Adaptive Prototypical Networks With Label Words and Joint Representation Learning for Few-Shot Relation Classification

Yan Xiao, Yaochu Jin, Fellow, IEEE, and Kuangrong Hao

Abstract—Relation classification (RC) task is one of fundamental tasks of information extraction, aiming to detect the relation information between entity pairs in unstructured natural language text and generate structured data in the form of entity–relation triple. Although distant supervision methods can effectively alleviate the problem of lack of training data in supervised learning, they also introduce noise into the data and still cannot fundamentally solve the long-tail distribution problem of the training instances. In order to enable the neural network to learn new knowledge through few instances such as humans, this work focuses on few-shot relation classification (FSRC), where a classifier should generalize to new classes that have not been seen in the training set, given only a number of samples for each class. To make full use of the existing information and get a better feature representation for each instance, we propose to encode each class prototype in an adaptive way from two aspects. First, based on the prototypical networks, we propose an adaptive mixture mechanism to add label words to the representation of the class prototype, which, to the best of our knowledge, is the first attempt to integrate the label information into features of the support samples of each class so as to get more interactive class prototypes. Second, to more reasonably measure the distances between samples of each category, we introduce a loss function for joint representation learning (JRL) to encode each support instance in an adaptive manner. Extensive experiments have been conducted on FewRel under different few-shot (FS) settings, and the results show that the proposed adaptive prototypical networks with label words and JRL has not only achieved significant improvements in accuracy but also increased the generalization ability of FSRC.

Index Terms—Adaptive prototypical networks, few-shot learning (FSL), relation classification (RC).

I. INTRODUCTION

As the core task of text information extraction, relation classification (RC) is essential to text understanding, which has found many applications in natural language processing (NLP), such as text mining, question answering, and knowledge graphs (KGs). The main purpose of RC is to generate relational triples \((e_1, r, e_2)\) from plain natural language text, where \(e_1\) and \(e_2\) are entities and \(r\) implies the relation between the two entities in a sentence. For example, the instance “Beijing is the capital of China” expresses the relation “capital of” between the two entities “Beijing” and “China.”

Since convolutional neural network (CNN)-based methods [1] that automatically capture relevant lexical and sentence level features were first proposed for supervised RC tasks, conventional methods for RC have been gradually replaced by neural network-based methods. However, as neural networks are data-hungry, the performance of supervised RC methods heavily depends on the quality and quantity of the training data, meaning that these methods suffer from the lack of large-scale training datasets. To address this issue, Mintz et al. [2] proposed distant supervision relation extraction (DSRE) that can automatically generate training data by aligning triples in KGS with sentences in corpus. Although it can effectively alleviate the problem of manually labeling large-scale datasets, DSRE also inevitably introduces noise into the training data due to the strong hypothesis it makes. Therefore, most current research has been devoted to solving this problem through various methods such as multi-instance learning and attention mechanisms.

However, there might be still a large number of categories with very few samples in DSRE tasks, resulting in a clear long-tail distribution in the entire dataset that will dramatically deteriorate the classification performance. In real life, on the other hand, humans are able to learn from a few examples or even a single sample [3] since we can reason by analogy. Inspired by this, researchers have recently introduced the concept of few-shot learning (FSL), which was first proposed in computer vision [4], [5], into RC tasks to address the long-tail distribution problem, where the main challenge of FSL is to produce models that can generalize from a small amount labeled data to understand new categories. By a new category, we mean here the category to which the test sample belongs and has never appeared during the training.

In the field of NLP, Han et al. [6] first introduced FSL to the relation extraction task and constructed a few-shot relation classification (FSRC) dataset, called FewRel, aiming to predict the relation for a pair of entities in a sentence by training with a few labeled examples in each relation. Table I shows a part of the dataset, which is an episode of data in
three-way two-shot scenarios. In the left column, two parts of label information are shown for each category, namely the specific label words and label description (enclosed in parentheses). The purpose of the FSRC task is to determine which category the query instance of the last line belongs to. Currently, the most typical and effective approach to FSRC is the prototypical networks [7], which are based on the assumption that there exists a prototype for each class. The class prototype is obtained by averaging the vector of a few support instances, and then, the query is classified to the nearest class after calculating the distances between the query and each class prototype. However, under the \( N \)-way \( K \)-shot setting of the FSL, the available information contained in only a few samples is limited, and an inaccurate representation of each class prototype may lead to performance degradation.

In this work, to maximize the advantage of existing data and learn a more expressive embedding for each class prototype, we design adaptive prototypical networks with label words and joint representation learning (APN-LW-JRL) based on metric learning for FSRC, which performs classification by calculating the distances in the learned metric space. On the one hand, we augment the metric-based prototypical networks with an adaptive mixture mechanism to get the class prototype by incorporating the label information. To be specific, not only the support samples from each category but also the label words of each category are considered in an adaptive convex combination of two homologous embedding spaces. In this way, the label information is effectively utilized and combined to create high-quality class prototypes, which is helpful for solving FSRC tasks. On the other hand, a representation learning loss function is proposed to be combined with the classification loss function to form a joint training algorithm, enabling to learn an effective representation of each sample and obtain a more accurate class prototype. The purpose of the representation learning loss is to make the distance between samples of the same category much greater than that of different types in a space, which is inspired by the margin triplet loss in the face recognition task [8]. Consequently, not only the problem of feature sparsity is effectively alleviated but also the performance degradation of models with fewer samples can be substantially improved, thereby improving the classification performance. Moreover, in order to fully verify the effectiveness of our proposed model, we conducted several experiments on the FewRel dataset, which is derived from real-world corpus Wikipedia and Wikidata. Experimental results demonstrate that our model achieves significant performance over the state-of-the-art baseline methods.

To sum up, the main contributions of this work are as follows.

1) We propose an adaptive label information mixture mechanism based on the prototypical networks. To the best of our knowledge, this is the first time that information of each class has been integrated into the class prototypes for FSRC, which effectively alleviates the problem of model performance degradation due to data paucity.

2) A new joint learning method is designed for FSRC tasks, which allows the model to perform effective classification training assisted by good representation ability, thereby improving the generalization ability of the FSRC model with limited training instances.

3) Experiments are conducted to compare the proposed algorithm with its ablation variants as well as with the state-of-the-art. Our comparative results show that the two mechanisms we introduced both play an important role for the proposed model to deliver significant performance gains over the state-of-the-art baseline methods.

The remainder of this article is structured as follows. In Section II, we review the related work on relation extraction, FSL, and the triplet loss function adopted in this work. Section III presents the specific definition of the FSRC task. In Section IV, we give the details of our proposed model. The dataset and implementation details are described in Section V, and we further analyze our model and compare it with state-of-the-art models in Section VI. Conclusions and future work are summarized in Section VII.

II. RELATED WORK

In this section, we briefly review the relevant background of RC, FSL, and the triplet loss function.

A. Relation Classification

Different from the joint entity and relation extraction task [9], RC is usually performed after the named entity recognition
to transfer the latent knowledge of the previously learned realizations by fine-tuning the weights of a pretrained network learned from basic classes with sufficient samples. It is mainly a method for FSL [33], [34] in early years, which aims to develop models for disease prediction tasks and SAMIE [31] for question answering tasks. For specific tasks, several FS models based on semisupervised learning were developed, such as AffinityNet [30] for disease prediction tasks and transformer [17], have constantly refreshed the record of feature learning history, producing remarkable theoretical and empirical results for classification and regression problems [45]. Its desirable benefits include its better generalization properties and robustness to input perturbations. A large body of research [46]–[48] has explored the benefits of encouraging a large margin in the context of deep networks, which was

\textbf{B. Few-Shot Relation Classification}

FSL [22] has recently attracted increasing attention, as it enables to generalize to new classes in classification tasks by training a classifier with only a handful of labeled samples. FSL was proposed to imitate the human’s ability of obtaining knowledge from a small number of examples through generalization and analogy. Among various techniques, data augmentation is the most straightforward method to solve the FSL problem with prior knowledge, which intuitively increases the number of training samples and enhances data diversity [23]. Based on whether extra supervised information is used, the augmentation methods can be divided into supervised paradigms [24], [25] and unsupervised paradigms [26], [27]. Ren et al. [28] solved the FS classification problem from the perspective of semisupervised, where unlabeled samples are also available within each episode. Moreover, a semisupervised meta-learning method called learning to self-train (LST) [29] was proposed to learn to cherry-pick and label the unsupervised data for performance improvement. In addition, several FS models based on semisupervised learning for specific tasks were recently proposed, such as AffinityNet [30] for disease prediction tasks and SAMIE [31] for question answering tasks. Apart from data augmentation, knowledge transfer [32] between associative categories is an important method for FSL [33], [34] in early years, which aims to learn new classes more quickly by making use of knowledge learned from basic classes with sufficient samples. It is mainly realized by fine-tuning the weights of a pretrained network to transfer the latent knowledge of the previously learned category to the target domain. Based on a similar principle, the recently proposed meta-learning method aims to solve new learning tasks only with few samples by training models on a variety of learning tasks. The simple neural attentive learner (SNAIL) meta-learning model [35] uses the time CNN and an attention mechanism to quickly learn from previous experience. The meta-network (MetaNet) [36] was proposed to learn metaknowledge across tasks and fast parameterize underlying neural networks for rapid generalizations. Finn et al. [37] proposed a model-agnostic meta-learning (MAML) algorithm for meta-learning that is model-agnostic to be compatible with any model trained with the gradient descent. Meanwhile, the metric-based FSL has attracted a lot of interest due to its simplicity and effectiveness, the basic idea of which is to model the distance distribution among all samples in the entire dataset and then classify them according to different metrics. Based on this idea, the Siamese neural network was proposed in [4] that employed a unique structure to learn a similarity metric from the data, while Vinyals et al. [5] explored matching networks with the idea of attention and external memories to map the data to obviate the need for fine-tuning when adapted to new class types. Snell et al. [7] defined the prototypical networks that usually classify the sample and its nearest class prototype into one category, where the class prototype is obtained by averaging all samples in each category. Moreover, a generalization of the above three models is proposed in [38], where a graph neural network architecture is utilized to process general information in FSL tasks.

Although most existing FS methods were developed in the CV field [39]–[41], their success has inspired researchers to explore the application of FSL to NLP. With respect to the FSRC, Han et al. [6] organized the FewRel dataset and adapted part of the methods above on FSRC, where the prototypical networks are pretty easy to implement and understand. Afterward, hybrid attention-based prototypical networks for noisy FSRC are proposed in [42] and hierarchical attention prototypical networks for FS text classification are proposed in [43]. To improve the conventional prototypical networks and maximize the role of existing data for the FSRC task, this work proposes an adaptive label information mixture mechanism with the joint representation learning (JRL) inspired by the triplet loss to improve the generalization of the FS model. The main motivation is to learn a metric space, which not only accurately expresses the class prototype of each category but also maps similar samples closer and dissimilar ones more distant in the metric space, thereby facilitating classification.

\textbf{C. Triplet Loss in Face Recognition}

Representation learning [44], also known as feature learning, generally refers to the model that automatically learns the input data to obtain the features that are more representative. The large margin principle has played a key role in the course of feature learning history, producing remarkable theoretical and empirical results for classification and regression problems [45]. Its desirable benefits include its better generalization properties and robustness to input perturbations. A large body of research [46]–[48] has explored the benefits of encouraging a large margin in the context of deep networks, which was
realized by defining the following triplet loss function [8]:

\[ L = \sum_i^N \left[ \|f(x_i^p) - f(x_i^n)\|_2^2 - \|f(x_i^n) - f(x_i^q)\|_2^2 + \eta \right]_+. \quad (1) \]

According to (1), given an image \( x_a^i \) of a specific person as anchor, the triplet loss optimizes the embedding space by adding a margin \( \eta \) in the objective function such that data points \( x_p^i \) with the same identity (positive) are closer to each other than those with different identities \( x_n^i \) (negative), which allows us to perform end-to-end representation learning between the input image and the desired embedding space. However, generating all possible triplets will not contribute to the training but result in a slower convergence instead. Therefore, an essential part of the triplet loss is the mining of triplets, and a variant of the triplet loss [49] was utilized to implement deep metric learning in person reidentification task afterward. It outperformed most other methods by a large margin for models trained from scratch as well as pretrained ones. Similarly, we also have made series of modifications to triplet loss and finally got the effective combination of its application in FSRC, which we will introduce in detail in Section IV-C2.

III. TASK DEFINITION

Suppose that \( D \) is the entire labeled data, which is usually divided into two parts according to the relation category: \( D_{\text{train}} \) and \( D_{\text{test}} \). In other words, the two datasets have different label spaces and they are disjoint with each other. Generally, most of the current FSL algorithms employ the “episode” training strategy [5], which has been proven to be practical. To this end, \( D_{\text{test}} \) is further split into two parts: \( D_{\text{test-support}} \) and \( D_{\text{test-query}} \). To be specific, in each episode, \( N \) classes are randomly sampled from \( D_{\text{test}} \) first, and then, \( K \) instances for each of \( N \) classes are sampled to constitute \( D_{\text{test-support}} = \{(x_p^i, y_p^i); i = 1, \ldots, N, j = 1, \ldots, K\} \). We usually call it \( N \)-way \( K \)-shot in FSL task. Besides, we randomly sample \( R \) instances from the remaining samples of those \( N \) classes to construct \( D_{\text{test-query}} = \{(x_q^i, y_q^i); k = 1, \ldots, R\} \). Likewise, we can also acquire \( D_{\text{train-support}} \) and \( D_{\text{train-query}} \) in a similar way, but \( N_{\text{for train}} \) is usually larger than \( N_{\text{for test}} \), which is a common practice in FSL. A data example of a three-way two-shot scenario of FSRC is given in Table I, where \( N = 3, K = 2, \) and \( R = 1 \).

As a result, FSRC can be defined as a task to predict the relation \( y_q \) of a query instance \( x_q \) in the query set \( Q \), given a relation set \( R = \{r_i; i = 1, \ldots, N\} \) and the support set \( S \). As we consider \( N = 5 \) or 10 and \( K = 5 \) or 10 in this work, the number \( N \) of classes and the number \( K \) of instances are usually so small that the FS model has to be trained from the few instances in the support set and then use the trained model to predict the relation for any given query instance. In addition, each instance in the dataset is composed of \( T \) words, which includes the mentioned entity pair \((h, t)\).

IV. METHOD

In this section, we present the details of the proposed APN-LW-JRL for FSRC. As shown in Fig. 1, all the input samples, both in the support set and in the query set, will be embedded to feed into the encoder to obtain their feature vector, while the label words are only expressed through the embedding layer. Then, the class prototype can be attainable via inputting the represented samples in each support set into the prototypical networks. After that, we will incorporate the embedded label words into the class prototype by our proposed adaptive mixture mechanism to make improvements. Finally, we will explain the specific JRL method that we proposed, in an effort to perform effective end-to-end learning between input samples and output prediction results.

A. Instance Encoder

The instance encoder mainly consists of two layers: embedding layer and instance encoding layer. The former aims to vectorize the input text, whereas the latter is meant to perform feature extraction on the vectorized text.

1) Embedding Layer: Given an instance \( x = \{w_1, w_2, \ldots, w_T\} \) with \( T \) words mentioning two entities \((h, t)\), each word is represented by a real-valued vector \( w_i \) via the pretrained embedding matrix \( V_p \in \mathbb{R}^{d_w \times |V|} \), where \(|V|\) denotes the size of corpus vocabulary \( V \) and \( d_w \) means the dimension of word embedding. Furthermore, the positional features of each word in the sentence, i.e., the order of each word appears in the sentence, are very crucial for the extraction of text information. Especially in the task of RC, the closer the words are to the entities, the more informative the words are. For this reason, the position embeddings [1] are introduced in our embedding layer, which we used to transform the relative distances between each word in the sentence and the two entities into vectors generated by looking up a randomly initialized position embedding matrix \( V_p \in \mathbb{R}^{d_p \times |V|} \).

Finally, an ultimate input embedding for each word is achieved by concatenating word embedding and position embedding that can be denoted as \( e_i = [w_i; p_{ui}; p_{vi}] \), where \( e_i \in \mathbb{R}^d \) and \( d = d_w + d_p \times 2 \).

2) Encoding Layer: In this work, CNN is adopted as our context encoder to obtain the hidden embedding of each word since it is very popular in most NLP tasks. After a convolution kernel with a window size \( u \) acted on the input embedding \([e_1, \ldots, e_T]\), the hidden annotations will be as follows:

\[ h_i = \text{CNN}(e_i; \mathcal{W}), \quad i = 1, \ldots, T. \quad (2) \]

Then, to determine the most useful feature in each dimension of the feature vectors, we perform a max-pooling operation on these hidden annotations:

\[ x = \max(h_1, \ldots, h_T). \quad (3) \]

Finally, we define the above two layers as a comprehensive function in (4) for simplicity, where \( \phi \) is the networks parameters to be learned

\[ x = f_\phi(x). \quad (4) \]
B. Prototypical Networks

The prototypical networks are simpler and more efficient than the meta-learning approaches in the FSL task, and they produce the state-of-the-art results even without sophisticated extensions developed for matching [7]. It is based on the idea that each class can be represented by means of its examples in a representation space learned by a neural network, which is expressed by (5). $p_i$ is defined as the class prototype for relation $r_i$ and $x^s_{ij}$ is the embedding of the $j$th sentence of relation $r_i$, where $s$ means the support set and $K$ is the number of sentences included for each class

$$p_i = \frac{1}{K} \sum_{j=1}^{K} f_\theta(x^s_{ij}).$$

After the prototypes for classes are calculated from supporting instances, we can classify a query instance by comparing the distance between the query instance and each prototype based on a certain distance metric. Actually, given a distance function $D(\cdot, \cdot)$, a distribution over classes is produced by prototypical networks for a query point $x^q$ based on a softmax function over distances to the prototypes in the embedding space. $|\mathcal{R}|$ denotes the total categories in the relation set

$$p_\theta(y = r_i|x^q) = \frac{\exp(-D(f_\theta(x^q), p_i))}{\sum_{r \in |\mathcal{R}|} \exp(-D(f_\theta(x^q), p_i))}.$$

As indicated in [7], there are multiple choices for the distance metric, among which the Euclidean distance is the most appropriate. Hence, we also adopt the Euclidean distance in this work.

C. Adaptive Prototypical Networks

In addition to their attractive simplicity, another reason why metric-based FSL approaches are so popular is that they are able to learn a good embedding space, where samples from the same class are clustered together, while samples from different classes are far away from each other. Afterward, a new sample from the unseen category can be recognized directly through the distance metric within the learned embedding space. Upon that, it is very crucial to learn a discriminative embedding space in metric-based FSL tasks. In this section, we will interpret in detail how we perform adaptive technologies based on the prototypical networks by means of adaptive label information mixture mechanism and adaptive joint training for class prototype.

1) Adaptive Label Information Mixture Mechanism: As prototypical networks classify a query by calculating the distance between the mapping of each category, the sample distribution learned by the network model will not be really satisfactory when the two categories are particularly similar, which may affect the classification performance.
XIAO et al.: APN-LW-JRL FOR FEW-SHOT RELATION CLASSIFICATION

Fig. 2. Illustration of the proposed adaptive mixture mechanism. Suppose that there is a query sample belonging to category $i$. (a) Originally, the prototype closest to the query sample $q$ is $p_j$. (b) Take the semantic feature of label words into account. (c) Location distribution of the class prototype has modified through the implementation of the mixture mechanism. (d) After the location distribution is updated, the closest prototype to the query is now the category $i$, correcting the classification.

Thus, to make the spatial mapping of different categories very distinctive, we hypothesize that semantic information of both samples and labels can be useful to make full use of the sample resources for FSL. Moreover, it is desirable to adaptively express the class prototypes, given different scenarios of categories. As shown in Fig. 2, we augment prototypical networks to model the new prototype representation as a convex combination of two pieces of information. Specifically, for each category $i$, we integrate the label information $c_i$ of each category on the basis of the original class prototype $p_i$, and then, the new prototype is computed as follows:

$$p_i' = \lambda_i \cdot p_i + (1 - \lambda_i) \cdot c_i$$  \hspace{1cm} (7)

where $\lambda_i$ is the adaptive mixture coefficient that depends on the representation of category information to ensure that the calculated class prototype is compatible and not redundant. $\lambda_i$ is calculated according to (8), where an adaptive mixing network $h(\cdot)$, such as a fully connected network, is used to infer the coefficient such that similar classes in the feature embedding space are better separated

$$\lambda_i = \frac{1}{1 + \exp(-h(c_i))}$$  \hspace{1cm} (8)

$$c_i = g(w_i)$$  \hspace{1cm} (9)

where $c_i$ refers to the embedding of label words for class $i$, which is obtained by calculating the mean of each column of the pretrained embedding matrix. The transformation is defined as $g(\cdot)$. In addition, we also use label description instead of label words in the proposed adaptive mixture mechanism, which will be further discussed in Section VI-C.

2) Adaptive Joint Learning for Class Prototype: The key issue of FSL is to learn to generalize and the performance of prototypical network for FSRC largely depends on the spatial distribution of sentence embeddings. To realize the full potential of metric-based methods for FS, we propose an adaptive joint learning method based on the margin principle to learn a more discriminative metric space and improve the generalization capacity of model as well. This is achieved by combining the objective functions of representation learning and classification to get better feature representations for each instance. In this way, the model is able to separate samples from different categories as far as possible in the metric space

$$L_{\text{representation}} = \sum_{a=1}^{[S]+[Q]} \left[ m + D_{\text{max}}^{a,p} - D_{\text{min}}^{a,n} \right]_+$$  \hspace{1cm} (10)

$$L_{\text{Cross-Entropy}} = -\sum_{k=1}^{[Q]} \log(p_\theta(y = r_i|x^q_k))$$  \hspace{1cm} (11)

$$L_{\text{Joint}} = L_{\text{Cross-Entropy}} + \alpha \cdot L_{\text{representation}}.$$  \hspace{1cm} (12)

Specifically, based on the attributes of the FSL tasks, we use the triple loss objective function introduced in Section II-C as the objective function of representation learning, which consists of triplets of the anchor, the positive, and the negative. Here, we use the Euclidean distance as our distance metric, denoted as $D$. For the triplet loss in the FSRC task, each sample of the support and query set in each episode will be defined as an anchor, to which the farthest instance with the same label (denoted as $D_{\text{max}}^{a,p}$) and the nearest instance with different labels (denoted as $D_{\text{min}}^{a,n}$) are defined as the positive...
and the negative, respectively. This is one main difference between the proposed method and the vanilla method, which is called the hardest triplets. In (10), $|S| + |Q|$ is the total triplets in each episode, and $m$ is a hyperparameter that controls the interval between categories. In Section VI-C, we further analyze the situation in which the triplet is composed of an anchor and its related class prototype. By combining the classification objective function and the representation learning objective function as shown in (12), our model is able to consider both classification error and representation quality at the same time during the training. Also, $\alpha$ is a hyperparameter controlling the weight of representation learning in the overall objective function.

D. Overall Framework

The overall framework of the proposed algorithm is summarized in Algorithm 1. After the formulation of the FSRC task into the $N$-way $K$-shot setting, i.e., the input sample preprocessing, as described in Section III, the proposed algorithm first uses the instance encoder to represent the support and query samples, and then, the support prototype can be obtained through the prototype network. Next, the proposed algorithm calculates the class prototype based on the proposed adaptive label information mixture mechanism and then calculates the representation objective function and the classification objective function according to the distance between the query samples and the class prototypes. Recall that the adaptive prototypical networks are composed of two parts: adaptive label information mixture mechanism and adaptive joint learning for class prototype. In other words, adaptive prototypical networks are proposed from two adaptive aspects on the basis of prototypical networks. Finally, the joint objective function in each episode is minimized to train the whole model.

V. EXPERIMENTAL SETTINGS

We describe the experimental settings in this section, beginning with introducing the dataset and learning configurations, followed by detailed implementation details of the models.

A. Dataset and Learning Configurations

In this work, we evaluate our model on the FewRel dataset [6], the only public FSRC dataset, which is first generated by distant supervision and then filtered by crowdsourcing to remove noisy annotations. It consists of 70,000 instances on 100 relations derived from Wikipedia, and each relation includes 700 instances. Also, both head and tail entities are marked in each sentence. Note, however, that only 80 relations are available, of which 48 are used for the training set, 12 for the verification, and 20 for test in the following comparative studies. In our experiments, we investigate four FSL configurations, namely five-way one-shot, five-way five-shot, ten-way one-shot, and ten-way five-shot, which are the same for the proposed algorithm and the compared baselines. All results are averaged over ten independent runs.

| Algorithm 1 Framework of the APN-LW-JRL for a Training Episode in FSRC |
|-----------------------------|-----------------------------|-----------------------------|
| **Input:** Support set $S = \{(x^i_{ij}, y^i_j)\}$, query set $Q = \{(x^q_i, y^q_i)\}$, class-words $w^i$, $i = 1, \ldots, N$, $j = 1, \ldots, K$, $R = \{r_i; i = 1, \ldots, N\}$, $N \times K = |S|$, $k = 1, \ldots, |Q|$. |
| **Output:** The joint learning objective function $L_{Joint}$ in an episode. |
| 1: Obtain the vector representation of each support samples $x^i_{ij}$ in class $i$ and query samples $x^q_i$ by Eq. (4); |
| 2: Obtain the support prototype for class $i$ by feeding the support instances of the $i$-th category into the prototypical networks by Eq. (5); |
| 3: Calculate the class prototype with the semantic vector for class $i$ in Eq. (7); |
| 4: Calculate the representation objective function by finding the farthest positive instance and the nearest negative instance to each sample in the support set and query set in Eq. (10); |
| 5: Calculate the probability of each query belonging to class $i$ according to the Eq. (6) and the classification objective function in Eq. (11); |
| 6: Obtain the final objective function $L_{Joint}$ for an episode by Eq. (12); |
| 7: Return $L_{Joint}$ to be minimized to train the model. |

| TABLE II HYPERPARAMETERS OF THE MODELS BUILT IN OUR EXPERIMENTS |
|-----------------------------|-----------------------------|-----------------------------|
| **Component** | **Parameter** | **Value** |
| Embedding | Word embedding dimension $d_e$ | 50 |
| | Position embedding dimension $d_p$ | 5 |
| Encoder | Convolutional Window Size $w$ | 5 |
| | Max length $T'$ | 40 |
| Joint loss | Margin $\gamma$ | 0.5 |
| | Alpha $\alpha$ | 1 |
| Optimization | Initial Learning Rate | 0.1 |
| | Weight Decay | 10-3 |
| | Dropout rate | 0.2 |

B. Implementation Details

All hyperparameters of each part of our model are listed in Table II. We tune the rest hyperparameters of all models by grid search using the validation set, and all models are trained on the training set. Finally, the models achieving the best validation performance are saved to be tested on the test set. It has been found in [36] that feeding more classes to the models may achieve better performances than using the same configurations at both training and testing stages. Accordingly, we set $N = 20$ to construct the support set in each training episode. Actually, the training episode is built by first randomly selecting a subset of relations from the training set and then sampling a subset of instances within each selected relation to build the support set, and the remainder is sampled to build the query set.

For fair comparisons, the settings of the embedding and encoding layers are consistent with the compared models. We use the same word embeddings pretrained by Glove [51]...
consisting of 6B tokens and 400k vocabularies, and the dimensions of the word vector and position vector are set to 50 and 5, respectively. According to the definition of the encoding layer, the final embedding dimension of a word is 60 and we define the maximum length of each instance as 40. After each instance is represented by the embedding layer, we use CNN as an encoder to extract features in the encoding layer, where the parameters of each convolution kernel are trained through backpropagation. Since the label words used in the adaptive label information mixture mechanism are a phrase composed of several words, we use the same word vector and perform the transformation described in Section IV-C1. In the comparison, the label description is treated as an instance, but the relative position vector is not considered when being inputted into the embedding and encoding layers because there are no entities in the label description. All experiments are based on the N-way K-shot setting and we employ the mini-batch stochastic gradient descent (SGD) to solve the optimization problem.

VI. RESULTS AND ANALYSIS

In this section, we will demonstrate the effectiveness of the proposed APN-LW-JRL on the FSRC task, followed by ablation studies. Finally, the visualization of sentence embeddings, the extension of support set, and case study are presented to show the validity of the proposed model.

A. Compared Algorithms

To verify the effectiveness of APN-LW-JRL, we compare it with the following most state-of-the-art algorithms for FSRC.

1) MetaNet [36]: A novel meta-learning method that combines slow weights and fast weights for prediction.
2) GNN [38]: A graph neural network that encodes each instance in support set and query set as a node in the graph for FSL.
3) SNAIL [35]: A meta-learning model that uses temporal CNNs and attention mechanisms to quickly learn from past experience.
4) ProNet [7]: A model based on metric learning, which assumes that each class can be represented by a prototype.
5) Pro-HATT [42]: A hybrid attention-based prototypical network consists of instance- and feature-level attention schemes.

B. Comparative Results

The comparative results are presented Table III. From these results, we can make the following observations.

1) APN-LW-JRL outperforms ProNet under the four different FSL settings, which is a simple but powerful baseline for FSL and is also the basis of our model. Comparing ProNet and APN-LW, APN-LW, and APN-LW-JRL, we can clearly see the effectiveness of introducing label words and joint training of classification and representation.

2) Strikingly, APN-LW-JRL also achieves better results compared with the other prototypical network-based models, Pro-HATT, verifying the feasibility of considering prototypical networks in an adaptive way, even better than integrating various attention mechanisms. On the one hand, the adaptive label information mixture mechanism proposed in this work can leverage the information advantages of two aspects and adjust its focus accordingly. On the other hand, benefiting from adaptive joint training for class prototype, our model allows to perform end-to-end learning between the input instances and the desired embedding space.

3) APN-LW-JRL has shown significantly better results than MetaNet, GNN, and SNAIL, which are the state-of-the-art learning models. This indicates that APN-LW-JRL is effective and the two adaptation mechanisms we introduced both make contributions to improve the performance.

C. Effects of Adaptive Prototypical Networks

In this section, ablation studies were conducted to evaluate the contributions of the individual model components. Variants of the proposed algorithm are designed to verify the respective function of the adaptive label information mixture mechanism and adaptive JRL for class prototype. In addition, we also discuss the influence of parameter m on model performance in the JRL.

1) APN-LD: The proposed adaptive label information mixture mechanism based on label description.
2) APN-LD-JRL: The proposed adaptive label information mixture mechanism based on label description with adaptive JRL based on the sample triplets.
3) ProNet-JRL: ProNet [7] model with the proposed adaptive JRL based on the sample triplets.
4) ProNet-PJRL: ProNet [7] model with the proposed adaptive JRL based on the prototype triplets.
5) APN-LW: The proposed adaptive label information mixture mechanism based on label words.
6) APN-LW-PJRL: The proposed adaptive label information mixture mechanism based on label words with adaptive JRL based on the prototype triplets.
7) APN-LW-JRL: The proposed adaptive label information mixture mechanism based on label words with adaptive JRL based on the sample triplets.

Table IV shows the performance of our model and its ablations on the divided FewRel test set. By comparing the three models of ProNet, APN-LD, and APN-LW, we can find that the adaptive label information mixture mechanism exhibits a positive influence on the expression of class prototypes. Here, LD denotes label description and its representation is obtained by the same encoder as the instance. Comparing APN-LD and APN-LW, we find that LD, which is much longer than LW, is not as good as LW. This is contrary to the intuitive belief that the longer the text, the richer information it contains. We take a closer look at the dataset and find that this might be attributed to the fact that the descriptions of some categories are too metaphysical. It may cause deviation
TABLE III
ACCURACIES (%) OF DIFFERENT MODELS ON THE DIVIDED FEWREL TEST SET UNDER FOUR DIFFERENT SETTINGS. HERE, APN STANDS FOR ADAPTIVE PROTOTYPICAL NETWORKS, LW STANDS FOR LABEL WORDS, AND JRL STANDS FOR JRL.

| Model     | 5 Way 1 Shot | 5 Way 5 Shot | 10 Way 1 Shot | 10 Way 5 Shot |
|-----------|--------------|--------------|---------------|---------------|
| Meta Network | 70.89 ± 0.64 | 71.02 ± 0.45 | 42.01 ± 0.12 | 43.87 ± 0.32 |
| GNN       | 67.30 ± 0.91 | 74.47 ± 0.61 | 42.84 ± 0.08 | 69.63 ± 0.21 |
| SNAIL     | 66.28 ± 0.24 | 78.07 ± 0.12 | 53.72 ± 0.10 | 62.88 ± 0.33 |
| ProNet    | 70.63 ± 0.55 | 85.10 ± 0.22 | 57.52 ± 0.91 | 74.74 ± 0.11 |
| Pro-HATT  | 71.76 ± 0.63 | 86.01 ± 0.33 | 59.22 ± 0.21 | 75.21 ± 0.32 |
| APN-LW    | 72.57 ± 0.44 | 86.29 ± 0.15 | 59.44 ± 0.35 | 76.32 ± 0.05 |
| APN-LW-JRL | 73.45 ± 0.83 | 87.27 ± 0.55 | 61.02 ± 0.61 | 77.69 ± 0.22 |

TABLE IV
ACCURACIES (%) OF ALL MODELS FOR ABLATION STUDY ON THE DIVIDED FEWREL TEST SET UNDER FOUR DIFFERENT SETTINGS.

| Model     | 5 Way 1 Shot | 5 Way 5 Shot | 10 Way 1 Shot | 10 Way 5 Shot |
|-----------|--------------|--------------|---------------|---------------|
| APN-LD    | 71.78 ± 0.31 | 85.82 ± 0.20 | 59.20 ± 0.11 | 75.70 ± 0.41 |
| APN-LW    | 72.57 ± 0.44 | 86.29 ± 0.15 | 59.44 ± 0.35 | 76.32 ± 0.05 |
| APN-LD-JRL | 73.18 ± 0.22 | 86.97 ± 0.31 | 60.67 ± 0.52 | 77.09 ± 0.16 |
| ProNet    | 70.63 ± 0.55 | 85.10 ± 0.22 | 57.52 ± 0.91 | 74.74 ± 0.11 |
| ProNet-PJRL | 71.63 ± 0.35 | 86.20 ± 0.72 | 58.92 ± 0.15 | 76.23 ± 0.66 |
| APN-LW-PJRL | 72.56 ± 0.11 | 86.55 ± 0.56 | 59.80 ± 0.22 | 76.99 ± 0.61 |
| ProNet-JRL | 72.98 ± 0.23 | 86.62 ± 0.45 | 60.25 ± 0.12 | 76.43 ± 0.35 |
| APN-LW-JRL | 73.45 ± 0.83 | 87.27 ± 0.55 | 61.02 ± 0.61 | 77.69 ± 0.22 |

Fig. 3. Effect of $m$ in the JRL on the accuracy, where the left is the five-way one-shot learning, while the right is five-way five-shot learning.

or redundancy when the description is used to represent the label information. For example, the LW “residence” is more idiographic than the LD “the place where the person is or has been,” which helps to get a better class prototype. At the same time, the result of comparing APN-LD-JRL and APN-LW-JRL can also be used as convincing evidence in support of this argument.

Similarly, we also compare ProNet and ProNet-JRL and APN-LW and APN-LW-JRL, respectively, from which we can conclude that the adaptive JRL method we proposed demonstrates an excellent end-to-end deep metric learning capability. PJRL here means that for each sample as an anchor, the farthest class prototype with the same label is chosen as positive, while the nearest class prototype with a different label is chosen as negative to form the triplets. Clearly, the generalization performance of selecting the triplets with the class prototype as the target is not as strong as the sample targeting because the selection of samples to form the triplets is more extensive, which can be concluded by comparing with ProNet-PJRL and ProNet-JRL and APN-LW-PJRL and APN-LW-JRL. In addition, Fig. 3 shows the performance of the proposed algorithm when the margin $m$ changes from 0.1 to 0.9 at an interval of 0.2, which shows that the JRL function is relatively insensitive to $m$ and achieves the best performance when the value is 0.5.

Fig. 4. Visualization of sentence embeddings of a five-way-20-shot scenario in the divided FewRel test set, where one point represents one instance and different colors correspond to different categories.
Fig. 5. Comparison of performance of Proto, Pro-HATT, and APN-LW-JRL for different data extensions of the support set on the divided FewRel test set. Left: value of the way is fixed to 10 and the accuracy changes with the shot. Right: value of the shot is fixed to 5 and the accuracy changes with the way. Each small vertical line segment on the curve represents an error interval of multiple experiments.

### TABLE V

ACCURACIES (%) OF PROTO, PHATT, AND HAPN IN DIFFERENT DATA EXTENSIONS OF THE SUPPORT SET ON THE DIVIDED FEWREL TEST SET

| K   | 5   | 6   | 7   | 8   | 9   | 10  | 11  | 12  | 13  | 14  | 15  |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| ProtoNet  | 74.92 | 76.02 | 76.79 | 77.36 | 77.84 | 78.29 | 78.61 | 78.95 | 79.16 | 79.41 | 79.57 |
| Pro-HATT | 75.83 | 77.02 | 77.81 | 78.51 | 79.01 | 79.39 | 79.83 | 80.06 | 80.34 | 80.52 | 80.73 |
| APN-LW-JRL | 77.16 | 78.27 | 78.98 | 79.54 | 80.03 | 80.46 | 80.70 | 80.98 | 81.24 | 81.43 | 81.61 |

### TABLE VI

SOME SPECIFIC RESULTS OF RC EXAMPLES FROM THE FEWREL DATASET. THE HEAD ENTITY AND THE TAIL ENTITY ARE MARKED AS BLUE AND RED, RESPECTIVELY, IN EACH INSTANCE

| Label | Sentence                                                                 | ProNet | APN-LW | APN-LW-JRL |
|-------|---------------------------------------------------------------------------|--------|--------|------------|
| "P1001" | Macelod was a territorial electoral district for the Legislative Assembly of Northwest Territories, Canada. | Location of formation (False) | Applies to jurisdiction (True) | Applies to jurisdiction (True) |
| "P463" | South Africa is part of the IBSA Dialogue Forum, alongside Brazil and India. | Part of (False) | Member of (True) | Member of (True) |
| "P937" | She lives in Montreal and is the common - law partner of novelist Rawi Hage. | Residence (False) | Work location (True) | Work location (True) |
| "P355" | Hotel Indigo competes with Starwood’s W Hotels as well as Andaz Hotels, Aloft Hotels and Le Meridien Hotels. | Has part (False) | Subsidiary (True) | Subsidiary (True) |
| "P1435" | Hampton Hill, John Thompson House, Twin Trees Farm, and Willow Mill Complex are listed on the National Register of Historic Places. | Instance of (False) | Applies to jurisdiction (False) | Heritage designation (True) |

Furthermore, we visualize an episode set of sentences from the divided FewRel test set to gain more insight into the model performance. It can be seen from Fig. 4 that as the adaptive mixture mechanism and adaptive JRL play an increasingly important role, APN-LW-JRL is able to learn more discriminative sentence embeddings, making it more capable of distinguishing confusing data than those of the previous ablation models. We attribute this capability to the rich feature expression of the adaptive mixture mechanism and the moderate spatial embedding learning of adaptive JRL, which increases the distance between different classes and reduces the distance within the same class.

### D. Extension of Support Set

It is recognized that more support instances can provide more useful information to the prototype vector, but more noise may be added in as well. In this section, to study how the number of categories and samples in the support set will affect the performance of different models, we define an extension of the support set as the additive value of accuracy. First, K is changed from 5 to 15 when N to 10. Then, the range of N is also 5–15 when K is fixed to 5. As shown in Fig. 5 and Table V, given the same extension setting, the proposed model shows much better performance than other benchmarks. It also verifies that the way we increase the
data utilization is beneficial to improve the robustness and generalization capability of our model.

E. Case Studies

In order to better understand why the proposed model outperforms the compared ones, we selected a few examples from the divided FewRel test set and analyze their classification results. Table VI shows five instances on which ProNet has failed, while our model has correctly predicted. Take the second query instance in Table VI as an example, and ProNet predicts this incorrectly into “Part of,” while our model classifies accurately. This instance is quite challenging since the expressions of these two relations are very similar. From these instances, we can conclude that it is of great importance to update both the way of class prototype generation and the learning target of the original prototypical networks, as done in our model.

VII. CONCLUSION AND FUTURE WORK

Metric learning-based FSL task aims to learn a set of projection functions that take support and query samples from the target problem, which is heavily dependent on the quality of the embedding space. Therefore, to maximize the efficiency of existing information resources and achieve a better class prototype, we improve the prototypical networks in an adaptive manner, which consists of a label information mixture mechanism and a JRL method for FSRC task. It can appropriately weigh the importance of support samples and label information to make adaptive adjustments and learn a more discriminative prototype representation. Comparative studies have been conducted on FewRel under different FS settings, from which we can conclude that our proposed APN-LW-JRL is successful in improving the classification performance of FS RC models.

In this work, label information of the data is made use of to improve the performance of the proposed APN-LW-JRL and the proposed algorithm is tested on the FewRel dataset only. In the future, we will extend the model to zero-shot learning and validate its performance on additional text datasets. In addition, we are going to investigate the use of FSL for joint relation extraction of NER and RC, and its application to medical and other real-world problems.

REFERENCES

[1] D. Zeng, K. Liu, S. Lai, G. Zhou, and J. Zhao, “Relation classification via convolutional deep neural network,” in Proc. COLING 25th Int. Conf. Comput. Linguistics, Tech. Papers, 2014, pp. 2335–2344.
[2] M. Mintz, S. Bills, R. Snow, and D. Jurafsky, “Distant supervision for relation extraction without labeled data,” in Proc. Joint Conf. 47th Annu. Meeting ACL 4th Int. Joint Conf. Natural Lang. Process. (ACL-IJCNLP), vol. 2, 2009, pp. 1003–1011.
[3] S. Carey and E. Bartlett, “Acquiring a single new word,” Papers Rep. Child Lang. Develop., vol. 15, pp. 17–29, Aug. 1978.
[4] G. Koch, R. Zemel, and R. Salakhutdinov, “Siamese neural networks for one-shot image recognition,” in Proc. ICML Deep Learn. Workshop, vol. 2, 2015.
[5] O. Vinyals et al., “Matching networks for one shot learning,” in Proc. Adv. Neural Inf. Process. Syst., 2016, pp. 3630–3638.
[6] X. Han et al., “FewRel: A large-scale supervised few-shot relation classification dataset with State-of-the-Art evaluation,” 2018, arXiv:1810.10147. [Online]. Available: https://arxiv.org/abs/1810.10147
[7] J. Snell, K. Swersky, and R. Zemel, “Prototypical networks for few-shot learning,” in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 4077–4087.
[8] F. Schroff, D. Kalenichenko, and J. Philbin, “FaceNet: A unified embedding for face recognition and clustering,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2015, pp. 815–823.
[9] T. Nayak and H. T. Ng, “Effective modeling of encoder-decoder architecture for joint entity and relation extraction,” in Proc. AAAI Conf. Artif. Intell., vol. 34, no. 5, 2020, pp. 4528–4535.
[10] R. C. Buncescu and R. J. Mooney, “A shortest path dependency kernel for relation extraction,” in Proc. Conf. Hum. Lang. Technol. Empirical Methods Natural Lang. Process. (HLT), 2005, pp. 724–731.
[11] D. Zelenko, C. Aone, and A. Richardella, “Kernel methods for relation extraction,” J. Mach. Learn. Res., vol. 3, pp. 1083–1106, Feb. 2003.
[12] N. Kambhatla, “Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations,” in Proc. ACL Interact. Poster Demonstration Sessions, 2004, p. 22.
[13] P. Qin, W. Xu, and J. Guo, “An empirical convolutional neural network approach for semantic relation classification,” Neurocomputing, vol. 190, pp. 1–9, May 2016.
[14] L. Wang, Z. Cao, G. de Melo, and Z. Liu, “Relation classification via multi-level attention CNNs,” in Proc. 54th Annu. Meeting Assoc. Comput. Linguistics, vol. 1, 2016, pp. 1299–1307.
[15] Z. Geng, G. Chen, Y. Han, G. Lu, and F. Li, “Semantic relation extraction using sequential and tree-structured LSTM with attention,” Inf. Sci., vol. 509, pp. 183–192, Jan. 2020.
[16] P. Zhou et al., “Attention-based bidirectional long short-term memory networks for relation classification,” in Proc. 54th Annu. Meeting Assoc. Comput. Linguistics, vol. 2, 2016, pp. 207–212.
[17] L. B. Soares, N. FitzGerald, J. Ling, and T. Kwaikowski, “Matching the blanks: Distributional similarity for relation learning,” 2019, arXiv:1906.03158. [Online]. Available: https://arxiv.org/abs/1906.03158
[18] Y. Lin, S. Shen, Z. Liu, H. Luan, and M. Sun, “Neural relation extraction with selective attention over instances,” in Proc. 54th Annu. Meeting Assoc. Comput. Linguistics, 2016, pp. 2124–2133.
[19] Y. Xiao, Y. Jin, R. Cheng, and K. Hao, “Hybrid attention-based transformer block model for distant supervision relation extraction,” 2020, arXiv:2003.11518. [Online]. Available: https://arxiv.org/abs/2003.11518
[20] J. Feng, M. Huang, L. Zhao, Y. Yang, and X. Zhu, “Reinforcement learning for relation classification from noisy data,” 2018, arXiv:1808.08013. [Online]. Available: https://arxiv.org/abs/1808.08013
[21] P. Qin, W. Xu, and W. Y. Wang, “DSGAN: Generative adversarial training for distant supervision relation extraction,” 2018, arXiv:1805.09929. [Online]. Available: https://arxiv.org/abs/1805.09929
[22] J. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, “Generalizing from a few examples: A survey on few-shot learning,” ACM Comput. Surv., vol. 53, no. 3, pp. 1–34, 2020.
[23] J. Lu, P. Gong, J. Ye, and C. Zhang, “Learning from very few samples: A survey,” 2020, arXiv:2009.02653. [Online]. Available: https://arxiv.org/abs/2009.02653
[24] C. H. Lampert, H. Nickisch, and S. Harmeling, “Attribute-based classification for zero-shot visual object categorization,” 2013, arXiv:1311.1518. [Online]. Available: https://arxiv.org/abs/1311.1518
[25] Z. Chen, Y. Fu, Y. Zhang, Y.-G. Jiang, X. Xue, and L. Sigal, “Multi-level semantic feature augmentation for one-shot learning,” 2018, arXiv:1804.05298. [Online]. Available: https://arxiv.org/abs/1804.05298
[26] A. Antoniou, A. Storkey, and H. Edwards, “Data augmentation generative adversarial networks,” 2017, arXiv:1711.04340. [Online]. Available: http://arxiv.org/abs/1711.04340
[27] Z. Chen, Y. Fu, Y.-X. Wang, L. Ma, W. Liu, and M. Hebert, “Image deformation meta-networks for one-shot learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 8680–8689.
[28] M. Ren et al., “Meta-learning for semi-supervised few-shot classification,” 2018, arXiv:1803.00676. [Online]. Available: http://arxiv.org/abs/1803.00676
[29] X. Li et al., “Learning to self-train for semi-supervised few-shot classification,” in Proc. Adv. Neural Inf. Process. Syst., 2019, pp. 10276–10286.
[30] T. Ma and A. Zhang, “Affininet: Semi-supervised few-shot learning for disease type prediction,” in Proc. AAAI Conf. Artif. Intell., vol. 33, no. 1, 2019, pp. 1069–1076.
[31] J. Wang, K. Chen, L. Shou, S. Wu, and S. Mehrotra, “Semi-supervised few-shot learning for dual question-answer extraction,” 2019, arXiv:1904.03898. [Online]. Available: https://arxiv.org/abs/1904.03898
[32] S. J. Pan and Q. Yang, “A survey on transfer learning,” IEEE Trans. Knowl. Data Eng., vol. 22, no. 10, pp. 1345–1359, Oct. 2009.
[33] R. Caruana, “Learning many related tasks at the same time with backpropagation,” in Proc. Adv. Neural Inf. Process. Syst., 1995, pp. 657–664.

[34] Y. Bengio, “Deep learning of representations for unsupervised and transfer learning.” in Proc. ICML Workshop Unsupervised Transf. Learn., 2012, pp. 17–36.

[35] N. Mishra, M. Rohaninejad, X. Chen, and P. Abbeel, “A simple neural attentive meta-learner,” 2017, arXiv:1707.03141. [Online]. Available: https://arxiv.org/abs/1707.03141

[36] Munkhdalai and H. Yu, “Meta networks,” in Proc. 34th Int. Conf. Mach. Learn., 2017, pp. 2554–2563.

[37] C. Finn, P. Abbeel, and S. Levine, “Model-agnostic meta-learning for fast adaptation of deep networks.” 2017, arXiv:1703.03400. [Online]. Available: https://arxiv.org/abs/1703.03400

[38] V. Garcia and J. Bruna, “Few-shot learning with graph neural networks.” 2017, arXiv:1711.04043. [Online]. Available: https://arxiv.org/abs/1711.04043

[39] J. Bai et al., “Class incremental learning with few-shots based on linear programming for hyperspectral image classification,” IEEE Trans. Cybern., early access, Nov. 24, 2020, doi: 10.1109/TCYB.2020.3032958.

[40] J. Lu, S. Jin, J. Liang, and C. Zhang, “Robust few-shot learning for user-provided data,” IEEE Trans. Neural Netw. Learn. Syst., vol. 32, no. 4, pp. 1433–1447, Apr. 2021.

[41] N. Lai, M. Kan, C. Han, X. Song, and S. Shan, “Learning to learn adaptive classifier-predictor for few-shot learning.” IEEE Trans. Neural Netw. Learn. Syst., vol. 32, no. 8, pp. 3458–3470, Aug. 2020.

[42] T. Gao, X. Han, Z. Liu, and M. Sun, “Hybrid attention-based prototypical networks for noisy few-shot relation classification,” in Proc. AAAI Conf. Artif. Intell., vol. 33, 2019, pp. 6407–6414.

[43] S. Sun, Q. Sun, K. Zhou, and T. Lv, “Hierarchical attention prototypical networks for few-shot text classification,” in Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP), 2019, pp. 476–485.

[44] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[45] G. Elsayed, D. Krishnan, H. Mobahi, K. Regan, and S. Bengio, “Large margin deep networks for classification,” in Proc. Adv. Neural Inf. Process. Syst., 2018, pp. 842–852.

[46] J. Sokolic, R. Giryes, G. Sapiro, and M. R. D. Rodrigues, “Robust large margin deep neural networks,” IEEE Trans. Signal Process., vol. 65, no. 16, pp. 4265–4280, Aug. 2017.

[47] W. Liu, Y. Wen, Z. Yu, and M. Yang, “Large-margin softmax loss for convolutional neural networks,” in Proc. 33rd Int. Conf. Mach. Learn., 2016, pp. 507–516.

[48] Y. Wang et al., “Large margin few-shot learning,” 2018, arXiv:1807.02872. [Online]. Available: https://arxiv.org/abs/1807.02872

[49] A. Hermans, L. Beyer, and B. Leibe, “In defense of the triplet loss for person re-identification,” 2017, arXiv:1703.07737. [Online]. Available: https://arxiv.org/abs/1703.07737

[50] R. T. Rockafellar, Convex Analysis. Princeton, NJ, USA: Princeton Univ. Press, 1970, vol. 36.

[51] J. Pennington, R. Socher, and C. D. Manning, “Glove: Global vectors for word representation,” in Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP), 2014, pp. 1532–1543.

---

Yaochu Jin (Fellow, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees from Zhejiang University, Hangzhou, China, in 1988, 1991, and 1996, respectively, and the Dr.Ing. degree from Ruhr-University Bochum, Germany, in 2001. He is currently a Distinguished Chair Professor in Computational Intelligence, Department of Computer Science, University of Surrey, Guildford, U.K. He was also a Finland Distinguished Professor, Finland, and a Changjiang Distinguished Visiting Professor, China. His research interests lie primarily in the cross-disciplinary areas of computational intelligence, computational neuroscience, and computational systems biology. He has (co)authored over 400 peer-reviewed journal articles and conference papers and has been granted eight patents on evolutionary optimization.

Dr. Jin is a member of Academia Europaea. He was a Distinguished Lecturer of IEEE from 2013 to 2015 and from 2017 to 2019. He was the Vice President for Technical Activities of the IEEE Computational Intelligence Society from 2014 to 2015. He was a recipient of the 2015, 2017, and 2020 IEEE Computational Intelligence Magazine Outstanding Paper Award and the 2018 and 2021 IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION Outstanding Paper Award. He was awarded the Alexander von Humboldt Professorship for Artificial Intelligence in 2021 by the German Federal Ministry of Education and Research. He was named a “Highly Cited Researcher” in 2019 and 2020 by the Web of Science Group of Clarivate Analytics. He is also the Editor-in-Chief of the IEEE TRANSACTIONS ON COGNITIVE AND DEVELOPMENTAL SYSTEMS and Complex & Intelligent Systems.

Kuangrong Hao received the B.S. and M.S. degrees in mechanical engineering from Hebei University of Technology, Tianjin, China, in 1984 and 1989, respectively, the M.S. degree from the Ecole Normale Supérieure de Cachan, Cachan, France, in 1991, and the Ph.D. degree in applied mathematics and computer sciences the Ecole Nationale des Ponts et Chausées, Paris, France, in 1995.

She is currently a Professor at the College of Information Sciences and Technology, Donghua University, Shanghai, China. She has published more than 200 technical articles and five research monographs. Her scientific interests include intelligent perception, intelligent systems and network intelligence, robot control, and intelligent optimization of textile industrial process.