Mutual CRF-GNN for Few-shot Learning

Shixiang Tang[1†] Dapeng Chen[2] Lei Bai[1] Kaijian Liu[2] Yixiao Ge[3] Wanli Ouyang[1]
[1]The University of Sydney, SenseTime Computer Vision Group, Australia
[2]Sensetime Group Limited, Hong Kong [3]The Chinese University of Hong Kong, Hong Kong
{stan3903, lei.bai, wanli.ouyang}@sydney.edu.au
liukaijian@sensetime.com dapengchenxjtu@yahoo.com yxge@link.cuhk.edu.hk

Abstract

Graph-neural-networks (GNN) is a rising trend for few-shot learning. A critical component in GNN is the affinity. Typically, affinity in GNN is mainly computed in the feature space, e.g., *pairwise features, and does not take fully advantage of semantic labels associated to these features. In this paper, we propose a novel Mutual CRF-GNN (MCGN). In this MCGN, the labels and features of support data are used by the CRF for inferring GNN affinities in a principled and probabilistic way. Specifically, we construct a Conditional Random Field (CRF) conditioned on labels and features of support data to infer a affinity in the label space. Such affinity is fed to the GNN as the node-wise affinity. GNN and CRF mutually contributes to each other in MCGN. For GNN, CRF provides valuable affinity information. For CRF, GNN provides better features for inferring affinity. Experimental results show that our approach outperforms state-of-the-arts on datasets* mini*ImageNet,* tiered*ImageNet, and CIFAR-FS on both 5-way 1-shot and 5-way 5-shot settings.*

1. Introduction

Few-shot learning attempts to classify unlabelled data (*i.e.*, query samples) when only a few labelled data (*i.e.*, support samples) are available. Instead of relying on regularization to compensate for the data scarcity, researchers have explored ways to learn a distribution of similar tasks (also called "meta-learning"). Meta-learning method introduces the concept of *episodic*, which means that one round of model training contains only few samples (*e.g.*, 1 or 5) for each class. By episodic training, meta-learning methods aim to train a meta-learner that can quickly propagate labels from support samples to query samples.

Recently, Graph Neural Network (GNN) [60, 31] becomes a rising method to transfer the knowledge from the support samples to the query samples. In particular, Garcia and Bruna [15] first modelled the few-shot learning problem as a supervised graph message passing task by defining each sample in the support set and query set as a node in GNN.

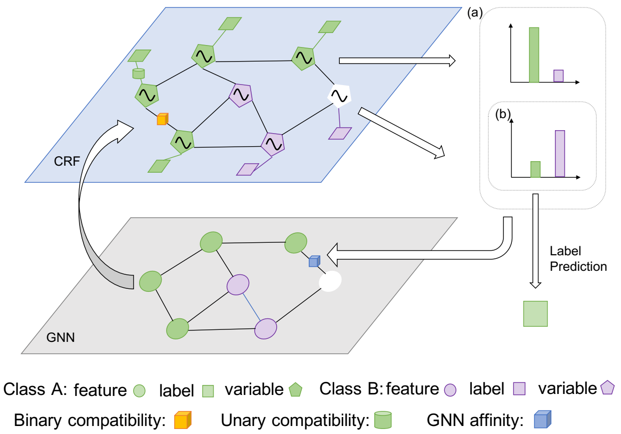

Figure 1. Illustration of Mutual CRF-GNN (MCGN). Green and purple indicate different classes. Unary compatibility contains label information and binary compatibility contains feature information from GNN. (a): The marginal distribution for pairwise variables can be used to predict the affinity for GNN. (b): the marginal distribution of single variable is used for label prediction.

Affinity, which measures the similarity between two samples/nodes, is a key component in GNN. Therefore, lots of approaches are proposed to have better affinity representation. EGNN [26] proposes to utilize labels for GNN affinity initialization and propagate the edge labels for explicitly modeling the intra-cluster similarity and inter-cluster dissimilarity. DPGN [53] propose to incorporates distribution propagation with GCN and combines both distribution-level relations with instance-level relations.

In this regard, we leverage CRF [48], which is a powerful probabilistic graphical model, to manipulate dependencies between variables, to collaborate with GNN. We model the labels as random variables in CRF. In our ap-

[†] This work was done when Shixiang Tang was an intern at SenseTime.

proach, the marginalized distributions in the unified CRF model have two functionalities. First, the marginalized distribution for single variable reflects the predicted possibility of label. Second, the marginalized probability of pair-wise variables defines the similarity of two samples, which is the affinity for GNN. Our design of MCGN is from the following two observations for CRF and GNN.

First, for CRF, the marginalized probabilities for single variable and pair-wise variables should be obtained by fusing feature information and label information. The unary compatibility term of each variable is used to model the relation between the variable/sample and the corresponding observed label information. The binary compatibility term utilizes the feature information. Specifically, it models the relation between two random variables/samples and is intuitively defined by the feature similarities of two corresponding random variables/samples. Since marginalizing the states of variables in CRF requires to multiply both unary and binary compatibility terms, the marginal distribution fuses the feature information and label information in a principled and probabilistic way.

Second, for GNN, its affinity should be defined by the probability in the label space, reflecting the possibility that two samples belong to the same class. Unlike typical GNN that determines the pairwise affinity in the feature space, *e.g.* using similarity of features, affinity determination by probabilities in label space merits two advantages. First, affinity defined in the label space is less sensitive to outliers. Take two samples that are visually similar but belong to different classes as an example. When using feature similarities to determine affinities, their affinity could be large, which leads to inappropriate feature aggregation between two samples. However, such affinity could be reduced in the label space because it is additionally guided by the provided semantic labels, *i.e.,* two samples have different class labels. Second, the labels given in the support set can guide the affinities in a probabilistic instead of deterministic manner. Different with EGNN and DPGN that initialize affinities to zero or one according to the corresponding labels, the unary compatibility term in CRF can set a tolerance for mislabelled samples, which makes our classification more robust than original GNN-based models.

Considering the above observations, we propose a unified model named Mutual CRF-GNN (MCGN), where the GNN and CRF are mutually correlated and can contribute to each other. The network consists of multiple layers and each layer alternately implements the CRF-based affinity inference and GNN-based feature aggregation. As illustrated in Fig. 1, we use features in GNN to define the binary compatibility in CRF. Next, with unary and binary compatibility, we estimate the marginal distribution of each variable. Afterwards, the obtained marginal distribution of each variable infers affinities in GNN. At last, more robust features are obtained by aggregation in GNN, which further leads to better compatibility in the next CRF layer. In such a feed-forward process, CRF produces better affinities with compatibilities defined by robust features in GNN and GNN produces robust features by taking affinity inferred by CRF.

In summary, our main contributions are two folds. First, we propose to introduce CRF to GNN, where CRF helps to implement dependencies between the predictions and define affinities of GNN in label space. Second, we propose a novel Mutual CRF-GNN where feature aggregation and relation inference could contribute to each other. Extensive experiments conducted on three popular datasets proved the effectiveness of Mutual CRF-GNN by a significant improvement in few-shot classification accuracy.

2. Related Work

Few-shot Learning. Research literature on few-shot learning is highly diverse. We focus on algorithms using supervised meta learning framework [21, 13, 49]. In particular, we divide these methods into three categories. (1) *Metric learning* based methods [52, 45, 47, 53, 9, 58, 42, 51] focus on obtaining a generalizable encoder to transform all samples into a common metric space and then use the distances between query features and support features to perform classification. (2) *Memory network* based methods [25, 34, 35] try to store knowledge from seen tasks and then generalize it to new tasks. (3) *Gradient descent* based methods [13, 40, 29, 18, 59] have a specific meta-learner that learns to adapt a specific base-learner to any few-shot learning task. Our method is closely related to *Metric learning* based methods, especially leveraging GNNs to measure similarities among few-shot samples.

GNN for Few-shot Learning. Recent development of *Metric learning* based methods is to leverage GNN [7]. GNNs can iteratively perform feature aggregations from neighbors, and therefore can explore complex similarities among features in the graph. Node-labeling GNN [15] aggregates node features to explore sample similarities. EGNN [26] additionally aggregates edges in GNNs for few-shot learning. DPGN [54] proposes to leverage distribution relations for affinity, which considers first-order global information. Our approach is different from the methods in two aspects. First, existing methods fused feature information and label information to determine affinities in a unprincipled and imbalance way. Node-labeling GNN concatenates the high-dimensional features and low-dimensional labels as a unified node feature, where label information is underestimated during aggregation. EGNN and DPGN ignore the labels information when feature transformation and only unitizes them for GNN initialization. Our method models affinity as the pair-wise marginal probabilities in CRF, where pair-wise marginal probabilities fuse feature and label information by multiplying unary compatibilities and binary compatibilities in a principled way. Second, the labels given the support set can guide affinity in a probabilistic manner. Different from EGNN and DPGN that initialize affinity by 0 or 1, the unary compatibility in CRF can set a tolerance for mislabelled samples, making our classification more robust.

CRF Approaches. Conditional Random Field (CRF) [27, 11, 12, 10, 6] is a popular probabilistic model to infer various dependencies within a image group or pixels in a image. Works that combine probabilistic models with neural networks to predict structured data can be found in various domains [3, 24, 55, 4]. [14] is the most relevant work to our method. It also uses CRF to enhance GNN but in the way of using the expectation of variables to generate new node features for transformation in the next GNN layer. Different from previous work that use CRF to predict labels or to generate new node features for GNN, we incorporate CRF in GNN models to calculate better affinities for GNN by marginal distributions, which has not been investigated before. Furthermore, CRF should collaborate with GNN to exploit the relations of the samples because it does not have a multiple-layer structure and can not be refined in an iterative way as GNN.

3. Methodology

3.1. Preliminaries

Problem Definition The target of few-shot learning is to learn a model that can generalize well to new tasks (*e.g.*, classes) with only a few labelled samples. Each few-shot task has a support set $\mathcal{S}$ and a query set $\mathcal{Q}$. The support set $\mathcal{S}$ contains N classes with K samples for each class (called N-way K-shot setting). Specifically, $\mathcal{S} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_{N\times K}, y_{N\times K})\}$, where $\mathbf{x}_*$ represents a sample and y_* represents its label. The query set $\mathcal{Q}$ has T samples, which can be denoted by $\mathcal{Q} = \{\mathbf{x}_{N\times K+1}, \mathbf{x}_{N\times K+2}, ..., \mathbf{x}_{N\times K+T}\}$. In the training stage, labels $\{y_{N\times K+1}, y_{N\times K+2}, ..., y_{N\times K+T}\}$ are provided for query set $\mathcal{Q}$. In the testing stage, we determine the label of the query sample according to the few labelled support samples. The labels of samples in the training stage and testing stage are mutually exclusive.

Modeling with GNN In few-shot learning, Graph Neural Network [17, 44] is a powerful post-processing tool to achieve robust features. Let $\mathbf{F} = (\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_{N\times K+T}) \in \mathbb{R}^{(N\times K+T)\times p}$ be the collection of $N \times K + T$ feature vectors in one few-shot task, where p is the feature dimension. The pairwise relationships of any two features are encoded in the affinity matrix $\mathbf{A} = \{a_{ij} | 1 \leq i, j \leq N \times K + T\} \in \mathbb{R}^{(N\times K+T)\times(N\times K+T)}$. A GNN usually contains several propagation (hidden) layers. Given an input $\mathbf{F}^0 = \mathbf{F}$ and the associated graph affinity $\mathbf{A}^0 = \mathbf{A}$, GNN conducts the following layer-wise propagation in the hidden layers as

$$\mathbf{F}^{l+1} = \sigma(\mathbf{D}^{-1/2}\mathbf{A}^l\mathbf{D}^{-1/2}\mathbf{F}^l), \tag{1}$$

where $l = 0, 1, ..., L-1$, $\mathbf{D}^l = \text{diag}(d_1, d_2, ..., d_n)$ is a diagonal matrix with $d_i = \sum_{j=1}^{n} a_{i,j}^l$ and σ is a trainable feature transformer. Typically, affinity $\mathbf{A}^l$ is often computed by the node-wise features and therefore may not be optimal for two reasons: (1) it only models pairwise information and ignores the neighboring information in GNN. (2) it only leverages feature information but does not incorporate the semantic information, *i.e.,* labels, when computing $\mathbf{A}^l$.

3.2. Introducing CRF to GNN

Conditional random fields (CRFs) are a class of statistical modeling method often used for structured prediction. To produce the affinities $\mathbf{A}^l$ that consider contexts, we utilize the marginal distribution of each random variable in CRFs to compute affinity in all GNN layers. Compared with conventional GNN that utilizes features to estimate affinity, using the marginal distribution of random variables brings three advantages. First, marginal distribution takes context into account whereas features only describes individual information. Second, marginal distribution incorporates both feature similarities (binary compatibility) and semantic labels of support samples (unary compatibility) into a unified quantity. Last, the space of marginal distribution is restricted by the label space. When used to estimate affinity, marginal distribution can help to explicitly illustrates whether two samples belong to the same class instead of whether two features are similar in typical GNN.

In particular, the l-th CRF layer is built upon a probabilistic graph $\mathcal{G}_l^{crf} = (\mathbf{V}_l^{crf}, \mathbf{E}_l^{crf})$ composed by all the samples in one few-shot learning task. $\mathbf{V}_l^{crf} = \{u_i^l\}_{i=1}^{N\times K+T}$ are the nodes of the graph, where u_i^l is the random variable associated to the sample i. It represents the labels assigned to the sample i, and may take any value from the label set. The conditional distribution for the CRF is given by:

$$\mathbf{P}(u_1^l, \dots, u_{N\times K+T}^l | \mathbf{F}_l, \mathcal{Y}_s) \propto \prod_{i=1}^{N\times K} \psi(u_i^l) \prod_{\langle j,k\rangle \in \mathbf{E}_l^{crf}} \phi(u_j^l, u_k^l), \tag{2}$$

where $\mathcal{Y}_s = \{y_1, y_2, ..., y_{N\times K}\}$ is a label collection for the support set, $\psi(u_i^l)$ is the unary compatibility between the random variable u_i^l and its label y_i, and $\phi(u_j^l, u_k^l)$ is the binary compatibility to describe the relationship between random variables u_j^l and u_k^l. In the following, we first introduce two compatibility functions in the CRF, then describe the details to estimate the marginal distribution $\mathbf{P}(u_i^l | \mathbf{F}_l, \mathcal{Y}_s)$ and the affinity $\mathbf{A}^l$ for the next layer GNN.

Unary Compatibility $\psi(u_i^l)$. Unary compatibility $\psi(u_i^l)$ is to describe the relation between the variable u_i^l of support samples and its corresponding observation, *i.e.* ground truth labels y_i. Mathematically, it can be formulated as

$$\psi(u_i^l = m) = \begin{cases} 1 - \eta & \text{if } m = y_i \\ \eta/(N-1) & \text{if } m \neq y_i \end{cases}, \tag{3}$$

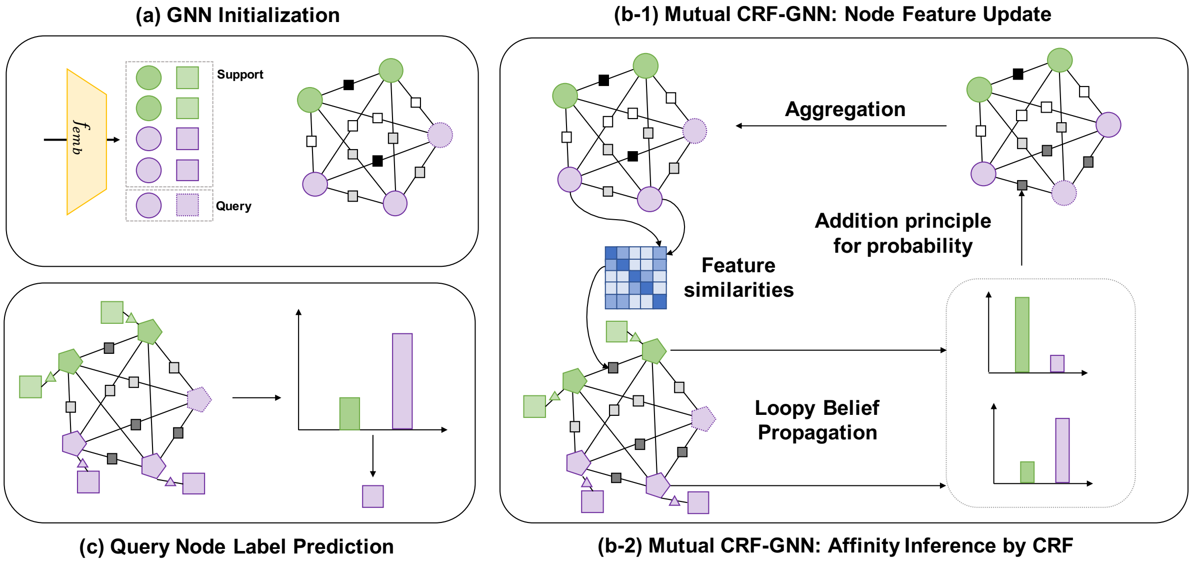
(a) GNN Initialization
f_{emb}
Support
Query
(b-1) Mutual CRF-GNN: Node Feature Update
Aggregation
Addition principle for probability
Feature similarities
Loopy Belief Propagation
(c) Query Node Label Prediction
(b-2) Mutual CRF-GNN: Affinity Inference by CRF

Figure 2. The overall framework of MCGN. This figure shows an example of a 2-way 2-shot setting, plus a query sample. The circles in GNN represent features extracted from backbone or aggregated by the previous layer. The squares in CRF means the labels of support samples. The label (dashed square) for the query sample is unknown. Purple and green colors for circles (squares) represent different classes. The pentagons in CRF denote the random variables which represent the labels assigned for the corresponding embedding in GNN.

where $\eta = 0.3$ is a small positive value which is the probability tolerance when the random variable takes the incorrect label. It is noteworthy that we define $\psi(u_i^l \neq y_i) = \eta/(N-1)$ because we consider there is tiny possibility for u_i^l to take incorrect labels. We set $\psi(u_i^l = y_i) = 1-\eta$ because we normalize summation $\sum_{m=1}^{N} \psi(u_i^l = m)$ to 1.

Binary Compatibility $\phi(u_j^l, u_k^l)$. Binary compatibility $\phi(u_j^l, u_k^l)$ is to describe the relations between the connected random variables, u_j^l and u_k^l. Mathematically, it can be formulated as

$$\phi(u_j^l = m, u_k^l = n) = \begin{cases} t_{j,k}^l & \text{if } m = n, \\ (1-t_{j,k}^l)/(N-1) & \text{if } m \neq n, \end{cases} \quad (4)$$

where m and n denote the labels assigned to u_j^l and u_k^l respectively, $t_{j,k}^l = \mathrm{ReLU}(\cos(\mathbf{f}_j^l, \mathbf{f}_k^l))$, and $\cos(\mathbf{f}_j^l, \mathbf{f}_k^l)$ indicates the cosine similarity between node features $\mathbf{f}_i^l$ and $\mathbf{f}_k^l$. According to Eq. 4, similar features lead to high compatibility when two samples take the same label and dissimilar features produce high compatibility when two samples take different labels.

Marginal Distribution $\mathbf{P}(u_i^l = m|\mathbf{F}_l, \mathcal{Y}_s)$. To incorporate states of other variables, we marginalize out all random variables other than u_i^l in Eq. 2 and derive marginal distribution $\mathbf{P}(u_i^l|\mathbf{F}_l, \mathcal{Y}_s)$ by

$$\mathbf{P}(u_i^l|\mathbf{F}_l, \mathcal{Y}_s) \propto \sum_{\mathcal{V}_l^{crf}\backslash\{u_i^l\}} \mathbf{P}(u_1^l, u_2^l, \ldots, u_{N\times K+T}^l|\mathbf{F}_l, \mathcal{Y}_s), \quad (5)$$

where $\mathbf{P}(u_i^l = m|\mathbf{F}_l, \mathcal{Y}_s)$ describes the probability of sample i being assigned with the label m after considering all possible states of random variables other than u_i^l. By considering all possible states of random variables other than u_i^l, $\mathbf{P}(u_i^l = m|\mathbf{F}_l, \mathcal{Y}_s)$ can exploit the contextual information of samples in both support set and query set. We adopt the loopy belief propagation [37] to calculate marginal distribution of each node in CRF (see Supplementary Material).

Affinity $\mathbf{A}^l$. Since marginal distribution $\mathbf{P}(u_i^l|\mathbf{F}_l, \mathcal{Y}_s)$ integrates both the contextual information in CRF and label information of support samples, we can use the marginal distribution to estimate a semantic affinity matrix $\mathbf{A}^l$. More specifically, the relation $\hat{a}_{ij}^l$ between $\mathbf{f}_i^l$ and $\mathbf{f}_j^l$ can be defined as the possibility of samples i and j belonging to the same class. Mathematically, it can be computed by addition theorem of probability:

$$\hat{a}_{ij}^l = \mathbf{P}(u_i^l = u_j^l) = \sum_{m=1}^{N} \mathbf{P}(u_i^l = m)\mathbf{P}(u_j^l = m). \quad (6)$$

Following implementation in EGNN [26], we aggregation the relation $\hat{a}_{ij}^l$ by its neighboring relations to get the final affinity for GNN, *i.e.*, $a_{ij}^l \leftarrow \frac{\hat{a}_{ij}^l a_{ij}^{l-1}}{\sum_k a_{ik}^{l-1}\hat{a}_{ik}^{l-1} / \sum_k a_{ik}^{l-1}}$, where k is the neighbor of i.

3.3. Mutual CRF-GNN

We propose a Mutual CRF-GNN (MCGN) that enables GNN and CRF to help each other. For GNN, CRF provides valuable affinity $\mathbf{A}^l$ for feature transformation $\mathbf{F}^{l+1}$. For CRF, GNN provides better features $\mathbf{F}^l$ for inferring affinity

$\mathbf{A}^l$. In the following, we describe how they can contribute to each other along with the overall pipeline of our method.

Initialization. Given the images in the support set and the query set, the raw feature $\mathbf{F}^1$ is extracted by a CNN-based feature extractor f_{emb}, *i.e.,*

$$\mathbf{F}^1 = f_{emb}(\mathcal{X}), \tag{7}$$

where $\mathcal{X} = \mathcal{S} \cup \mathcal{Q}$ contains all the samples in one task. The initial affinity matrix $\mathbf{A}^0$ in GNN is initialized by semantic labels from the support set, *i.e.*

$$a_{ij}^0 = \begin{cases} 1 & \text{if } y_i = y_j \text{ and } i, j \leq N \times K, \\ 0 & \text{if } y_i \neq y_j \text{ and } i, j \leq N \times K, \\ 0.5 & \text{otherwise}, \end{cases} \tag{8}$$

Feed-forward Implementation of MCGN. Given the raw feature $\mathbf{F}^1$ and the initialized affinity matrix $\mathbf{A}^0$, the final feature $\mathbf{F}^{L+1}$ for classification are transformed by MCGN for L iterations. We describe the detailed process of MCGN where CRF and GNN can mutually help each other for extracting discriminative features in few-shot learning. For the l-th layer, the whole process can be divided into 4 steps.

- *Step1:* Given the affinity $\mathbf{A}^{l-1}$ and output features $\mathbf{F}^l$ from $(l-1)$-th iteration, we estimate the unary and binary compatibility in the CRF by Eq. 3 and Eq. 4, respectively. The estimated compatibility functions define the affinities between two connected random variables in CRF.

- *Step2:* The marginal distribution (Eq. 5) for random variables in CRF is inferred by loopy belief propagation [37], using the compatibility functions obtained from *Step 1* and the labels of samples in the support set.

- *Step3:* The affinities $\mathbf{A}^l$ in GNN is derived from the marginal distributions obtained in step 2 by Eq. 6.

- *Step4:* The output features $\mathbf{F}^{l+1}$ of the l-th iteration are computed by aggregating their neighboring features with $\mathbf{A}^l$ as their weights by Eq. 1.

We repeat above process layer by layer for L iterations and get the final output $\mathbf{F}^{L+1}$ and affinity matrix $\mathbf{A}^L$ for network optimization and inference.

3.4. Training and Testing

Training. We supervise the output of GNN and CRF simultaneously. In particular, GNN is supervised by a verification loss $\mathcal{L}^{gnn}$ on affinity $\mathbf{A}^l$, and CRF can be supervised by a cross-entropy loss over the marginal distribution. This is because the marginal distribution $\mathbf{P}(u_i^l|\mathbf{F}^l, \mathcal{Y}_0)$ is represented as a N-dimensional vector $(p_{i,0}^l, p_{i,1}^l, ..., p_{i,N}^l)$, and $p_{i,j}^l$ represents the possibility u_i^l assigned label j, which is essentially a classification problem. The two loss functions can be defined as:

$$\mathcal{L}^{crf} = \sum_{i=N\times K}^{N\times K+T} \sum_{l=1}^{L+1} \mu_l^{crf} \mathbf{CE}(\mathbf{P}(u_i^l|\mathbf{F}^l, \mathcal{Y}_0), y_i), \tag{9}$$

$$\mathcal{L}^{gnn} = \sum_{i=N\times K}^{N\times K+T} \sum_{j=1}^{N\times K} \sum_{l=1}^{L} \mu_l^{gnn} \mathbf{BCE}(a_{ij}^l, c_{ij}), \tag{10}$$

where **CE** indicates the cross entropy, μ_l^{crf} is the weights of each layer; **BCE** indicates the binary cross entropy loss, μ_l^{gnn} is the weights of each layer, c_{ij} is 1 if $y_i = y_j$ and 0 if $y_i \neq y_j$. The total objective function can be a weighted summation of two losses, *i.e.,* $\mathcal{L} = \lambda_{crf}\mathcal{L}^{crf} + \lambda_{gnn}\mathcal{L}^{gnn}$, where $\lambda_{crf}, \lambda_{gnn}$ of each loss are set to balance their importance.

Testing. The class of each sample can be inferred by its final marginal distribution. We take the label that can maximize the mariginal distribution

$$\hat{y}_i = \mathbf{argmax}\ \mathbf{P}(u_i^{L+1}|\mathbf{F}^{L+1}, \mathcal{Y}_0). \tag{11}$$

4. Experiments

4.1. Datasets and Experimental Setup

Datasets: Our experiments are conducted on several widely used few-shot learning benchmarks, including *mini*ImageNet [52], *tiered*ImageNet [41], and CIFAR-FS [28]. ***mini*ImageNet** consists of 100 classes with 600 labeled instances in each category. We follow the standard protocol that utilizes 64 classes as the training set to train the feature extractor, 16 classes as the validation set, and 20 classes as the testing set. ***tiered*ImageNet** is a larger dataset compared with *mini*ImageNet, and its categories are selected with a hierarchical structure to split training and testing datasets semantically. We follow the dataset partition in [41] with 351 classes for training, 97 classes for validation and 160 classes for testing. The average number of images in each class is 1281. **CIFAR-FS** is a dataset with images from CIFAR-100. It contains 100 classes with 600 instances in each class. We follow the partition protocol given by [28], using 64 classes to construct the training set, 16 classes for validation and 20 classes for testing.

Evaluation Protocols: Evaluations are conducted in 5way-1shot/5shot settings on standard few-shot learning datasets, including *mini*ImageNet, *tiered*ImageNet and CIFAR-FS. The evaluation process is exactly the same as previous works [26, 28, 56]. In N-way K-shot setting, a meta-test task is composed of N classes, in which there are K samples. We randomly sample 600 meta-test tasks from the test dataset and then report the mean accuracy as well as the 95% confidence interval. For each meta-test task, we additionally sample 15 queries for each of 5 classes in 5way-1shot/5shot settings.

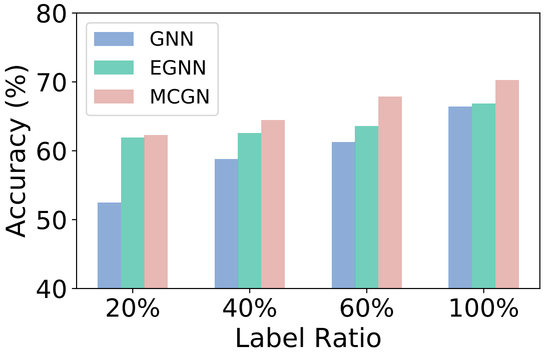

Figure 3. Semi-supervised few-shot learning accuracy in 5way-5shot on *mini*ImageNet. MCGN surpasses GNN and EGNN by a considerable margin consistently.

Network Architecture: We use two popular network backbones for a fair comparison, which are ConvNet and ResNet12 that are widely used in few-shot learning tasks [26, 28, 43, 8]. ConvNet is composed by four Conv-BN-ReLU blocks without any skip-connections, where the last of two blocks contain extra two Dropout layers [46]. ResNet12 is the same as the one proposed in [20]. The output of both ConvNet and ResNet12 is followed by a global average pooling and a fully-connected layer with batch normalization [23] to obtain a 128-dimension instance embedding.

Data Augmentation: Data augmentations are implemented before training as in [16, 56], which consists of horizontal flip, random crop and colour jitter. Each meta-train episode consists of N classes with K samples. We randomly sample 40 meta-train episodes in each iteration. Adam optimizer is leveraged in all experiments with the initial learning of 10^{-3}. The learning rate decay is set by 0.1 per 15000 iterations and the weight decay is set to 10^{-5}.

4.2. Episodic Training for Few-shot Learning

Episodic training is firstly introduced by Vinyals et al. [52] in few-shot learning. The training set and testing set are organized by episodes, each of which contains a support set and a query set. We compare the proposed MCGN with several state-of-the-art models including graphical and non-graphical models on the task of few-shot classification. The results (in the transductive setting) are reported in Table 2, where the mean value and deviations are averaged over 600 episodes. Our method outperforms current state-of-the-arts for both 5-way 1-shot setting and 5-way 5-shot setting on *mini*ImageNet, *tiered*ImageNet.

We analyze the results of our main competing methods, including EGNN [26], TPN [15] and DPGN [54] with ConvNet backbone in Table 2. EGNN employs features and affinities aggregations but these affinities are determined by the feature-wise similarities, whose performance is close to GNN-only method as reported in Table 3. Compared with EGNN, the main difference of our method is to employ the class-level affinity based on CRF inference, leading to 7.69%, 7.23% accuracy gain on *mini*ImageNet, *tiered*ImageNet respectively for 5-way 1-shot learning. For 5way-5shot learning, MCGN outperforms EGNN by 6.69% and 5.74% on *mini*ImageNet and *tiered*ImageNet. TPN propagates labels of the support samples to query samples by Laplacian matrix and takes the advantage of the support sample labels. However, it doesn't have a multi-layer structure and thus cannot refine the features by affinities layer by layer. Our method improves TPN by a large margin 13.17% and 11.3% on *mini*ImageNet and *tiered*ImageNet respectively for 5 way 5 shot few-shot learning. DPGN [54] leverages the distribution propagation to involve the contextual information for inference, but it does not leverage the given labels of the support samples. The lack of such semantic information may make the DPGN less effective than our method. Our method outperforms DPGN by 1.2% and 1.8% for 5-way 1-shot and 5-way 5-shot setting.

Method	***mini*ImageNet**		***tiny*ImageNet**	
	5w1s	5w5s	5w1s	5w5s
EP [42]	66.50	81.28	76.53	87.32
LST [32]	70.1	78.7	77.7	85.2
EMD [58]	65.91	82.41	71.16	86.03
FEAT [56]	66.78	82.05	70.80	84.79
Tian et.al (simple) [50]	62.02	79.64	69.74	84.41
Tian et.al (distillation) [50]	64.82	82.14	71.52	86.03
MCGN	68.87	86.58	77.12	89.22

Table 1. Feature-pretrained experiments on *mini*ImageNet and *tiny*ImageNet for both 5-way 1-shot (5w1s) and 5-way 5-shot (5w5s) few shot learning.

Inductive scenario. In the inductive scenario, we have to learn a function that produces a label for any given input. In such scenario, we treat each query sample independently and can not make the use of the relationships among query samples. The comparison of our proposed method and other states of the art is illustrated in Table 2. The proposed MCGN outperforms other inductive methods by about 1%. This is because MCGN can better utilize the contextual information in the feature space. On the other hand, the prevention of using the relation between query samples reduces the improvements in the inductive scenario by our method because our method relies on fully exploiting complex relations among all features in the graph.

Semi-supervised few-shot learning. We follow the same setting with [15] for semi-supervised experiments. In this setting, the labels of samples in support set are partially given, and are balanced among different classes to have same amount of labelled and unlabelled samples for each class. Results are presented in Figure 3, from which we can conclude: (1) With the increase of labelled ratio of samples in the support set, the testing accuracy of GNN, EGNN and MCGN improves by a large margin. (2) Our method, MCGN, outperforms EGNN [26] in different label ratio, namely 20%, 40%, 60%, 100%. The superiority of our method is relative small (about 1%) compared with EGNN when the label ratio = $20\%, 40\%$. This is because MCGN cannot take the full advantage of the labels in the support set. With the number increase of the labelled samples, our method can utilize more information in the support set, leading to larger margin with EGNN (about 3.5%).

4.3. Feature Pretrained Few-shot Learning

Feature pretrained few-shot learning is implemented by [50, 42, 22], where the feature embedding is trained by all samples in the meta-training stage. The baseline method proposed in [50] uses all samples in the meta-training stage to train the feature extractor and uses the samples in the support set during meta-testing stage to train the classifier in each meta-testing task. Different from the baseline method, all parameters in our proposed method are trained by meta-training samples. Specifically, our proposed method consists of two steps. First, we train the feature extractor following the same method in [50]. Second, we train the GNN and CRF components on the top by the fixed features that are computed by the feature extractor obtained.

To validate the effectiveness of our MCGN compared with other methods [42, 56, 30] in few-shot learning, we fix the feature extractor and train MCGN on top of the feature extractor. The results are illustrated in Table 1. Our proposed MCGN improves the baseline performance, *i.e,* Tian el.al (distill), by 4% in *mini*ImageNet and by 3% in *tiered*ImageNet, which shows that MCGN can further improve the classification even with very strong features. Our method also outperforms other methods that use pretrained models but finetune the backbone by at least 2%, which is consistent with the conclusion in [50].

4.4. Ablation Study

To investigate the contribution of GNN and CRF, we incrementally evaluate each of them on *mini*ImageNet by constructing 4 variants of our method. In particular, **Baseline** is the MatchingNet where similarities between support samples and query samples are directly calculated from feature embeddings. **GNN-only** is the GNN embedding model which can aggregate features and affinities but the affinity for GNN is defined by the embeddings of two connected nodes. **CRF-only** is the model where a single CRF directly follows the backbone. **CRF+GNN** is the model with two branches. One is the GNN branch which is the same as GNN-only and the other is the CRF branch which is the same as CRF-only. In this setting, CRF and GNN can not mutually contribute to each other. **MCGN** is the proposed method where CRF inference is leveraged to infer the affinity in GNN. Table 3 presents the performance of all variants. In this section, we first illustrate the contribution of GNN and CRF respectively by incrementally adding them to the baseline and then explore the mutual benefits between CRF and GNN. We also explore the contribution of the new CRF loss (Eq. 9) and the influence of an important hyperparameter in GNNs, i.e., number of layers in MCGN.

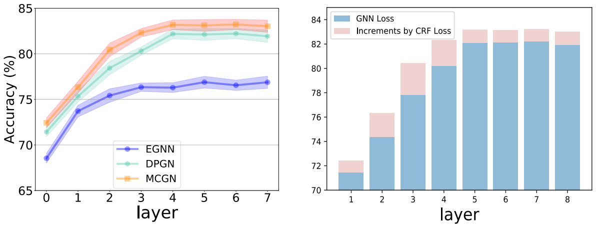

Figure 4. Experiments on *mini*ImageNet in 5-way 5-shot few shot learning (transductive scenario). Left: The few-shot classification accurcies with different number of layers in MCGN. The width of the colored region region indicates the variance of the performance. Right: Comparison of GNN loss and GNN loss+CRF loss.

Contribution of GNN and CRF. We compare Baseline and GNN-only in Table 3 to illustrate to what extend GNN contributes to the proposed method. We observe that GNN plays an important role for few-shot classification as only apply feature aggregation only improves the performance of the model by $10\% - 13\%$ for 5-way 1-shot setting and $7\% - 10\%$ for 5-way 5-shot setting. With the comparison between Baseline and CRF-only, we conclude the CRF can provide better affinity between support samples and query samples. CRF improves the model by $4\% - 6\%$ on *mini*ImageNet, *tiered*ImageNet and CIFAR-FS in 5-way 1-shot setting while the improvement on 5-way 5-shot setting is about $2\% - 4\%$. We attribute the improvement difference to the inaccurate marginal distribution estimation by loopy belief propagation [37] when CRF has dense connections and a large number of nodes [57]. The performance of GNN+CRF shows that GNN and CRF are cumulative.

Contribution of CRF and GNN Mutual Benefits. By comparing with CRF+GNN and GNN only, we can see the performance only improves a little. However, the comparison between CRF+GNN and MCGN in Table 3 illustrates that GNN and CRF can mutually help each other because we can see the classification accuracy by MCGN is significantly higher than that by CRF+GNN on *mini*ImageNet, *tiered*ImageNet and CIFAR-FS.

Contribution of Marginal Distribution Supervision. CRF loss (Eq. 9) uses marginal distribution of variables to supervised the network optimization. To illustrate the effectiveness of supervision on the marginal distribution of each random variable in CRF, we conduct experiments under GNN loss only (Eq. 10) and GNN (Eq. 10)+CRF loss (Eq. 9). From Fig. 4 (right), we can see the CRF+GNN loss

Method	Backbone	*mini*ImageNet		*tired*ImageNet	
		1-shot	5-shot	1-shot	5-shot
Inductive Learning					
Matching Network [52]	64-64-64-64	43.56 ± 0.84	55.31 ± 0.73	51.67 ± 1.81	70.30 ± 1.75
Prototypical Network [45]	64-64-64-64	49.42 ± 0.7	68.20 ± 0.66	53.31 ± 0.89	72.69 ± 0.74
TAML [25]	64-64-64-64	51.77 ± 1.86	66.05 ± 0.85	-	-
SAML [19]	64-64-64-64	52.22±n/a	66.49± n/a	-	-
GCR [30]	64-64-64-64	53.21 ± 0.80	72.34 ± 0.64	-	-
IMP [2]	64-64-64-64	49.2 ± 0.7	64.7 ± 0.7	-	-
KTN (Visual) [39]	64-64-64-64	54.61 ± 0.80	71.21 ± 0.66	-	-
R2D2 [5]	96-192-384-512	51.2 ± 0.6	68.8 ± 0.1	-	-
Reptile [38]	64-64-64-64	47.07 ± 0.8	62.74 ± 0.58	-	-
SNAIL [34]	ResNet-12	55.71 ± 0.99	68.88 ± 0.92	-	-
AdaResNet [36]	ResNet-12	56.88 ± 0.62	71.94 ± 0.57	-	-
GNN [15]	64-64-64-64	50.33 ± 0.36	66.41 ± 0.63	-	-
EGNN [26]	64-96-128-256	-	66.85 ± 0.63	-	70.98±n/a
DPGN† [54]	128-192-256-512	-	72.83 ± 0.74	-	-
MCGN	64-96-128-256	57.89 ± 0.87	73.58 ± 0.87	58.45 ± 0.59	74.58 ± 0.84
Transductive Learning					
Relation Network [47]	64-64-64-64	49.97 ± 0.32	65.99 ± 0.58	54.48 ± 0.93	71.32 ± 0.78
MAML [38]	64-64-64-64	48.70 ± 184	63.11 ± 0.92	-	-
Reptile [38]	64-64-64-64	50.44 ± 0.82	65.32 ± 0.70	-	-
EGNN [26]	64-96-128-256	59.63 ± 0.52	76.34 ± 0.48	63.52 ± 0.53	80.24 ± 0.87
TPN [33]	64-64-64-64	55.51 ± 0.86	69.86 ± 0.65	59.91 ± 0.94	73.30 ± 0.75
DPGN† [54]	128-192-256-512	66.14 ± 0.43	81.23 ± 0.41	69.91 ± 0.43	83.13 ± 0.46
MCGN	64-96-128-256	67.32 ± 0.43	83.03 ± 0.54	71.21 ± 0.85	85.98 ± 0.98

Table 2. Few-shot classification accuracies on *mini*ImageNet and *tiered*ImageNet. Results are reported in the inductive scenario and the transductive scenario, respectively. † denotes results re-implemented by public codes [1].

Method	*mini*ImageNet		*tiered*ImageNet		CIFAR-FS	
	5way 1shot	5way 5shot	5way 1shot	5way 5shot	5way 1shot	5way 5shot
Baseline	49.42 ± 0.98	68.20 ± 0.85	53.34 ± 0.78	72.69 ± 0.78	55.50 ± 0.86	72.01 ± 0.90
GNN only	58.93 ± 0.76	76.12 ± 0.94	62.62 ± 0.98	79.64 ± 0.87	69.47 ± 0.69	82.14 ± 0.74
CRF only	53.21 ± 0.76	71.34 ± 0.79	57.43 ± 0.72	76.04 ± 0.73	59.98 ± 0.89	75.69 ± 1.02
CRF+GNN	60.12 ± 0.57	78.64 ± 0.84	65.43 ± 0.93	82.23 ± 1.02	71.98 ± 0.99	84.22 ± 0.23
MCGN	67.32 ± 0.43	83.03 ± 0.54	71.21 ± 0.85	85.98 ± 0.98	76.45 ± 0.99	88.42 ± 0.23

Table 3. Ablation study of the baseline and three variants on *mini*ImageNet, *tiered*ImageNet and CIFAR-FS. Thare are 3 layers in GNN-only, CRF+GNN and MCGN. The accuracies are tested on 600 episodes in the transductive scenario.

can improve GNN loss only by about 1%.

Contribution of Multiple Layers in MCGN. We investigate the effects of the number of layers in MCGN. MCGN has a cyclic architecture which includes embeddings aggregation by GNN and a marginal distribution inference by CRF. To obtain the trend of testing accuracy, we report the results on *mini*ImageNet for 5-way 5-shot setting in Table 4 (left). More specifically, we ensure the convergence of Loopy Belief Propagation [37] and change the number of layers in MCGN. By changing the number of layers from 0 to 1, we can see the testing accuracy has a significant jump from 72.43% to 76.34%. When the number of layers continuously increases, the testing accuracy only marginally increases and will come to convergence in the last several numbers of layers. Comparing with EGNN, DPGN and MCGN, the performance of our proposed MCGN is consistently higher than that of the other two methods.

5. Conclusion

In this work, we present a novel framework, Mutual CRF-GNN (MCGN) for few-shot classification. MCGN combines GNN and CRF as a unified model, where the CRF can offer a better affinity for GNN and the GNN can produce a robust embedding by taking affinity from CRF. Our method significantly outperforms the current states of the art on extensive benchmarks.

6. Acknowledgement

Wanli Ouyang was supported by the SenseTime, Australian Research Council Grant DP200103223, and Australian Medical Research Future Fund MRFAI000085.

References

[1] Dpgn: Distribution propagation graph network for few-shot learning. https://github.com/megvii-research/DPGN#dpgn-distribution-propagation-graph-network-for-few-shot-learning. 7

[2] Kelsey R Allen, Evan Shelhamer, Hanul Shin, and Joshua B Tenenbaum. Infinite mixture prototypes for few-shot learning. *arXiv preprint arXiv:1902.04552*, 2019. 7

[3] Thierry Artieres et al. Neural conditional random fields. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 177–184, 2010. 2

[4] Yoshua Bengio, Yann LeCun, and Donnie Henderson. Globally trained handwritten word recognizer using spatial representation, convolutional neural networks, and hidden markov models. In *Advances in neural information processing systems*, pages 937–944, 1994. 2

[5] Luca Bertinetto, Joao F Henriques, Philip HS Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. *arXiv preprint arXiv:1805.08136*, 2018. 7

[6] Dapeng Chen, Dan Xu, Hongsheng Li, Nicu Sebe, and Xiaogang Wang. Group consistent similarity learning via deep crf for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8649–8658, 2018. 2

[7] Riquan Chen, Tianshui Chen, Xiaolu Hui, Hefeng Wu, Guanbin Li, and Liang Lin. Knowledge graph transfer network for few-shot recognition. *arXiv preprint arXiv:1911.09579*, 2019. 1

[8] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. *arXiv preprint arXiv:1904.04232*, 2019. 5

[9] Arnout Devos and Matthias Grossglauser. Subspace networks for few-shot classification. *arXiv preprint arXiv:1905.13613*, 2019. 1

[10] Justin Domke. Parameter learning with truncated message-passing. In *CVPR 2011*, 2011. 2

[11] Justin Domke. Learning graphical model parameters with approximate marginal inference. *IEEE transactions on pattern analysis and machine intelligence*, 35(10):2454–2467, 2013. 2

[12] Frederik Eaton and Zoubin Ghahramani. Choosing a variable to clamp. In *Artificial Intelligence and Statistics*, pages 145–152, 2009. 2

[13] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*, 2017. 1

[14] Hongchang Gao, Jian Pei, and Heng Huang. Conditional random field enhanced graph convolutional neural networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 276–284, 2019. 2

[15] Victor Garcia and Joan Bruna. Few-shot learning with graph neural networks. *arXiv preprint arXiv:1711.04043*, 2017. 0, 1, 5, 7

[16] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4367–4375, 2018. 5

[17] Marco Gori, Gabriele Monfardini, and Franco Scarselli. A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pages 729–734. IEEE, 2005. 2

[18] Erin Grant, Chelsea Finn, Sergey Levine, Trevor Darrell, and Thomas Griffiths. Recasting gradient-based meta-learning as hierarchical bayes. *arXiv preprint arXiv:1801.08930*, 2018. 1

[19] Fusheng Hao, Fengxiang He, Jun Cheng, Lei Wang, Jianzhong Cao, and Dacheng Tao. Collect and select: Semantic alignment metric learning for few-shot learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8460–8469, 2019. 7

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[21] Geoffrey E Hinton and David C Plaut. Using fast weights to deblur old memories. In *Proceedings of the ninth annual conference of the Cognitive Science Society*, pages 177–186, 1987. 1

[22] Shaoli Huang and Dacheng Tao. All you need is a good representation: A multi-level and classifier-centric representation for few-shot learning. *arXiv preprint arXiv:1911.12476*, 2019. 6

[23] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 5

[24] Viren Jain, Joseph F Murray, Fabian Roth, Srinivas Turaga, Valentin Zhigulin, Kevin L Briggman, Moritz N Helmstaedter, Winfried Denk, and H Sebastian Seung. Supervised learning of image restoration with convolutional networks. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007. 2

[25] Muhammad Abdullah Jamal and Guo-Jun Qi. Task agnostic meta-learning for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11719–11727, 2019. 1, 7

[26] Jongmin Kim, Taesup Kim, Sungwoong Kim, and Chang D Yoo. Edge-labeling graph neural network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11–20, 2019. 0, 1, 3, 4, 5, 6, 7

[27] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001. 2

[28] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10657–10665, 2019. 4, 5

[29] Yoonho Lee and Seungjin Choi. Gradient-based meta-learning with learned layerwise metric and subspace. *arXiv preprint arXiv:1801.05558*, 2018. 1

[30] Aoxue Li, Tiange Luo, Tao Xiang, Weiran Huang, and Liwei Wang. Few-shot learning with global class representations. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9715–9724, 2019. 6, 7

[31] Suichan Li, Dapeng Chen, Bin Liu, Nenghai Yu, and Rui Zhao. Memory-based neighbourhood embedding for visual recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 0

[32] Xinzhe Li, Qianru Sun, Yaoyao Liu, Qin Zhou, Shibao Zheng, Tat-Seng Chua, and Bernt Schiele. Learning to self-train for semi-supervised few-shot classification. In *Advances in Neural Information Processing Systems*, pages 10276–10286, 2019. 5

[33] Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, Eunho Yang, Sung Ju Hwang, and Yi Yang. Learning to propagate labels: Transductive propagation network for few-shot learning. *arXiv preprint arXiv:1805.10002*, 2018. 7

[34] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. *arXiv preprint arXiv:1707.03141*, 2017. 1, 7

[35] Tsendsuren Munkhdalai and Hong Yu. Meta networks. *Proceedings of machine learning research*, 70:2554, 2017. 1

[36] Tsendsuren Munkhdalai, Xingdi Yuan, Soroush Mehri, and Adam Trischler. Rapid adaptation with conditionally shifted neurons. In *International Conference on Machine Learning*, pages 3664–3673. PMLR, 2018. 7

[37] Kevin Murphy, Yair Weiss, and Michael I Jordan. Loopy belief propagation for approximate inference: An empirical study. *arXiv preprint arXiv:1301.6725*, 2013. 3, 4, 6, 7

[38] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018. 7

[39] Zhimao Peng, Zechao Li, Junge Zhang, Yan Li, Guo-Jun Qi, and Jinhui Tang. Few-shot image recognition with knowledge transfer. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 441–449, 2019. 7

[40] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2016. 1

[41] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. Meta-learning for semi-supervised few-shot classification. *arXiv preprint arXiv:1803.00676*, 2018. 4

[42] Pau Rodríguez, Issam Laradji, Alexandre Drouin, and Alexandre Lacoste. Embedding propagation: Smoother manifold for few-shot classification. *arXiv preprint arXiv:2003.04151*, 2020. 1, 5, 6

[43] Andrei A Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. *arXiv preprint arXiv:1807.05960*, 2018. 5

[44] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2008. 2

[45] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in neural information processing systems*, pages 4077–4087, 2017. 1, 7

[46] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014. 5

[47] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018. 1, 7

[48] Charles Sutton and Andrew McCallum. An introduction to conditional random fields for relational learning. *Introduction to statistical relational learning*, 2:93–128, 2006. 0

[49] Sebastian Thrun and Lorien Pratt. Learning to learn: Introduction and overview. In *Learning to learn*, pages 3–17. Springer, 1998. 1

[50] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? *arXiv preprint arXiv:2003.11539*, 2020. 5, 6

[51] Pavel Tokmakov, Yu-Xiong Wang, and Martial Hebert. Learning compositional representations for few-shot recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6372–6381, 2019. 1

[52] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016. 1, 4, 5, 7

[53] Ling Yang, Liangliang Li, Zilun Zhang, Erjin Zhou, Yu Liu, et al. Dpgn: Distribution propagation graph network for few-shot learning. *arXiv preprint arXiv:2003.14247*, 2020. 0, 1

[54] Ling Yang, Liangliang Li, Zilun Zhang, Xinyu Zhou, Erjin Zhou, and Yu Liu. Dpgn: Distribution propagation graph network for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1, 5, 7

[55] Kaisheng Yao, Baolin Peng, Geoffrey Zweig, Dong Yu, Xiaolong Li, and Feng Gao. Recurrent conditional random field for language understanding. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4077–4081. IEEE, 2014. 2

[56] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. Learning embedding adaptation for few-shot learning. *arXiv preprint arXiv:1812.03664*, 2018. 4, 5, 6

[57] KiJung Yoon, Renjie Liao, Yuwen Xiong, Lisa Zhang, Ethan Fetaya, Raquel Urtasun, Richard Zemel, and Xaq Pitkow. Inference in probabilistic graphical models by graph neural networks. In *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, pages 868–875. IEEE, 2019. 6

[58] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12203–12213, 2020. 1, 5

[59] Ruixiang Zhang, Tong Che, Zoubin Ghahramani, Yoshua Bengio, and Yangqiu Song. Metagan: An adversarial ap-

proach to few-shot learning. In *Advances in Neural Information Processing Systems*, pages 2365–2374, 2018. 1

[60] Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *arXiv preprint arXiv:1812.08434*, 2018. 0