MHFC: Multi-Head Feature Collaboration for Few-Shot Learning

Shuai Shao\textsuperscript{4,1}, Lei Xing\textsuperscript{1,2}, Yan Wang\textsuperscript{3}, Rui Xu\textsuperscript{1}, Chunyan Zhao\textsuperscript{4}, Yanjiang Wang\textsuperscript{*,1}, Baodi Liu\textsuperscript{*,1}
\textsuperscript{1}College of Control Science and Engineering, China University of Petroleum (East China)
\textsuperscript{2}College of Oceanography and Space Informatics, China University of Petroleum (East China)
\textsuperscript{3}Beihang University
\textsuperscript{4}Suzhou Centennial College
shuaishao@s.upc.edu.cn, upc_xl@163.com, wangyan9509@gmail.com, ruixu@s.upc.edu.cn
zhaoicy@scc.edu.cn, yjwang@upc.edu.cn, thu.liubaodi@gmail.com

ABSTRACT

Few-shot learning (FSL) aims to address the data-scarce problem. A standard FSL framework is composed of two components: (1) Pre-train. Employ the base data to generate a CNN-based feature extraction model (FEM). (2) Meta-test. Apply the trained FEM to acquire the novel data's features and recognize them. FSL relies heavily on the design of the FEM. However, various FEMs have distinct emphases. For example, several may focus more attention on the contour information, whereas others may lay particular emphasis on the texture information. The single-head feature is only a one-sided representation of the sample. Besides the negative influence of cross-domain (e.g., the trained FEM can not adapt to the novel class flawlessly), the distribution of novel data may have a certain degree of deviation compared with the ground truth distribution, which is dubbed as distribution-shift-problem (DSP).

To address the DSP, we propose Multi-Head Feature Collaboration (MHFC) algorithm, which attempts to project the multi-head features (e.g., multiple features extracted from a variety of FEMs) to a unified space and fuse them to capture more discriminative information. Typically, first, we introduce a subspace learning method to transform the multi-head features to aligned low-dimensional representations. It corrects the DSP via learning the feature with more powerful discrimination and overcomes the problem of inconsistent measurement scales from different head features. Then, we design an attention block to update combination weights for each head feature automatically. It comprehensively considers the contribution of various perspectives and further improves the discrimination of features. We evaluate the proposed method on five benchmark datasets (including cross-domain experiments) and achieve significant improvements of 2.1%-7.8% compared with state-of-the-arts.

CCS CONCEPTS
\begin{itemize}
\item Computing methodologies $\rightarrow$ Image representations; Object identification.
\end{itemize}

\textsuperscript{1} indicates equal contribution. \textsuperscript{*} indicates corresponding author.

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

\textsuperscript{1} INTRODUCTION

Machine learning has helped machines achieve outstanding performances in computer vision tasks, such as person re-identification [8, 45, 46], image classification [17, 35, 50]. One indispensable factor is attributed to the large-scale labeled data. However, as the limitation of actual circumstances, it may be infeasible to collect large amounts of labeled data in the real world. Thus, few-shot learning (FSL) has attracted growing attention recently. It targets to help machines achieve or even surpass human beings' level with scarce labeled samples. Generally, in FSL-based classification tasks, the current popular model usually includes two components: (1) Pre-train. Employ the base data $D_{base}$ to generate a convolutional neural network (CNN) based feature extraction model (FEM). (2) Meta-test. First, extract the features of novel data $D_{novel} = \{S, U, Q\}$, where $S$, $U$ and $Q$ denote support set, unlabeled set and query set. Next, design a classifier to recognize the query samples $Q$. Notably,
$\Omega$}

We dub this fundamental problem as a distribution-shift-problem (DSP) which exists a problem of inconsistent measurement scales. Thus, (DSP). 

To tackle the DSP and overcome these two challenges, we propose a novel Multi-Head Feature Collaboration (MHFC) algorithm to improve the discrimination of sample’s feature. MHFC is a simple non-parametric model that can directly fuse multi-head features extracted from the existing FEMs, such as ICI-Net [47], MetaOpt-Net [2]. To solve the first challenge, we introduce a subspace learning method to transform the raw multi-head features to a unified space with reconstructed low-dimensional representation. It is also helpful to reduce redundant information. Next, we design an attention block to automatically update combination weights for each head of the feature to solve the second challenge efficiently. Finally, we generate the collaborative representation of the sample by concatenating the processed multi-head features. We illustrate the flowchart in Figure 2.

Besides, according to the data adopted in the design of the classifier, researchers categorize the FSL-based approaches as three sorts: (1) Inductive few-shot learning (IFSL). (2) Transductive few-shot learning (TFSL). (3) Semi-supervised few-shot learning (SSFSL). This paper extends the proposed MHFC to these three settings.

In summary, the main contributions focus on: 

(1) We propose a novel method for FSL, dubbed as Multi-Head Feature Collaboration (MHFC) and extend it to three FSL-based settings. It addresses the DSP by comprehensively considering the multi-head features to improve the discrimination of sample’s representation.

(2) Compared to tuning the network to solve DSP, the proposed MHFC is more straightforward and effective. It is a simple non-parametric method that directly fuses multi-head features extracted from the existing FEMs. Besides, the more robust feature representation enables the proposed MHFC to achieve amazing performance especially when the samples are extremely scarce (e.g., inductive
setting on 5-way 1-shot case, please refer to Table 1, 2). Thus, we consider this paper may be meaningful to be applied in reality.

(3) We evaluate the proposed method on four benchmark datasets (mini-ImageNet, tiered-ImageNet, CIFAR-FS, FC100) and achieve significant improvements of 2.1%-7.8% compared with other state-of-the-art methods. Besides, to prove that the proposed method can overcome DSP, we design the cross-domain experiments (e.g., mini-ImageNet → CUB) and achieve far better performance than state-of-the-art methods of at least 7.8%.

2 RELATED WORK

2.1 Few-Shot Learning

In the past decade, FSL based works have attracted lots of attention. Researchers have proposed various classical frameworks to solve this problem. We list the two most popular types, including (1) Meta-learning based methods, such as MAML [10], Reptile [24], LEO [34], which purpose to obtain a universal model to rapidly adapt to new tasks. (2) Metric learning based methods, focusing on looking for ideal distance metrics to strengthen model’s robustness, including ProtoNet [36], MetaOpt [2], TADAM [25] et al. In addition, all these methods can be split into another taxonomy, e.g. inductive few-shot learning (IFSL), transductive few-shot learning (TFSL), and semi-supervised few-shot learning (SSFSL). For example, MAML [10], LEO [34], S2M2 [23] et al. are based on inductive setting; DPGN [51], TEAM [27], SIB [12] et al. are TFSL methods; and LST [16], EPNet [30], ICI [47] et al. are based on semi-supervised setting.

2.2 Multi-Modal Learning

Just as every coin has two sides, it would be incomplete to define objects from a single perspective. Therefore, multi-modal learning has received wide attention in recent years. There exist lots of classical methods and corresponding applications. For example, Liu et al. proposed a sparse coding based multi-modal method MHDSC [21] for image annotation task; Liu et al. proposed SPM-CRC [19], which improves the collaborative representation model from multi-modal learning to classify remote sensing images; Jan et al. proposed MVCCA [32] and employed it in natural language processing. Liu et al. proposed MHL [20] to solve Alzheimer’s Disease Predicting problem; Zhang et al. proposed IMHL [55], which is an inductive hypergraph learning from multi-modal and applied it for 3D object recognition. All these methods may help FSL, and some multi-modal based FSL methods were proposed, including [6] (introduced ensemble strategy in pretrain), [54] (adopted multi-head classifier in fine-tuning ), [7] (fused multi-domain representation). This paper makes an orthogonal contribution towards efficient semantic-preserving pretraining.

2.3 Subspace Learning

Subspace learning is capable of transferring the samples to another representation. It is an efficient dimension reduction method. Here, we list several classical subspace learning methods, which have been employed in the proposed MHFC. The first one is Principal Component Analysis (PCA) [39]. It projects the raw features to a low-dimensional space by using singular value decomposition. The second one is Locally Linear Embedding (LLE) [31], which tries to preserve distances within local according to seeking a low-dimensional projection of the data. The last one is Laplacian Eigenmap (LE) [1]. It exploits a spectral decomposition of the graph Laplacian to project the raw data to a low-dimensional representation. All these methods are helpful for our MHFC, and we will show the comparison results in Section 4.2.1.

3 PROBLEM FORMULATION

In this section, we introduce the complete procedure in details. It is composed of two phases, including pre-train and meta-test. i) In pre-train phase, given base data \( D_{base} = \{(x_i, y_i) | y_i \in C_{base}\}_{i=1}^{N_{base}} \), where \( x \) and \( y \) denote the sample and corresponding label, respectively. \( N_{base} \) denotes the total number of base data. \( C_{base} \) indicates the base category set. We train the CNN-based FEM \( M_\theta() \) on \( D_{base} \), where \( \theta \) indicates the parameters in CNN. In this paper, divers FEMs are employed to extract features for different heads, and we define the FEM on the \( h_{th} \) head as \( M^h_{\theta}(\cdot) \), where \( h = 1, 2, \ldots, H \), more details please refer to Section 4.4.

ii) For the meta-test phase, the novel data \( D_{novel} = \{(x_j, y_j) | y_j \in C_{novel}, j \in \{1\}, \ldots, \{N_{novel}\}\} \), where \( C_{novel} \) denotes the novel category set, \( N_{novel} \) denotes the number of novel data. \( C_{base} \cap C_{novel} = \emptyset \). \( D_{novel} \) includes three components, e.g. \( D_{novel} = \{S, U, Q\} \), where \( S, U \) and \( Q \) denote support set, unlabeled set and query set, \( S \cap U = \emptyset \), \( S \cap Q = \emptyset \), \( Q \cap U = \emptyset \). In this phase, we first utilize the trained FEM to extract \( D_{novel} \)’s feature, then design a classifier to classify \( Q \). There exist three settings to design the classifier, e.g., inductive setting, transductive setting, and semi-supervised setting. Section 4.3 show more details. We follow standard C-way-T-shot per episode as [47] for classification, where C-way denotes C classes, and T-shot indicates T samples per class. We average the accuracies of all the episodes with 95% confidence intervals as the final result.

4 METHODOLOGY

In this section, we first briefly review the linear regression classifier as an example to introduce our method. Then, we propose the novel Multi-Head Feature Collaboration (MHFC) algorithm to collaboratively represent the samples with processed multi-head features. Next, we extend the proposed MHFC to three kinds of few-shot learning settings. Finally, we introduce the employed multi-head feature extraction model.

4.1 Review of Linear Regression Classifier

This proposed method pays attention to the various features that can integrate all types of conventional classification strategy (such as linear regression, support vector machine, logistic regression). In this paper, we employ a simple regularized linear regression model as an example to show the details of our method. We formulate the objective function as:

\[
\arg\min_W \mathcal{F} = \|Y - WX\|^2_F + \mu \|W\|^2_F
\]

where \( \|\cdot\|^2_F \) represents (·)’s Frobenius-norm. \( \mu \) is the hyperparameter. \( X = [x_1, x_2, \ldots, x_N] \in \mathbb{R}^{dim \times N} \), \( Y = [y_1, y_2, \ldots, y_N] \in \mathbb{R}^{C \times N} \). \( dim \) and \( N \) indicate the dimension and number of labeled samples, respectively. \( C \) denotes
the number of categories. \( x_n, y_n \) (\( n = 1, 2, \ldots, N \)) denote the feature embedding vector and one-hot label vector of the \( n_{th} \) sample. \( W \in \mathbb{R}^{C \times \text{dim}1} \) represents the to-be-learned classifier. We directly optimize the objective function and obtain the \( W \) as:

\[
W = YX^T \left( XX^T + \mu I \right)^{-1}
\]

where \( I \) is the identity matrix. Following, given a testing sample embedding \( x_{ts} \in \mathbb{R}^{\text{dim}1} \), we predict the \( x_{ts} \)'s category by:

\[
\mathcal{A}(x_{ts}) = \max \{ Wx_{ts} \}
\]

where \( \max \) denotes an operator to obtain the index of the max value in the vector.

### 4.2 Multi-Head Feature Collaboration

#### 4.2.1 Multi-Head Feature Transformation

Since one kind of incomplete feature cannot reflect a sample well, we introduce various features to represent samples from different heads collaboratively. Assume that we have \( H \) heads in total. Each head corresponds to one kind of feature \( X^h \), where \( h = 1, 2, \ldots, H \). As mentioned in Section 1, the multi-head features are in separate spaces, which exists a problem of inconsistent measurement scales.

To address this challenge, we introduce a conventional subspace learning method (denoted as \( \mathcal{F}(\cdot) \)) to transform the raw features into a unified space with reconstructed low-dimensional representation. Specifically, we treat \( H \) heads of the same sample as \( H \) samples, and denote the features of the expanded dataset as \( X_{\text{exp}} = [X^1, X^2, \ldots, X^H] \in \mathbb{R}^{\text{dim}1 \times (N \times H)} \). We conduct subspace learning operation \( \mathcal{F}(X_{\text{exp}}) \) and obtain the novel features \( X'_{\text{exp}} = [P^1, P^2, \ldots, P^H] \in \mathbb{R}^{\text{dim}2 \times (N \times H)} \), where \( P^h \in \mathbb{R}^{\text{dim}2 \times N} \) denotes the feature on the \( h_{th} \) head after subspace transformation. \( \text{dim}2 \) denotes the novel dimension.

#### 4.2.2 Multi-Head Feature Attention Block

Consider that the significance of these features is different per episode. We try to find the optimal combination weights \( \Omega = [\Omega^1, \Omega^2, \ldots, \Omega^H]^T \) to let these features have different influences for the final decision. where \( \Omega \) is a weight vector, \( \Omega^h (h = 1, 2, \ldots, H) \) denotes the \( h_{th} \) element in \( \Omega \). We first employ the transformed feature \( P^h \) to replace \( X^h \) and obtain a novel classifier \( W^h \in \mathbb{R}^{C \times \text{dim}2} \) according to Equation 2, where \( W^h \) indicates the trained novel classifier on the \( h_{th} \) head, which can be formulated as:

\[
W^h = \gamma P^h X^T \left( P^h X^T + \mu I \right)^{-1}
\]

Then use \( P^h \) and \( W^h \) to re-calculate the objective function’s loss on the \( h_{th} \) head \( \mathcal{F}^h \) by Equation 1, which can be formulated as:

\[
\mathcal{F}^h = \| Y - W^h P^h \|^2_F + \frac{\mu}{2} \| W^h \|^2_F
\]

Next, exploit the \( \mathcal{F}^h \) to compute the combination weights. The objective function can be formulated as:

\[
\arg \min_{\Omega} \sum_{h=1}^{H} \left( \mathcal{F}^h \right) + \eta \| \Omega \|^2_2 \quad \text{s.t.} \quad H \Omega^h = 1 \Omega^h \geq 0
\]

where \( \Omega^h \) indicates the weight of \( h_{th} \) head. \( \| \cdot \|_2 \) represents (·)'s \( \ell_2 \)-norm. \( \eta \) is the parameter. We introduce the Lagrangian to solve the problem, the Equation 6 can be rewritten as:

\[
\arg \min_{\Omega, \lambda} \sum_{h=1}^{H} \left( \mathcal{F}^h \right) + \eta \| \Omega \|^2_2 - \lambda^T \Omega
\]

where \( \lambda \) is a constant, \( \Lambda = [\Lambda^1, \Lambda^2, \ldots, \Lambda^H]^T \) is a vector. Assume \( \hat{\Omega}, \hat{\lambda}, \hat{\Lambda} \) are the optimal solutions, we solved this problem as:

\[
\hat{\Omega}^h = \frac{1}{2\eta} \max \left\{ \frac{\sum_{h=1}^{H} \mathcal{F}^h}{H} + \frac{2\eta}{H} - \mathcal{F}^h - \hat{\lambda}_{\text{avg}}, 0 \right\}
\]

where \( \hat{\lambda}_{\text{avg}} \) is a constant, denotes the average of \( \hat{\lambda} \). For the detailed optimization process, please refer to Appendix A.2.

#### 4.2.3 Multi-Head Feature Collaborative Classifier

After giving multi-head features the combination weights, we obtain the final collaborative feature \( Z = [z_1, z_2, \ldots, z_N] \in \mathbb{R}^{\text{dim}3 \times N} \) by:

\[
Z \leftarrow z = \text{Concat} \left( \hat{\Omega}^1_p \nabla_1, \hat{\Omega}^2_p \nabla_2, \ldots, \hat{\Omega}^H_p \nabla_H \right)
\]

where \( p^1_z, z^2_z (n = 1, 2, \ldots, N) \) denote the \( n_{th} \) vector of \( P^h \) and \( Z^h \).

Next, according to Equation 2, we use \( Z \) to replace \( X \) and obtain the final collaborative classifier \( W_z \in \mathbb{R}^{C \times \text{dim}3} \), which can be formulated as:

\[
W_z = YZ^T \left( ZZ^T + \mu I \right)^{-1}
\]

Finally, given a testing sample feature \( x_{ts}^h (h = 1, 2, \ldots, H) \), we first obtain the collaborative feature \( z_{ts} \in \mathbb{R}^{\text{dim}3} \) by Equation 8, 9, then predict the label by:

\[
\mathcal{A}(z_{ts}) = \max \{ W_z z_{ts} \}
\]

### 4.3 MHFC for Few-Shot Learning

Define the feature of \( D_{\text{novel}} \) on the \( h_{th} \) head as \( X^h_{\text{novel}} = [X^h_1, X^h_2, \ldots, X^h_N] \), where \( X^h_k = M^h_0(S)_k \), \( X^h_{\text{unlabeled}} = M^h_0(U)_k \), and \( X^h_{\text{query}} = M^h_0(Q)_k \) denote the features of support, unlabeled, and query data on the \( h_{th} \) head. Thus, the \( [X^h_{\text{novel}}] \)'s feature after transformation can be defined as \( P^h_{\text{novel}} = [P^h_1, P^h_2, \ldots, P^h_H] \). Researchers employ different data to design the classifier, and these methods can be split into three settings, e.g., inductive setting, semi-supervised setting, and transductive setting.

#### 4.3.1 Semi-Supervised MHFC

Semi-supervised setting in few-shot learning adopts the support set \( S \) and unlabeled set \( U \) to train the classifier, and then predict the query label. In this paper, we extend our MHFC to the semi-supervised setting by introducing the simple self-training strategy. It can employ the unlabeled data to strengthen the classifier. We show the detailed steps as:

1. Exploit the support data to train the basic classifier on each head, Equation 4 can be rewritten as:

\[
W^h = Y^h s^h T \left( P^h s^h T + \mu l \right)^{-1}
\]

where \( Y_s \) denotes the one-hot label matrix of support data.
where
\[ p \]

(e.g.,
\[ Z \]

FSL (e.g., there not exist unlabeled data). Thus, we can use the steps (e.g., (1), (2), (3), (7), (8)) to implement the inductive MHFC.

\[ \mathcal{L}_c = - \sum_c y_{c(x)} \log(p_{c(x)}) \]

(20)

where \( c \in C_{\text{base}} \) denotes the \( c_{th} \) class. \( y_{c(x)} \), \( p_{c(x)} \) indicate the probabilities that the truth label and predicted label of \( x_{th} \) sample belongs to \( c_{th} \) class. Then, we rotate each sample to \( r \) degree and \( r \in CR = \{0^\circ, 90^\circ, 180^\circ, 270^\circ\} \). We define rotation loss as:

\[ \mathcal{L}_r = - \sum_r y_{(r,x)} \log(p_{(r,x)}) \]

(21)

where \( y_{(r,x)} \), \( p_{(r,x)} \) indicate the probabilities that the truth label and predicted label of \( x_{th} \) sample belongs to \( r_{th} \) class. Thus, the first loss function is defined as \( \mathcal{L}_c + \mathcal{L}_r \), and the feature based on this kind of FEM is dubbed as SS-R-Fea.

The second feature is denoted as SS-M-Fea, which extracted from another category of self-supervised FEM. Specifically, this FEM adds the loss \( \mathcal{L}_c \) and auxiliary mirror loss \( \mathcal{L}_m \) to the neural network to predict image mirrors. Assume that there are \( m \) ways and \( m \in \mathcal{C}_M = \{\text{vertically, horizontally, diagonally}\} \), we define the mirror loss as:

\[ \mathcal{L}_m = - \sum_m y_{(m,x)} \log(p_{(m,x)}) \]

(22)

4.4 Multi-Head Feature Extraction Model

The multi-head features we adopted come from different feature extraction models (FEMs). As examples: (1) Standard feature (Std-Fea), the FEM utilizes a standard CNN-based classification structure, such as [47]. (2) Meta feature (Meta-Fea), the FEM introduces the meta-learning strategy to the network, just like [2]. (3) Self-supervised-feature (SS-Fea), the FEM adds auxiliary losses to the standard CNN-based classification structure from a self-supervised perspective to strengthen the robustness of the network, similar as [23]. We discuss the results of all kinds of stacking ways in Appendix B.2.

In this paper, we merely fuse two kinds of SS-Feas for most of the experiments as an example for convenience. For the first category, we design the FEM by introducing standard classification loss \( \mathcal{L}_c \) and auxiliary rotation loss \( \mathcal{L}_r \). \( \mathcal{L}_c \) can be formulated as:

4.3.3 Transductive MHFC. In transductive few-shot learning (TFSL), besides the support data’s features and label information, researchers also apply the features of query data to construct the classifier and then predict the query label. To implement the transductive MHFC, we need to make some adjustments to the step (4) in Section 4.3.1. We re-represent (4) as:

\[ Z_q \leftarrow z_{qn} = \text{Concat} \left( \hat{\Omega}^1 y_{qn}, \hat{\Omega}^2 p_{qn}, \cdots, \hat{\Omega}^H p_{qn} \right) \]

\[ Y_{\text{pseudo}} = W_z z_q \]

(19)

where \( p_{qn}^h (n = 1, 2, \ldots, N) \) indicates the \( n_{th} \) vectors in \( p_q^{th} \). \( z_q \) represents the collaborative feature embedding of the unlabeled set. \( z_{qn} \) denotes the \( n_{th} \) vector in \( z_q \). \( Y_{\text{pseudo}} \) represents the predicted soft-pseudo-label of query data.

After that, we achieve transductive MHFC by steps (1), (2), (3), (9), (5), (6), (8).

4.3.2 Inductive MHFC. Unlike semi-supervised few-shot learning (SSFSL), inductive few-shot learning (IFSL) based methods only employ the support data to train the classifier and predict the query’s category. IFSL can be viewed as the special case of SSFSL (e.g., there not exist unlabeled data). Thus, we can use the steps (e.g., (1), (2), (3), (7), (8)) to implement the inductive MHFC.
where \( y_{(m,x)} \cdot p_{(m,x)} \) indicate the probabilities that the truth label and predicted label of \( x_{th} \) sample belongs to \( m_{th} \) class. Next, we summarize the loss function as \( L_c + L_m \).

5 EXPERIMENTS
In this section, we first briefly review the benchmark datasets and show the implementation details. Then, we list the experimental results in Table 1, 2 and analyse them. Next, we perform ablation studies to discuss the factors that influence MHFC’s performance, e.g., multi-head feature fusion, subspace transformation, and attention block that calculates the combination weights. In the following, we take a cross-domain experiment to further evaluate the ability and robustness of the proposed method. We conduct all the experiments on a Tesla-V100 GPU with 32G memory. All the source codes will be made available to the public.

5.1 Datasets
We carry out experiments on five benchmark datasets, including mini-ImageNet [43], tiered-ImageNet [28], CIFAR-FS [2], FC100 [25], and CUB [44]. Both mini-ImageNet and tiered-ImageNet are the subsets of ImageNet dataset [33]. mini-ImageNet consists of 100 classes and tiered-ImageNet contains 608 classes. For both datasets, the number of images for each class is 600 and the size of each image is \( 84 \times 84 \). We follow standard split as [47], selecting 64 classes as the base set, 16 classes as the novel set, 20 classes as the novel set for mini-ImageNet, and selecting 351 classes as the base set, 97 classes as the validation set, 160 classes as the novel set for tiered-ImageNet. Both CIFAR-FS and FC100 are the subsets of CIFAR-100 dataset [14], and consist of 100 classes. We follow the split introduced in [2] to divide CIFAR-FS into 64 classes as base set, 16 classes as validation set, 20 classes as novel set, and divide FC100 into 60 classes as base set, 20 classes as validation set, 20 classes as novel set. All the image size is \( 32 \times 32 \). CUB totally includes 11,788 images with 200 categories. We follow the setting in ICI [47] to split it into 100 classes as base set, 50 classes as validation set and 50 classes as novel set. The images are cropped into \( 84 \times 84 \).

5.2 Implementation Details
In this paper, all the FEMs on different heads adopt the ResNet12 [11] backbone, consisting of four residual blocks (\( 3 \times 3 \) convolution layer, batch normalization layer, LeakyReLU layer), four \( 2 \times 2 \) max pooling layers, and four dropout layers. We adopt stochastic gradient descent (SGD) optimizer with Nesterov momentum (0.9) for the optimizer. For the parameter \( \eta \) in Equation 6, we fix it to 1.4 for convenience. We set the training epochs to 120 and test over 600 episodes with 15 query samples per class for all the models. Besides, the selected subspace learning methods all follow the default implementation of scikit-learn [26]. And there has no fine-tuning process when classifying the novel data. For other settings, such as the learning rate, data augmentation, filters’ number, we follow the ICI [47].

5.3 Experimental Results
We compare the proposed MHFC (only fuse SS-F-Fea and SS-M-Fea) with several state-of-the-art methods, the results are listed in Table 1 and 2. Here, we list several observations.

(1) Researchers split the few-shot learning methods into three settings, e.g., inductive, transductive, and semi-supervised. While, in the related works of FSL, the results on all settings are usually compared together. Compared with the methods proposed recently, our MHFC has achieved state-of-the-art performance. It has far surpassed other models on the four datasets, especially on 5-way 1-shot case, the MHFC outperforms other methods at least 4.4%, 2.1%, 7.7% and 7.8% on mini-ImageNet, tiered-ImageNet, CIFAR-FS, FC100 datasets. Our results on the 5-way 1-shot case are even better than many other methods on the 5-way 5-shot case. And on the 5-way 5-shot case, the MHFC also exceeds others at least 3.0%, 2.8%, 4.8% and 6.9% on mini-ImageNet, tiered-ImageNet, CIFAR-FS, FC100 datasets.

(2) Compared with the methods of each setting, our proposed MHFC has achieved excellent performances, in particular on the 5-way 1-shot case with the inductive setting. MHFC has significant improvements of at least 7.7%, 10.0%, 6.8% and 4.7% on mini-ImageNet, tiered-ImageNet, CIFAR-FS, FC100 datasets. Notably, the performance of MHFC with the inductive setting has exceeded many other methods with transductive or semi-supervised settings. Besides, for the methods based on semi-supervised setting, the final results are influenced by the number of employed unlabeled samples. Thus, we observe the impact and list the results in Appendix B.1. With the increase of unlabeled samples, the proposed method has become more effective. And the results start to saturate after 100 unlabeled samples.

5.4 Ablation Studies
In this paper, we propose MHFC for few-shot learning. There exist three factors influence the classification performance, e.g., (1) introducing multi-head features; (2) transforming the multi-head features into a unified space; (3) designing attention mechanism to weight features. This section conducts ablation studies on 5-way 1-shot case with the inductive setting to evaluate the efficiency of the three blocks.

Figure 3: Ablation studies to show the performances of employing different kinds of features with the inductive setting.
Table 1: The 5-way few-shot classification accuracies on mini-ImageNet and tiered-ImageNet with 95% confidence intervals over 600 episodes. (-*), (-†), and (-‡) in Table 1, 2, 4 indicate inductive, transductive, and semi-supervised settings, respectively. (-★★) in Table 1, 2 and 3 denotes the non-standardized inductive setting, which adopts the query feature when reducing the feature’s dimension. 4CONV, ResNet12, ResNet18 and WRN are the exploited FEM’s architectures. The (80), (100) in the semi-supervised setting indicate the number of employed unlabeled samples per class. The top two results are shown in red and blue, respectively.

| Method | Backbone | mini-ImageNet | tiered-ImageNet |
|--------|----------|---------------|-----------------|
|        |          | 5-way 1-shot  | 5-way 5-shot     | 5-way 1-shot  | 5-way 5-shot     |
| Baseline* [4] (ICLR,2019) | ResNet18 | 51.75 ± 0.80 | 74.27 ± 0.63 | - | - |
| Baseline++* [4] (ICLR,2019) | ResNet18 | 51.87 ± 0.77 | 75.68 ± 0.63 | - | - |
| TapNet* [32] (ICML,2019) | ResNet12 | 61.65 ± 0.15 | 76.36 ± 0.10 | 63.08 ± 0.15 | 80.26 ± 0.12 |
| LEO* [34] (ICLR,2019) | WRN | 61.76 ± 0.08 | 77.59 ± 0.12 | 66.33 ± 0.05 | 81.44 ± 0.09 |
| AM3* [48] (NIPS,2019) | ResNet12 | 65.30 ± 0.49 | 78.10 ± 0.36 | 69.08 ± 0.47 | 82.58 ± 0.31 |
| CTM* [15] (CVPR,2019) | ResNet18 | 64.12 ± 0.82 | 80.51 ± 0.13 | - | - |
| MABAS* [13] (ECCV,2020) | ResNet12 | 64.21 ± 0.82 | 81.01 ± 0.57 | - | - |
| MEMLR* [9] (ICLR,2021) | ResNet12 | 67.40 ± 0.43 | 83.40 ± 0.28 | 72.14 ± 0.51 | 87.01 ± 0.35 |
| Our MHFC** | ResNet12 | **73.10 ± 1.00** | **81.75 ± 0.56** | **82.10 ± 1.03** | **87.99 ± 0.60** |
| TPN† [22] (ICLR,2019) | 4CONV | 55.51 ± 0.86 | 69.86 ± 0.65 | 59.91 ± 0.94 | 73.30 ± 0.75 |
| TEAM† [27] (ICCV,2019) | ResNet12 | 60.07 ± 0.63 | 75.90 ± 0.52 | - | - |
| Fine-tuning† [5] (ICLR,2020) | WRN | 65.73 ± 0.68 | 78.40 ± 0.52 | **73.34 ± 0.71** | 85.50 ± 0.50 |
| DPGN† [51] (CVPR,2020) | ResNet12 | **67.77 ± 0.32** | **84.60 ± 0.43** | 72.45 ± 0.51 | **87.24 ± 0.39** |
| ODE† [49] (CVPR,2021) | ResNet12 | 67.76 ± 0.46 | 82.71 ± 0.31 | 71.89 ± 0.52 | 85.96 ± 0.35 |
| Our MHFC† | ResNet12 | **74.81 ± 1.12** | **85.58 ± 0.61** | **83.95 ± 1.13** | **90.75 ± 0.58** |
| TPN‡ [22] (ICLR,2019) | 4CONV | 52.78 ± 0.27 | 66.42 ± 0.21 | 55.74 ± 0.29 | 71.01 ± 0.23 |
| LST‡ (100) [16] (NIPS,2019) | ResNet12 | 70.10 ± 1.90 | 78.70 ± 0.80 | 77.70 ± 1.60 | 85.20 ± 0.80 |
| EPNet‡ (100) [30] (ECCV,2020) | ResNet12 | **75.36 ± 1.01** | **84.07 ± 0.60** | 81.79 ± 0.97 | 88.45 ± 0.61 |
| TransMatch‡ (100) [53] (CVPR,2020) | WRN | 63.02 ± 1.07 | 81.19 ± 0.59 | - | - |
| ICI‡ (80) [47] (CVPR,2020) | ResNet12 | 71.41 | 81.12 | **85.44** | **89.12** |
| Our MHFC‡ (80) | ResNet12 | 79.26 ± 1.14 | 87.30 ± 0.55 | 87.57 ± 1.03 | 91.80 ± 0.56 |
| Our MHFC‡ (100) | ResNet12 | **79.76 ± 1.16** | **87.64 ± 0.53** | 87.56 ± 1.04 | **91.90 ± 0.56** |

5.4.1 Influence of Multi-Head Feature. The results listed in Table 1, 2 employ two heads of features (SS-R-Fea and SS-M-Fea). Here, we compare the results of employing the raw single-head feature with the collaborative feature (e.g., multi-head-fea). The results of four datasets on 5-way 1-shot case with inductive setting are listed in Figure 3. Obviously, employing the collaborative feature improves significantly compared with exploiting the raw single-head feature. It demonstrates the efficiency of our “multi-head feature” to some extent. Moreover, the proposed model can integrate more than two heads of features. We list the comparison results in Appendix B.2.

5.4.2 Influence of Subspace Transformation. For all the results listed in Table 1, 2 with inductive setting, we select LE [1] to transform the multi-head features. While, besides the LE, there also exist several other choices to pull multi-head features to a unified space, such as PCA [39], LLE [31]. We illustrate Figure 4 to show the comparison results. We can see that all kinds of subspace learning methods are helpful to our MHFC, and LE is the most.

Figure 4: The comparison results of adding transformation or not on 5-way 1-shot case with the non-standardized inductive setting.
Table 2: The 5-way few-shot classification accuracies on CIFAR-FS and FC100 with 95% confidence intervals over 600 episodes. The top two results are shown in red and blue, respectively.

| Method             | Backbone | CIFAR-FS  | FC100   |
|--------------------|----------|-----------|---------|
|                    |          | 5-way 1-shot | 5-way 5-shot | 5-way 1-shot | 5-way 5-shot |
| ProtoNet* [36]     | 4CONV    | 55.50 ± 0.70 | 72.00 ± 0.60 | 35.30 ± 0.60 | 48.60 ± 0.60 |
| MAML* [10]        | 4CONV    | 58.90 ± 1.90 | 71.50 ± 1.00 | -            | -            |
| MABAS* [13]       | ResNet12 | 73.24 ± 0.95 | 83.65 ± 0.65 | 41.74 ± 0.73 | 57.11 ± 0.75 |
| Our MHFC**        | ResNet12 | 80.00 ± 1.02 | 86.33 ± 0.61 | 46.44 ± 0.93 | 59.41 ± 0.72 |
| TEAM† [27]        | ResNet12 | 70.43 ± 1.03 | 81.25 ± 0.92 | -            | -            |
| Fine-tuning§ [5]  | WRN      | 76.58 ± 0.68 | 85.79 ± 0.50 | 43.16 ± 0.59 | 57.57 ± 0.55 |
| Our MHFC†         | ResNet12 | 81.83 ± 1.16 | 89.27 ± 0.63 | 47.02 ± 1.05 | 61.88 ± 0.80 |
| ICI‡ (80) [47]    | ResNet12 | 78.07       | 84.76     | -            | -            |
| Our MHFC‡ (80)    | ResNet12 | 84.74 ± 1.14 | 90.19 ± 0.63 | 50.95 ± 1.11 | 64.05 ± 0.82 |
| Our MHFC‡ (100)   | ResNet12 | 85.76 ± 1.08 | 90.42 ± 0.64 | 50.72 ± 1.09 | 64.45 ± 0.83 |

Table 3: Comparison results with fixed weights on 5-way few-shot case. (a, b) denotes that the SS-R-Fea’s weight is “a”, and SS-M-Fea’s weight is “b”. MHFC employs our designed attention block to update the weights automatically for each episode. The top two results are shown in red and blue.

| Weight     | mini-ImageNet 1-shot | mini-ImageNet 5-shot | tiered-ImageNet 1-shot | tiered-ImageNet 5-shot |
|------------|----------------------|----------------------|------------------------|------------------------|
| 0.1, 0.9   | 70.85                | 80.09                | 79.66                  | 87.12                  |
| 0.3, 0.7   | 71.32                | 80.72                | 80.25                  | 87.14                  |
| 0.5, 0.5   | 71.88                | 80.15                | 80.86                  | 87.38                  |
| 0.7, 0.3   | 70.56                | 80.48                | 80.19                  | 87.19                  |
| 0.9, 0.1   | 70.36                | 79.97                | 79.69                  | 86.93                  |
| MHFC**     | 73.10                | 81.75                | 82.10                  | 87.99                  |

5.4.3 Influence of Attention Block. We can use multi-head features to describe a category of samples, but the degrees of importance are different on each head. To this end, it is crucial to design the attention block to calculate weights for different categories automatically. Here, we compare the results with fixed weights to our MHFC, which is listed in Table 3. The results show that the updated weights are more reasonable for our method. Besides, from Equation 6, we know that $\eta$ is a parameter to influence the to-be-learned weights. For fairness and convenience, we have fixed $\eta$ to 1.4 for all the experiments. We list the other comparison results in Appendix B.3.

5.5 Cross-Domain Few-Shot Learning

After introducing multi-head features from different views, we believe that the MHFC is an extremely robust method in practical scenarios. Therefore, we evaluate the proposed method with transductive setting on a cross-domain dataset: e.g., mini-ImageNet $\rightarrow$ CUB. In pre-train stage, we use mini-ImageNet to train the FEM, and in meta-test stage, we classify the CUB dataset. The results are reported in Table 4. Compared to the state-of-the-arts, we have significant improvements at least 13.3% on 1-shot case and 7.8% on 5-shot case. Thus, the performance on the cross-domain few-shot learning task demonstrates that the MHFC can solve the DSP better, and the proposed MHFC would be powerful in real practice.

Table 4: Comparison in cross-domain dataset scenario. Our MHFC is in transductive setting. (·)§ and (·)§ indicate the reported results come from [3] and [23], respectively. The top two results are shown in red and blue.

| Method             | mini-ImageNet 5-way 1-shot | mini-ImageNet 5-way 5-shot |
|--------------------|-----------------------------|---------------------------|
| Baseline§ [4]      | -                           | 53.1                      |
| ProtoNet§ [36]     | -                           | 62.0                      |
| RelationNet§ [37]  | -                           | 57.7                      |
| GNN§ [40]          | -                           | 66.9                      |
| Neg-Cosine§ [18]   | -                           | 67.0                      |
| LaplacianShot§ [56]| -                           | 66.3                      |
| TIM-GD§ [3]        | -                           | 71.0                      |
| MetaOpt§ [2]       | 44.79 ± 0.75                | 64.98 ± 0.68              |
| Manifold Mixup§ [42]| 46.21 ± 0.77                | 66.03 ± 0.71              |
| S2M2§ [23]         | 48.24 ± 0.84                | 70.44 ± 0.75              |
| MHFC§              | 61.57 ± 1.28                | 78.80 ± 0.78              |

6 CONCLUSION

Few-shot learning (FSL) based tasks have a fundamental problem, e.g., distribution-shift-problem (DSP). To address this challenge, we propose Multi-Head Feature Collaboration (MHFC), which attempts to collaboratively represent samples by fusing multi-head
features. It is helpful to strengthen the FSL based model’s efficacy and robustness. MHFC is a simple non-parametric method that can directly employ the existing FEMs. Experimental results have demonstrated the effectiveness of MHFC.

ACKNOWLEDGMENT
The paper was supported by the National Natural Science Foundation of China (Grant No. 62072468), the Natural Science Foundation of Shandong Province, China (Grant No. ZR2019MF073), the Fundamental Research Funds for the Central Universities, China University of Petroleum (East China) (Grant No. 20CX05001A), the Graduate Innovation Project of China University of Petroleum (East China) (YCX2021117, YCX2021123).

REFERENCES
[1] Mikhail Belkin and Partha Niyogi. 2002. Laplacian eigenmaps and spectral techniques for embedding and clustering. In NeurIPS 585–591.
[2] Luca Bertinetto, Joao F. Henriques, Philip Torr, and Andrea Vedaldi. 2019. Meta-learning with differentiable closed-form solvers. In ICLR.
[3] Mamshad Nayeem Rizve, Salman Khan, Fahad Shabbaz Khan, and Mubarak Shah. 2021. Exploring Complementary Strengths of Invariant and Equivariant Representations for Few-Shot Learning. In ICLR.
[4] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. 2019. A closer look at few-shot classification. In ICLR.
[5] Guneet S Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto. 2019. Meta-learning via modeling episode-level relationships for few-shot learning. In TMM 23 (2020), 1581–1591.
[6] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. 2011. The Caltech-UCSD Birds-200-2011 Dataset. In IJCV 82 (2009), 133–150.
[7] Nikita Dvornik, Cordelia Schmid, and Julien Mairal. 2020. Selecting relevant features from a multi-domain representation for few-shot classification. In ICLR.
[8] Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In NeurIPS 4077–4087.
[9] đổ Sung, Tong Xiang, Qing Liu, Xiao Yuan, and Tao Xiang. 2018. Learning to compare: Relation network for few-shot learning. In NeurIPS 12836–12845.
[10] Sheng Chen, Xingyu Tan, and Yanbing Fan. 2020. instance-level credibility inference for few-shot learning. In NeurIPS 10981–10990.
[11] Ling Yang, Liangliang Li, Zilun Zhang, Xinyu Zhou, Erjin Zhou, and Yu Liu. 2020. Learning to propagate labels: Transductive propagation network for few-shot learning. In ICLR.
[12] Bin Liu, Yue Cao, Yutong Lin, Qi Li, Zheng Zhang, Mingsheng Long, and Han Hu. 2020. Negative margin matters: Understanding margin in few-shot classification. In NeurIPS 488–495.
[13] Luo Li, Jie Meng, Wen-Yang Xie, Shuai Shao, Ye Li, and Yanjiang Wang. 2019. Weighted spatial pyramid matching for few-shot learning. In NeurIPS.
[14] Zhihua Zhang, Shuicheng Yan, and Ming-Yu Liu. 2020. Multi-modal hybrid model for human action recognition. In IEEE Journal of Biomedical and Health Informatics 24, 4 (2020), 1224–1235.
[15] Zhang, and Jiashi Feng. 2020. Learning to self-train for semi-supervised few-shot classification. In NeurIPS.
[16] Rui Yan, Weifeng Si, and Jie Sun. 2020. Deep Residual Networks for few-shot learning. In ICLR 1–10.
[17] Xingye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. 2019. Meta-learning with latent embedding techniques for embedding and clustering. In NeurIPS 585–591.
[18] Xiaohui Tang, Shaochun Zhu, Xiaojun Chang, and Bo Huang. 2021. Exploring Complementary Strengths of Invariant and Equivariant Representations for Few-Shot Learning. In ICLR.
[19] Jianwei Chen, Xiaohui Tang, and Shaochun Zhu. 2021. Learning to compare: Relation network for few-shot learning. In NeurIPS. 122–131.
[20] Hong-Yu Tseng, Hsin-Ying Lee, Jia-Bin Huang, and Ming-Hsuan Yang. 2020. A baseline for few-shot image classification. In ICLR.
[21] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. 2020. Rethinking few-shot image classification: a good embedding is all you need? In CVPR.
[22] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. 2011. The Caltech-UCSD Birds-200-2011 Dataset. In IJCV 82 (2009), 133–150.
[23] Ling Yang, Liangliang Li, Zilun Zhang, Xinyu Zhou, Erjin Zhou, and Yu Liu. 2020. Learning to propagate labels: Transductive propagation network for few-shot learning. In ICLR.
[24] Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In NeurIPS 4077–4087.
[25] Fengyang Li, Zhihuan Liu, Wei Wang, and Xiaoying Zhang. 2019. A closer look at few-shot classification. In ICLR.
[26] Yikai Wang, Chengming Xu, Chen Liu, Li Zhang, and Yanwei Fu. 2020. A baseline for few-shot image classification. In ICLR.
[27] Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In NeurIPS 4077–4087.
[28] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. 2019. Meta-learning with latent embedding techniques for embedding and clustering. In NeurIPS 585–591.
[29] Xiaohui Tang, Shaochun Zhu, Xiaojun Chang, and Bo Huang. 2021. Exploring Complementary Strengths of Invariant and Equivariant Representations for Few-Shot Learning. In ICLR.
[30] Jianwei Chen, Xiaohui Tang, and Shaochun Zhu. 2021. Learning to compare: Relation network for few-shot learning. In NeurIPS.
[31] Liu, Yue Cao, Yutong Lin, Qi Li, Zheng Zhang, Mingsheng Long, and Han Hu. 2020. Negative margin matters: Understanding margin in few-shot classification. In NeurIPS 488–495.
[32] Bao-Di Liu, Jie Meng, Wen-Yang Xie, Shuai Shao, Ye Li, and Yanjiang Wang. 2019. Weighted spatial pyramid matching for few-shot learning. In NeurIPS 488–495.
[32] Sung Whan Yoon, Jun Seo, and Jaekyun Moon. 2019. Tapnet: Neural network augmented with task-adaptive projection for few-shot learning. In ICML. 7115–7123.

[33] Zhongjie Yu, Lin Chen, Zhongwei Cheng, and Jiebo Luo. 2020. TransMatch: A Transfer-Learning Scheme for Semi-Supervised Few-Shot Learning. In CVPR. 12856–12864.

[34] Zhongqi Yue, Hanwang Zhang, Qianru Sun, and Xian-Sheng Hua. 2020. Interventional few-shot learning. In NeurIPS.

[35] Zizhao Zhang, Haojie Lin, Xibin Zhao, Rongrong Ji, and Yue Gao. 2018. Inductive multi-hypergraph learning and its application on view-based 3D object classification. TIP 27, 12 (2018), 5957–5968.

[36] Imtiaz Ziko, Jose Dolz, Eric Granger, and Ismail Ben Ayed. 2020. Laplacian regularized few-shot learning. In ICML. 11660–11670.