. Here, we further conduct experiments with these two kinds of knowledge and present their results in Table 2. We find that introducing both kinds of knowledge leads to obvious improvement than the baseline SGM and those with non-informative correlations, which
In this work, we explore incorporating prior knowledge of category correlations to guide exploiting knowledge from base categories to help learn the classifier weights of novel categories. To this end, we formulate a novel Knowledge Graph Transfer Network model, which correlates classifier weights with a graph constructed based on their correlations and introduce a graph neural network to iteratively propagate and update classifier weights. Extensive experiments on the widely used ImageNet-FS and newly constructed ImageNet-6K dataset demonstrate the effectiveness of our proposed model over state-of-the-art methods.

**Conclusion**

Table 2: Top5-accuracy in “novel” and “all” metrics of our proposed model with semantic correlation knowledge (s-graph), with category hierarchy knowledge (c-graph), uniform correlation value (u-graph), and random correlation value (r-graph).

| Graph | Novel | All |
|-------|-------|-----|
|       | 1     | 2   | 5   | 10  | 1   | 2   | 5   | 10  |
| u-graph | 53.4 | 67.4 | 77.8 | 81.5 | 63.8 | 73.3 | 80.3 | 82.9 |
| r-graph | 54.4 | 67.4 | 77.8 | 81.9 | 64.5 | 73.3 | 80.5 | 83.2 |
| c-graph | 60.1 | 69.4 | 78.1 | 82.1 | 67.0 | 74.4 | 80.7 | 83.3 |
| s-graph | 62.1 | 70.9 | 78.4 | 82.3 | 68.3 | 75.2 | 80.8 | 83.5 |

Table 3: Top5-accuracy on ImageNet2012/2010 dataset.

| Method          | Novel | All |
|-----------------|-------|-----|
|                 | 1     | 2   | 3   | 4   | 5   |
| ImageNet2012/2010 |      |     |     |     |     |
| NN              | 34.2  | 43.6 | 48.7 | 52.3 | 54.0 |
| SGM (Harisharan and Girshick 2017) | 31.6 | 42.5 | 49.0 | 53.5 | 56.8 |
| PPA (Qiao et al. 2018a) | 33.0 | 43.1 | 48.5 | 52.5 | 55.4 |
| LSD (Douze et al. 2018) | 33.2 | 44.7 | 50.2 | 53.4 | 57.6 |
| KTCH (Li et al. 2019) | 39.0 | 48.9 | 54.9 | 58.7 | 60.5 |
| Ours            | 42.5  | 50.3 | 55.4 | 58.4 | 60.7 |

Figure 4: Top5-accuracy in “novel” and “all” metrics of our model that initializes $W_{base}$ with random values (Ours-random) and with the parameters pre-trained at the first training stage (Ours-pretrained).

Table 4: Top5-accuracy on ImageNet2012/2010 dataset.

| Method          | Novel | All |
|-----------------|-------|-----|
|                 | 1     | 2   | 3   | 4   | 5   |
| ImageNet2012/2010 |      |     |     |     |     |
| NN              | 34.2  | 43.6 | 48.7 | 52.3 | 54.0 |
| SGM (Harisharan and Girshick 2017) | 31.6 | 42.5 | 49.0 | 53.5 | 56.8 |
| PPA (Qiao et al. 2018a) | 33.0 | 43.1 | 48.5 | 52.5 | 55.4 |
| LSD (Douze et al. 2018) | 33.2 | 44.7 | 50.2 | 53.4 | 57.6 |
| KTCH (Li et al. 2019) | 39.0 | 48.9 | 54.9 | 58.7 | 60.5 |
| Ours            | 42.5  | 50.3 | 55.4 | 58.4 | 60.7 |
[Yang et al. 2019] Yang, W.; Wang, X.; Farhadi, A.; Gupta, A.; and Mottaghi, R. 2019. Visual semantic navigation using scene priors. In International Conference on Machine Learning.

[Zhu, Mumford, and others 2007] Zhu, S.-C.; Mumford, D.; et al. 2007. A stochastic grammar of images. Foundations and Trends® in Computer Graphics and Vision 2(4):259–362.