Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 9620755, 12 pages
https://doi.org/10.1155/2022/9620755

Research Article

Hybrid Fine-Tuning Strategy for Few-Shot Classification

Lei Zhao,[1,2] **Zhonghua Ou**,[2] **Lixun Zhang**,[2] **and Shuxiao Li**[1]

[1]*Institute of Automation, Chinese Academy of Sciences, Beijing, China*
[2]*University of Electronic Science and Technology of China, Chengdu, China*

Correspondence should be addressed to Shuxiao Li; shuxiao.li@ia.ac.cn

Received 5 July 2022; Revised 16 September 2022; Accepted 23 September 2022; Published 8 October 2022

Academic Editor: Nouman Ali

Few-shot classification aims to enable the network to acquire the ability of feature extraction and label prediction for the target categories given a few numbers of labeled samples. Current few-shot classification methods focus on the pretraining stage while fine-tuning by experience or not at all. No fine-tuning or insufficient fine-tuning may get low accuracy for the given tasks, while excessive fine-tuning will lead to poor generalization for unseen samples. To solve the above problems, this study proposes a hybrid fine-tuning strategy (HFT), including a few-shot linear discriminant analysis module (FSLDA) and an adaptive fine-tuning module (AFT). FSLDA constructs the optimal linear classification function under the few-shot conditions to initialize the last fully connected layer parameters, which fully excavates the professional knowledge of the given tasks and guarantees the lower bound of the model accuracy. AFT adopts an adaptive fine-tuning termination rule to obtain the optimal training epochs to prevent the model from overfitting. AFT is also built on FSLDA and outputs the final optimum hybrid fine-tuning strategy for a given sample size and layer frozen policy. We conducted extensive experiments on mini-ImageNet and tiered-ImageNet to prove the effectiveness of our proposed method. It achieves consistent performance improvements compared to existing fine-tuning methods under different sample sizes, layer frozen policies, and few-shot classification frameworks.

1. Introduction

Deep learning has recently attracted attention due to its outstanding performances in computer vision (e.g., image classification and object detection), NLP, and reinforcement learning. In the military domain, unmanned aerial vehicles (UAVs) play a significate role in jamming and reconnaissance. Bai et al. [1] established a 3D UAV air combat model and a UAV maneuvering decision algorithm based on deep reinforcement learning to achieve autonomous operation of UAVs in the future. Saqlain et al. [2] applied deep learning and computer vision to retail management to boost retail sales, proposing a hybrid approach that can effectively monitor retail shelves and satisfy planograms. In face recognition systems, Yang and Song [3] improved the face recognition effect in different light intensities combined with the deep learning algorithm, which is of great practical value.

The success of deep learning is mainly attributed to the following three factors, i.e., powerful computing resources, complex network frameworks, and large-scale datasets. However, obtaining sufficient labeled data in many application scenarios, such as rare diseases, new species, and defective industrial products, is difficult or even impossible. When the annotated data are scarce, traditional deep learning methods generally perform unsatisfactorily. Considering that humans can rapidly establish cognition to novel concepts from just a single or a handful of examples, we hope the network can acquire the ability to recognize visual objects for novel classes with high accuracy and generalization by learning from only a few samples.

Towards the goal of shrinking the gap between human intelligence and artificial intelligence, few-shot learning, especially few-shot classification (FSC), was proposed. FSC aims to learn an effective classifier from the target dataset, which only contains a few labeled images for novel classes. However, different from general deep learning, it is impossible to train an effective classification model from scratch only using the target dataset due to its limited capacity. Therefore, current FSC methods usually employ a base dataset, which contains abundant labeled images for base classes and has no category intersection with the target dataset. The model is firstly pretrained on the base dataset to learn a feature extractor and then is transferred to the target domain for fine-tuning to boost the performance of FSC. At the pretraining stage, the feature extractor is pretrained either on the base dataset directly or by meta-learning which constructs massive few-shot tasks to imitate the target scenarios. As for the fine-tuning stage, current methods always choose the fine-tuning settings relying on experience, e.g., how to set the learning rate, which layers are selected to be frozen, and how many training epochs to be set. They prefer to set the learning rate as 0.001 [4, 5], usually select linear probing (updating only the last linear layer) [6] or full fine-tuning (updating all the model parameters) [7–9], and rarely mention how many training epochs are set. Since there are no validation and test images in the target dataset, it is impossible to evaluate the performance of the fine-tuned model, so how to set hyperparameters beyond experience remains a problem. In addition, the classifier parameters will also be quickly converged to a nonoptimal solution under few-shot conditions, which further reduces the classification performance.

To address the problems mentioned above, in this work, we propose a hybrid fine-tuning strategy (HFT) for FSC, as shown in Figure 1. We first pretrain on the base dataset to get the pretrained model and then fine-tune it on the target dataset according to the acquired hybrid fine-tuning strategy by HFT. The proposed HFT includes an FSLDA module and an AFT module. FSLDA constructs the optimal linear classifier by fully excavating the professional knowledge of the target dataset, which provides the last fully connected layer of the pretrained model a better starting point that fine-tuning with backpropagation probably cannot reach, thus guaranteeing the lower bound of the model accuracy. AFT executes adaptive epoch learning using the validation classes of the base dataset by designing an adaptive fine-tuning termination rule to obtain the optimal training epochs. Therefore, AFT sets hyperparameters by learning instead of experience and can prevent the model from overfitting. AFT also implements model performance evaluation to obtain the hybrid fine-tuning strategy. Finally, we update the pretrained model with the acquired hybrid fine-tuning strategy using the target dataset to get the HFT model. In summary, the main contributions of this study are as follows:

(1) We improve linear discriminant analysis for FSC and propose the FSLDA module, which can be used to initialize the last fully connected layer parameters of the pretrained model and guarantees the lower bound of the model accuracy. Ablation studies on mini-ImageNet dataset show that the Meta-Baseline method [10] with the FSLDA module alone has an average performance improvement of 3.07% and 2.99% under the layer frozen policy "Last1" and "All," respectively.

(2) We introduce adaptive epoch learning to the fine-tuning stage and propose the AFT module, which can prevent the model from overfitting and output the hybrid fine-tuning strategy under different sample sizes and different layer frozen policies. Ablation results on mini-ImageNet dataset show that the Meta-Baseline method [10] with AFT under the layer frozen policy "All" further brings 0.40%, 0.99%, and 0.79% performance improvements for sample sizes of 10-shot, 20-shot, and 30-shot, respectively.

(3) The acquired hybrid fine-tuning strategy is evaluated under three recently proposed few-shot classification methods. Comparative experiments show that the proposed HFT has an average performance improvement of 2.30% on the mini-ImageNet dataset and 2.78% on the tiered-ImageNet dataset over current experience-based finetuning methods.

2. Related Works

2.1. Few-Shot Classification. Currently, many works have been proposed to address FSC [11–19], which can be mainly divided into three categories: initialization-based methods, metric-based methods, and hallucination-based methods. Initialization-based methods use the target dataset to fine-tune the pretrained model with a small number of gradient backpropagation steps [20, 21]. Metric-based methods extract features from both the labeled and unlabeled images and predict the class labels by computing the similarity metric function, such as cosine similarity [22], Euclidean distance [23], and relation modules [24]. Hallucination-based methods [25] focus on data augmentation by learning a generator from the base dataset and applying it to novel classes to expand the capacity of the target dataset. Recently, some works have employed self-supervision [26, 27], knowledge distillation [28, 29], and distribution calibration [30, 31] to strengthen the feature extractor or the last classifier. Our work is built on the metric-based pretraining methods and improves the initialization-based fine-tuning methods by introducing a hybrid fine-tuning strategy.

2.2. Fine-Tuning Strategy. Before fine-tuning the model with the target dataset, key hyperparameters need to be set, such as the layer frozen policy, the learning rate, and the training epochs. Due to the scarcity of the target dataset, we cannot judge whether the model is suboptimal, overfitted, or underfitted. Thus, current methods usually set the above hyperparameters by experience. There are two popular strategies for the layer frozen policy: running gradient descent on all model parameters [7–9] and fine-tuning the head but freezing lower layers [32]. Some works [33, 34] claim that fine-tuning all model parameters leads to better accuracy than only fine-tuning the head, while most researchers have no consistent conclusions about this. For the learning rate, the mainstream methods [35, 36] on FSC select to set it as 0.001. As for the training epochs, current methods use fixed settings, and their value is rarely mentioned. Recently, an evolutionary algorithm [37] has been proposed for searching the best finetuning configuration, focusing on the learning rate and the layer frozen policy. Our work emphasizes learning the best training epochs, which is essential to prevent the model from overfitting or underfitting and is

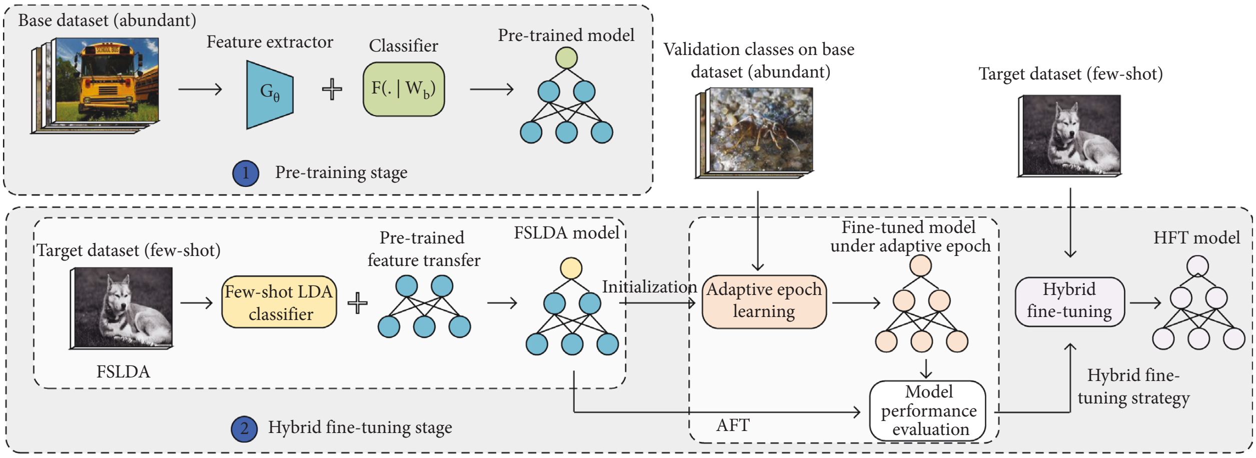

Base dataset (abundant)
Feature extractor
Classifier
Pre-trained model
G_θ
$F(.\mid W_b)$
1 Pre-training stage
Validation classes on base dataset (abundant)
Target dataset (few-shot)
Target dataset (few-shot)
Few-shot LDA classifier
Pre-trained feature transfer
FSLDA model
FSLDA
Initialization
Adaptive epoch learning
Fine-tuned model under adaptive epoch
Model performance evaluation
AFT
Hybrid fine-tuning
HFT model
Hybrid fine-tuning strategy
2 Hybrid fine-tuning stage

Figure 1: Main idea and flowchart of the proposed HFT method for FSC. HFT performs the fine-tuning process based on the pretrained model. It includes an FSLDA module and an AFT module. FSLDA constructs the optimal linear classifier under the few-shot conditions to get the FSLDA model. AFT executes adaptive epoch learning and model performance evaluation using the validation classes of the base dataset to obtain the hybrid fine-tuning strategy, which is finally adopted for fine-tuning the pretrained model using the target dataset to get the HFT model.

complementary to the work in [37]. In addition, we propose the FSLDA module to construct the optimal linear classifier for FSC to avoid suboptimal solutions.

3. Methods

This section first introduces the preliminary foundations, including problem definition and model pretraining for FSC. We then give the technical details for the FSLDA and AFT modules, respectively.

3.1. Preliminary Foundations

3.1.1. Problem Definition. In the standard FSC task, we generally have a base dataset $\mathscr{D}_b$ and a target dataset $\mathscr{D}_n$. Generally, $\mathscr{D}_b$ contains abundant labeled samples for base classes, while $\mathscr{D}_n$ has only a few labeled samples for novel classes (usually 1 to 30 for each class). Denote $\mathscr{C}_b$ and $\mathscr{C}_n$ as the category spaces of base classes and novel classes, respectively, which are nonoverlapping, i.e., $\{\mathscr{C}_b \cap \mathscr{C}_n\} = \varnothing$. Let $\mathscr{N}_b$ and $\mathscr{N}_n$ denote the number of samples in the base and the target datasets, respectively. With these definitions, $\mathscr{D}_b$ and $\mathscr{D}_n$ can be further denoted as $\mathscr{D}_b = \{(x_i, l_i) | l_i \in \mathscr{C}_b\}_{i=1}^{\mathscr{N}_b}$ and $\mathscr{D}_n = \{(x_j, l_j) | l_j \in \mathscr{C}_n\}_{j=1}^{\mathscr{N}_n}$, where x represents the sample in the dataset and l indicates the label that the sample was annotated with. The goal of FSC is to train models with $\mathscr{D}_b$ and $\mathscr{D}_n$ for predicting the labels of samples in the test dataset of novel classes. Specifically, considering a C-way K-shot metric-based meta-learning FSC task, massive meta-learning tasks, each of which includes a support set and a query set, are randomly sampled from the base dataset to imitate the target task. The support set consists of C classes with K labeled samples in each class, and the corresponding query set has the same classes as the support set, each of which has Q unlabeled samples. The goal of metric-based meta-learning is to update the model to predict the labels of the $C \times Q$ samples in the query set by computing their similarities to the support set. Through continuous learning from massive meta-learning tasks, the pretrained model can memorize more scene knowledge and thus has better generalization ability for FSC tasks.

3.1.2. Model Pretraining. A fundamental step for FSC is pretraining the model on the base dataset to provide a suitable feature extractor G_θ. Specifically, the model is firstly trained with standard cross-entropy loss on the base dataset for all the classes to get the initialized model. Then, metric-based meta-learning is performed to continually train the model by building massive C-way K-shot tasks, finally outputting the pretrained model. This scheme can help the model improve its stability and generalization ability by imitating the few-shot settings that will be encountered in the target task. In fact, the proposed fine-tuning method in this study only uses the parameters of the pretrained model, which has nothing to do with the pretraining method. Thus, other pretraining methods based on different theories are also applicable.

3.2. Few-Shot LDA Module. Linear discriminant analysis (LDA) is a dimensionality reduction technique for supervised learning and is mainly used for classification. The core idea of LDA is to project high-dimensional data samples into the best vector space so that interclass distances are larger and intraclass distances are smaller in the new subspace. LDA needs to calculate the covariance matrix using the feature vectors of data samples in the support set or the target dataset. For FSC tasks, the feature dimension is usually larger than the number of data samples; thus, the covariance matrix is irreversible. To address this issue, FSLDA is proposed to initialize the head of the pretrained model by constructing the optimal linear classification function under few-shot conditions. As shown in Figure 2, we introduce the rank factor α, which is related to the feature dimension d and

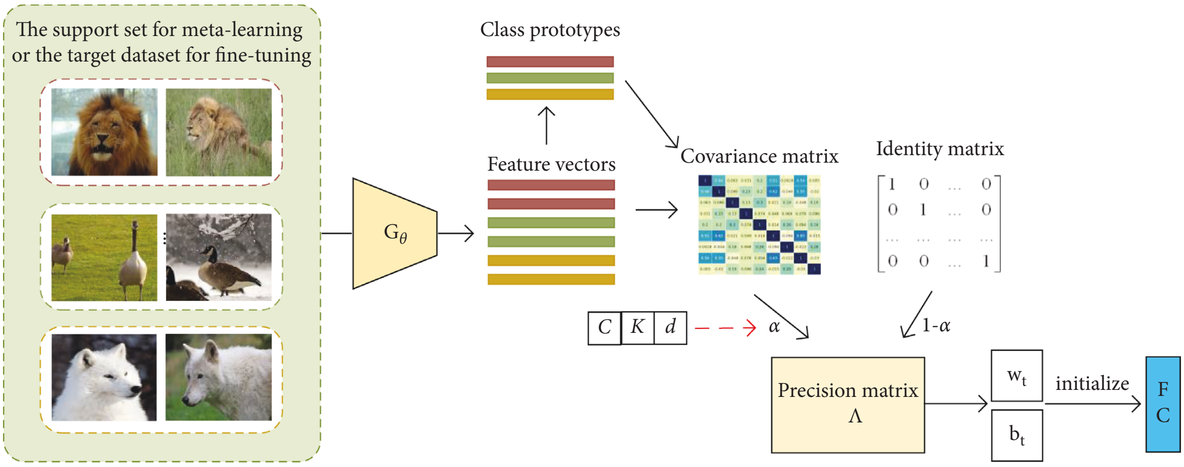


Figure 2: Diagram of the proposed FSLDA module. Given a C-way K-shot support set or target dataset, we first get the feature vector for each sample, the prototype for each class, and the covariance matrix for all feature vectors sequentially. Then, the rank factor α is introduced to obtain the precision matrix Λ for FSC tasks based on the weighted mean of the covariance matrix and the identity matrix. Finally, we obtain the parameter value of the last fully connected layer by Λ and initialize it.

the number of data samples, to illustrate the reliability of the covariance matrix. Based on the rank factor α, the weighted mean of the covariance matrix and the identity matrix is computed instead to obtain the precision matrix so that the invertible condition can be satisfied. By doing so, we get the optimal solution of the FSLDA classifier, which fully excavates the professional knowledge of the given tasks.

Formally, the CNN model we train can be expressed as $y_i = F(G(x_i))$, where x_i is the input sample and y_i is the predicted class label. We decompose the network into two nested functions: the feature extractor denoted as $G(\cdot|\theta_G)$ and the last fully connected layer denoted as $F(\cdot|\theta_F)$. The goal of FSLDA is to initialize the parameters θ_F of $F(\cdot|\theta_F)$, which can be formulated as

$$F(G(x_i)) = Wv + b, \tag{1}$$

where $v \in \mathbb{R}^d$ denotes the output of feature extractor $G(\cdot|\theta_G)$ for the input sample x_i, $W \in \mathbb{R}^{c\times d}$ and $b \in \mathbb{R}^c$ are, respectively, the weight matrix and the bias vector of $F(\cdot|\theta_F)$, d is the output dimension of feature extractor $G(\cdot|\theta_G)$, and c is the number of classes.

According to the LDA theory (details are shown in the Appendix section), given a C-way K-shot task, the optimal linear classifier for class t is given by

$$\begin{aligned} f_t(v) &= \mu_t^T\Sigma^{-1}v - \frac{1}{2}\mu_t^T\Sigma^{-1}\mu_t, \\ \mu_t &= \frac{1}{K}\sum_{i=1}^{K} G(x_t^i), \\ \Sigma &= \frac{1}{C\cdot(K-1)}\sum_{i=1}^{K}\sum_{t=1}^{C}\left[G(x_t^i)-\mu_t\right]\cdot\left[G(x_t^i)-\mu_t\right]^{\mathrm{T}}, \end{aligned} \tag{2}$$

where x_t^i denotes the ith sample for the tth class, μ_t is the mean feature vector (also called the prototype) for class t, and Σ is the covariance matrix of the whole dataset. It can be seen that the rank of the covariance matrix Σ is $C\cdot(K-1)$ for nonlinear data samples, which is usually smaller than the feature dimension d. Thus, the covariance matrix is irreversible and LDA cannot be directly used for FSC tasks.

To this end, we compute the precision matrix Λ directly based on the covariance matrix Σ by harmonic weighting, i.e.,

$$\begin{aligned} \Lambda &= [\alpha\cdot\Sigma + (1-\alpha)\cdot I]^{-1}, \\ \alpha &= 1 - \mathrm{ReLU}\left(1 - \frac{C\cdot(K-1)}{d}\right), \end{aligned} \tag{3}$$

where $I \in \mathbb{R}^{d\times d}$ is the identity matrix and α is the rank factor to measure the reliability of the covariance matrix Σ, making the precision matrix Λ both reversible and informative. When K equals 1, α gets the value of 0 and FSLDA degenerates into prototype initialization. For non-FSC tasks (K is sufficiently large), α gets the value of 1 and FSLDA degenerates into LDA. Thus, prototype initialization and LDA are special cases of FSLDA.

Once the precision matrix Λ is available, FSLDA classifier can be constructed as

$$f_t(v) = \mu_t^T\Lambda v - \frac{1}{2}\mu_t^T\Lambda\mu_t. \tag{4}$$

Finally, we use FSLDA classifier to compute w_t, i.e., the rows of W, and b_t, i.e., the individual elements of b, as

$$\begin{aligned} w_t &= \mu_t^T\Lambda, \\ b_t &= -\frac{1}{2}\mu_t^T\Lambda\mu_t. \end{aligned} \tag{5}$$

The FSLDA enables to initialize the parameters in $F(\cdot|\theta_F)$ by computing the precision matrix Λ of the samples in novel classes before fine-tuning, which gives the model a better initial point than random initialization. By leveraging the knowledge of samples in novel classes and optimizing it for the classifier, the FSLDA

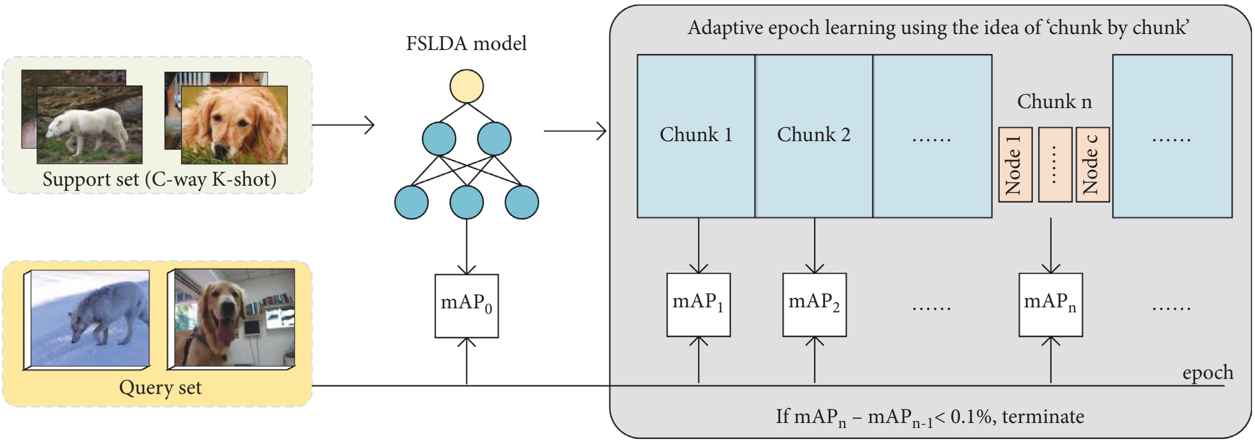

Support set (C-way K-shot)
FSLDA model
mAP$_0$
Query set
Adaptive epoch learning using the idea of 'chunk by chunk'
Chunk 1
Chunk 2
......
Chunk n
Node 1
......
Node c
......
mAP$_1$
mAP$_2$
......
mAP$_n$
......
epoch
If mAP$_n$ – mAP$_{n-1}$< 0.1%, terminate

Figure 3: Illustration of adaptive epoch learning. Once the FSLDA model is available, we fine-tune it with the support set using the idea of "chunk by chunk" and get the corresponding sequential mAP with the query set. The fine-tuning process terminates if the accuracy gain is negligible. Note that adaptive epoch learning runs on the validation classes of the base dataset.

ensures a lower bound on the model's performance and makes the model converge quickly for the fine-tuning stage.

3.3. Adaptive Fine-Tuning Module. Drawing on the experience of meta-learning-based pretraining methods, we propose the AFT module to obtain the hybrid finetuning strategy. AFT firstly performs adaptive epoch learning using the idea of "chunk by chunk" on the validation classes of the base dataset, which evaluates the model's performance for each chunk and establishes an adaptive termination rule to output an adaptive epoch that needs to be set at the fine-tuning stage. Then, the higher one between the FSLDA model and the adaptive fine-tuned model is retained, and the optimal hybrid epoch is acquired. Finally, the above procedures are executed on massive pseudofine-tuning tasks to output the final hybrid fine-tuning strategy, ensuring that most tasks converge to higher performance.

Specifically, massive pseudofine-tuning tasks, each of which includes a support set and a query set, are randomly sampled from the validation classes of the base dataset to imitate the fine-tuning task. Like metric-based meta-learning, the support set here is also of the C-way K-shot style. All the remaining samples in the selected classes are used as the query set to evaluate the performance of the model. As shown in Figure 3, we first use the support set to get the FSLDA model and obtain its accuracy mAP_m^0 using the query set. During adaptive epoch learning, we divide the maximum allowable epochs into N chunks, and each chunk contains c nodes. To improve the learning speed, only the model at the last epoch in each node is evaluated by the query set to get its accuracy. We regard the mean of all nodes' performance in a chunk as a representation of the chunk's performance, so as to get the macrochange trend of the accuracy curve. For the mth pseudofine-tuning task, we can get its "chunk by chunk" performance series, denoted as $\{\mathrm{mAP}_m^0, \ldots, \mathrm{mAP}_m^b, \mathrm{mAP}_m^{b+1}, \mathrm{mAP}_m^n, \cdots\}$, where b is the starting evaluation chunk index to avoid disturbances at the initial fine-tuning stage. The process terminates if the accuracy gain is negligible and outputs the adaptive chunk index:

$$Iter_m = \min_n \left\{ mAP_m^n - mAP_m^{n-1} < 0.1\% \right\}, n \in [b, N]. \tag{6}$$

Then, we combine the advantages of the FSLDA model and the adaptive epoch learning and set the optimal hybrid epoch as

$$\mathrm{epoch}_m = \left\{ \begin{array}{ll} a \cdot \mathrm{Iter}_m, & \mathrm{mAP}_m^{\mathrm{Iter}_m} > \mathrm{mAP}_m^0, \\ 0, & \text{otherewise}, \end{array} \right\}, \tag{7}$$

where a is the number of epochs contained in a chunk.

When the optimal hybrid epochs for M pseudofine-tuning tasks are ready, the optimal hybrid finetuning strategy can be finally acquired by

$$\mathrm{epoch} = \begin{cases} \mathrm{Quantile}\left(\{\mathrm{epoch}_m\}, 0.9\right), & M' > \dfrac{M}{2}, \\ 0, & \text{otherwise}, \end{cases} \tag{8}$$

where $M' = \sum_{m=1}^{M} 1\left(\mathrm{epoch}_m\right)$ indicates the number of tasks needing to be fine-tuned, and 1 is the indicator function. When most pseudofine-tuning tasks do not need the fine-tuning stage (epoch = 0), the optimal hybrid fine-tuning strategy adopts FSLDA as the final strategy. Otherwise, it uses the 0.9 quantile of the optimal hybrid epochs to ensure that most tasks can be converged. In the latter case, the optimal hybrid fine-tuning strategy performs both FSLDA and AFT.

The pipeline for AFT is summarized as Algorithm 1.

4. Experiments

In this section, we first briefly describe the experimental setup. Then, HFT experiments are carried out to give the hands-on hybrid fine-tuning strategy under different sample sizes and layer frozen policies. Finally, extensive comparison

```
Input: the validation dataset: val_$dl$, the pretrained model model$^0$
Output: hybrid fine-tuning strategy represented by adaptive epoch: epoch
  Hyper-parameters: the total number of pseudofine-tuning tasks $M$, the maximum number of epochs $E^{\max}$, the number of epochs
contained in a chunk $a$, the number of nodes contained in a chunk $c$, and the starting chunk number $b$.
  for= 1: $M$ do
    $dl$_train$_m$, $dl$_test$_m$ = RandomTaskSample (val_$dl$); #get train and test sets for task $m$
    mo del$_m^0$ = FSLDA ($dl$_train$_\mathrm{m}$, mo del$^0$); #initialize the model by FSLDA
    mAP$_m^0$ = Evaluate (model$_m^0$, $dl$_test$_m$);
    $N = E^{\max}/a$; #number of chunks
    mo del$_m^{b-1}$ = Backpropagation ($dl$_train$_m$, $(b-1)*a$); #train the model by $(b-1)*a$ epochs
    for each chunk $n = b$: $N$ do
      for each node $j = 1$: $c$ do
        mo del$_m^{n,j}$ = Backpropagation ($dl$_train$_m$, $a/c$);
        mAP$_m^{n,j}$ = Evaluate (model$_m^{n,j}$, $dl$_test$_m$);
      end for
      mAP$_m^n = 1/c \sum_{j=1}^{c}$ mAP$_m^{n,j}$; #the average accuracy of chunk $n$
      ifmAP$_m^n$ − mAP$_m^{n-1} < 0.1\%$ or $n = N$ then
        epoch$_m = n \cdot a$; #adaptive epochs
        if mAP$_m^0$ > mAP$_m^n$ thenepoch$_m$ = 0; #optimal hybrid epoch
        break;
      end if
    end for
  end for
  Set $s = \{\text{epoch}_m | m = 1, 2, \cdots M\}$, $M' = 1(s)$. #number of tasks needing finetuning
  if $M' > M/2$ then epoch = Quantile$(s, p = 0.9)$; #get the $p$ quantile of $s$
  else epoch = 0.
```

Algorithm 1: Pseudocode for the AFT module.

and ablation experiments on the benchmark datasets are conducted to demonstrate the effectiveness of our strategy.

4.1. Experimental Setup

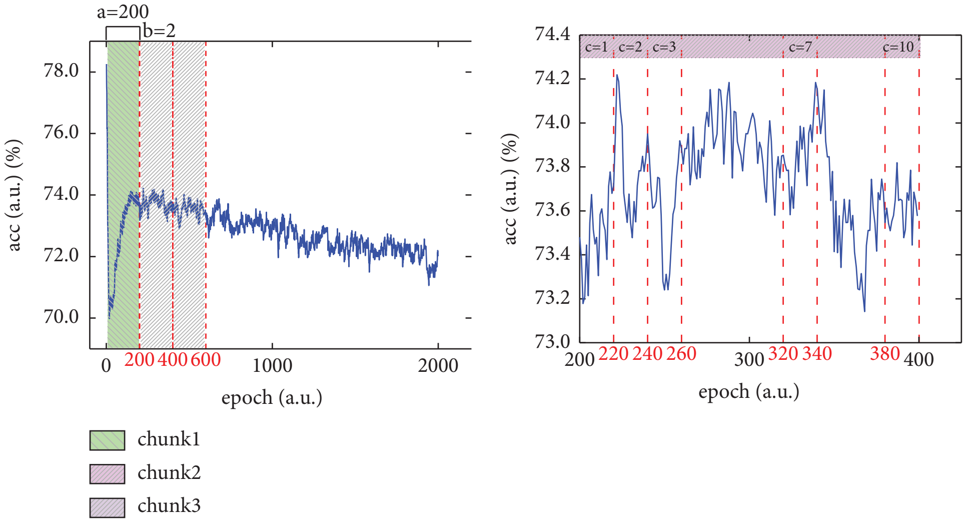
4.1.1. Dataset. We employ mini-ImageNet [22] and tiered-ImageNet [38] datasets. Mini-ImageNet is a subset of ImageNet. It consists of 100 classes, and each class has 600 images with a size of 84 × 84. We follow the setting proposed by [39] to split the datasets into 64, 16, and 20 classes as the training, validation, and testing sets, respectively. Tiered-ImageNet is a larger subset of ImageNet than mini-ImageNet. It has 608 classes, and each class contains 1,281 images on average. In the experiment, 351, 97, and 160 classes are selected as the training, validation, and test set stemming from 20, 6, and 8 superclasses, respectively.

4.1.2. Implementation Details. Following the settings in [10], for the pretraining stage, we first train 100 epochs with batch size 128 on mini-ImageNet, and the learning rate decays at epoch 90. We use SGD optimizer with momentum 0.9, the learning rate 0.1, the decay factor 0.1, and the weight decay 0.0005. For the meta-learning stage, we use SGD optimizer with the weight decay 0.0005 and the learning rate 0.001. For the fine-tuning stage, we set up two kinds of layer frozen policies following [40], namely, fine-tuning all layers ("All," updating all parameters of the model) and fine-tuning the last layer ("Last1," allowing to update only the last fully connected layer of the model). We use the SGD optimizer with momentum 0.9, the weight decay 0.0005, and the learning rate 0.001. We use ResNet-18 as the backbone network and apply standard data augmentation, including random resized crop and random horizontal flip.

For the hyperparameter M, we refer to related work [37] and follow the general meta-learning configurations, setting the total number of pseudofine-tuning tasks $M = 100$. As for the maximum number of epochs $E^{\max}$, we find that the maximum value of the optimal epoch does not exceed 2000. Therefore, we set $E^{\max} = 2000$ to save computing resources. As per Figure 4(a), the accuracy curve has short-term vibration at the beginning and returns to normal before the epoch around 200. So, we set the number of epochs contained in a chunk $a = 200$ and the starting chunk number $b = 2$ to make the adaptive algorithm avoid the influence of short-term vibration during the initial fine-tuning stage. According to Figure 4(b), we see a slight variation in accuracy within a chunk. In order to get the balance between estimation accuracy and calculation efficiency, we set the number of nodes contained in a chunk $c = 10$, only evaluating the model 10 times for each chunk.

4.2. HFT Experiments. Following Algorithm 1, we perform experiments on mini-ImageNet to give the hands-on hybrid fine-tuning strategy under different sample sizes (1, 5, 10, 20, 30) and different layer frozen policies ("Last1," "All").

The main results are shown in Table 1. For the layer frozen policy "Last1," the optimal adaptive epoch is always 0 under different sample sizes, which means the FSLDA has



(a)

(b)

Figure 4: Typical accuracy curve to illustrate the hyperparameter settings for AFT.

Table 1: The hands-on hybrid fine-tuning strategy acquired by the proposed method under different sample sizes and layer frozen policies.

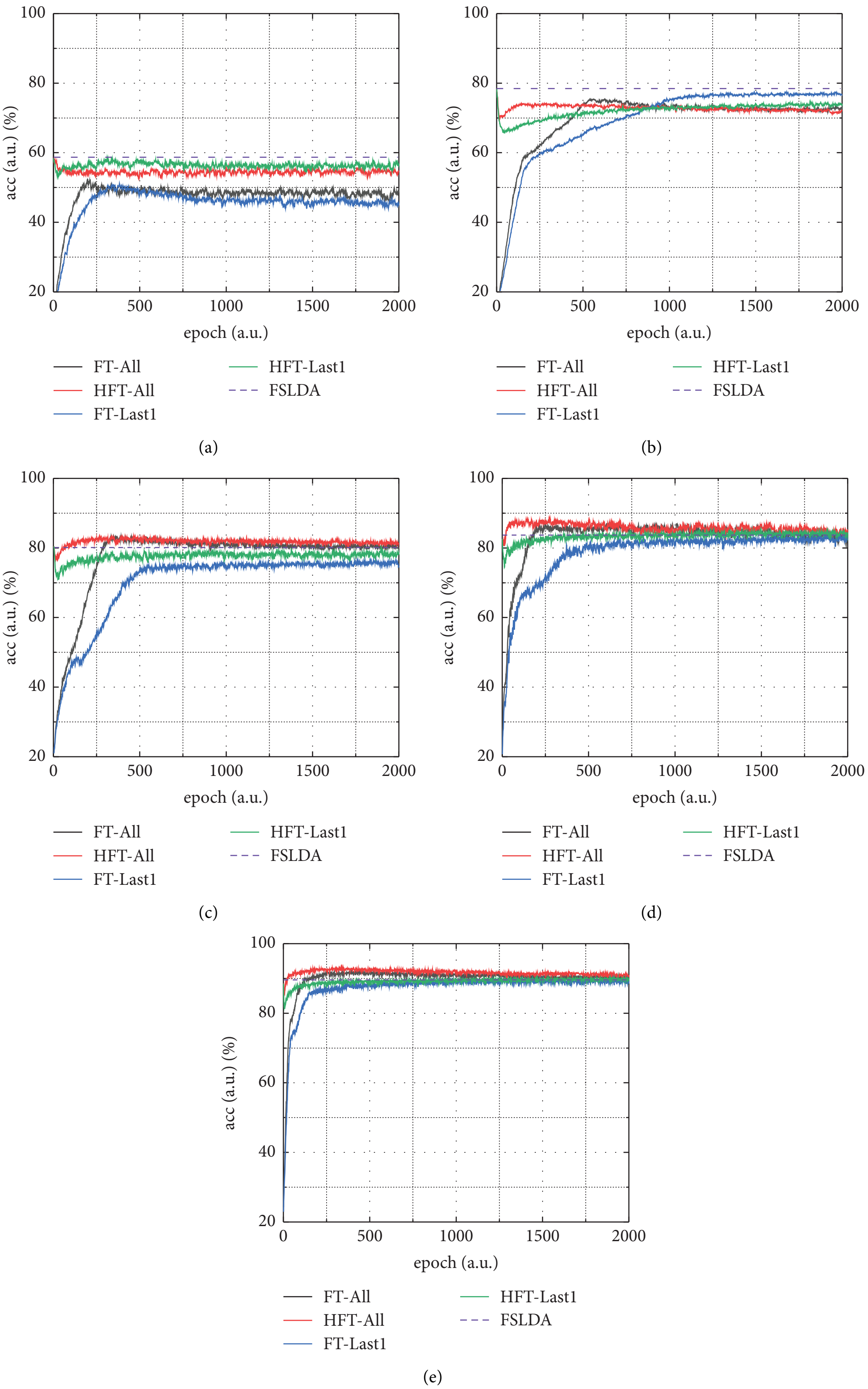
Layer frozen policy	1-Shot	5-Shot	10-Shot	20-Shot	30-Shot
Last1	0	0	0	0	0
All	0	0	1400	1600	1800

initialized the head of the pretrained model so well that only fine-tuning the last layer cannot make the model achieve better performance. Thus, the hands-on hybrid fine-tuning strategy under the layer frozen policy “Last1” is only FSLDA that has constructed the optimal solution for the classifier. In this case, further fine-tuning may lead to suboptimal solutions. In contrast, the hands-on hybrid fine-tuning strategy is inconsistent for the layer frozen policy “All” under different sample sizes. For sample sizes of 1-shot and 5-shot, the hands-on hybrid fine-tuning strategy is also only FSLDA. A common assumption is that too few samples in the support set are not enough to update all the model parameters for better performance. While for sample sizes of 10-shot, 20-shot, and 30-shot, the optimal adaptive epoch is no longer 0. Moreover, as the sample size increases, the optimal adaptive epoch increases, but it is always smaller than the maximum number of epochs. Thus, the hands-on hybrid fine-tuning strategy for sample sizes of 10-shot, 20-shot, and 30-shot contains both FSLDA and AFT. This indicates that adaptive fine-tuning can achieve better performance under the layer frozen policy “All” as the sample size increases.

Furthermore, Figure 5 shows typical convergence curves of testing accuracy during adaptive epoch learning on mini-ImageNet under different layer frozen policies and sample sizes. Here, FT-All and FT-Last1, respectively, refer to updating all parameters of the model and updating only the head, where the head is initialized randomly and the fixed epoch is set by experience. HFT-All and HFT-Last1 refer to performing fine-tuning under the corresponding layer frozen policies “All” and “Last1,” where the head is initialized by FSLDA and the epoch is set according to the acquired hands-on hybrid fine-tuning strategy. FSLDA refers to testing accuracy of the FSLDA model without fine-tuning. Note that we show the full curves for HFT-All and HFT-Last1 in Figure 5 for better comparison. We can see that, for sample sizes of 1-shot and 5-shot, the performance of the FSLDA model (purple dotted horizontal line) is always better than those of other methods, indicating that FSLDA is enough when the sample size is no more than 5. While for sample sizes of 10-shot, 20-shot, and 30-shot, the FSLDA model outperforms FT-Last1 (blue lines) and HFT-Last1 (green lines) but is not as good as FT-All (black lines) and HFT-All (red lines) and the latter one is slightly better. These all indicate the reasonableness of the acquired hands-on hybrid fine-tuning strategy.

4.3. Comparative Experiments. Based on the hands-on hybrid fine-tuning strategy obtained in Section 4.2, we now compare the performance of the hybrid fine-tuning strategy (HFT-Last1/HFT-All) with that of the traditional fine-tuning strategy (FT-Last1/FT-All) under different pretraining methods including RFS-simple [29], SKD-GEN0 [41], and R2D2 [42]. For the sake of fairness, the training epoch for FT-Last1/FT-All is set as $E^{\max}$, i.e., the hyperparameter in Algorithm 1, and other parameter settings are consistent with those of HFT-Last1/HFT-All.

Table 2 shows the comparison results on mini-ImageNet. We can see that the accuracy of HFT-Last1/HFT-All is consistently higher than its corresponding accuracy of FT-Last1/FT-All under all sample sizes, layer frozen policies, and pretraining methods. Compared with FT-Last1/FT-All, HFT-Last1/HFT-All has an average performance improvement of 2.30% on the whole, which proves the effectiveness of combining the advantages of FSLDA and AFT.



(a)

(b)

(c)

(d)

(e)

Figure 5: Typical convergence curves of testing accuracy during adaptive epoch learning on mini-ImageNet for sample sizes of 1-shot (a), 5-shot (b), 10-shot (c), 20-shot (d), and 30-shot (e).

Table 2: Comparison results under different pretraining methods on mini-ImageNet. “Pre-tra” and “Lay-fro” are short for the pretraining method and the layer frozen policy, respectively. We report the mean accuracy of 600 episodes and the 95% confidence intervals.

Pre-tra	Lay-fro	1-Shot	5-Shot	10-Shot	20-Shot	30-Shot	Average gain
R2D2	FT-Last1	50.58 ± 0.74	66.15 ± 0.36	71.07 ± 0.71	75.63 ± 0.87	76.56 ± 0.96	**3.83↑**
	HFT-Last1	**53.47 ± 0.61**	**70.13 ± 0.50**	**74.72 ± 0.45**	**79.67 ± 0.45**	**81.16 ± 0.47**	
	FT-all	51.39 ± 0.81	68.63 ± 0.40	73.38 ± 0.61	79.43 ± 0.66	80.84 ± 0.99	**1.90↑**
	HFT-all	**53.47 ± 0.61**	**70.13 ± 0.50**	**75.49 ± 0.59**	**81.41 ± 0.70**	**82.66 ± 0.38**	
SKD-GEN0	FT-Last1	57.83 ± 0.53	73.91 ± 0.53	78.19 ± 1.03	85.03 ± 0.76	87.01 ± 0.45	**2.36↑**
	HFT-Last1	**60.74 ± 0.68**	**77.45 ± 0.49**	**81.30 ± 0.38**	**86.31 ± 0.42**	**87.96 ± 0.39**	
	FT-all	59.94 ± 0.77	74.34 ± 0.56	81.69 ± 1.09	86.96 ± 0.32	87.43 ± 0.74	**1.19↑**
	HFT-all	**60.74 ± 0.68**	**77.45 ± 0.49**	**82.34 ± 1.01**	**87.15 ± 0.38**	**88.63 ± 0.52**	
RFS-simple	FT-Last1	56.99 ± 0.60	72.43 ± 0.32	76.27 ± 0.29	82.97 ± 1.29	84.02 ± 0.94	**1.38↑**
	HFT-Last1	**58.41 ± 0.71**	**73.66 ± 0.51**	**78.85 ± 0.45**	**83.58 ± 0.56**	**85.10 ± 0.55**	
	FT-all	57.10 ± 0.21	72.79 ± 0.59	79.31 ± 0.28	83.01 ± 0.51	85.23 ± 0.91	**0.86↑**
	HFT-all	**58.41 ± 0.71**	**73.66 ± 0.51**	**79.69 ± 0.75**	**83.81 ± 0.73**	**86.16 ± 0.42**	
Average gain		**2.29↑**	**2.85↑**	**2.50↑**	**1.78↑**	**2.12↑**	**2.30↑**

In addition, the results show that the average performance gains of the layer frozen policy “Last1” are higher than those of the layer frozen policy “All” (3.83% vs. 1.90%, 2.36% vs. 1.19%, and 1.38% vs. 0.86%). Since HFT-Last1 is indeed FSLDA, this phenomenon validates that the linear classifier constructed by FSLDA is much better than that acquired by fine-tuning. Thirdly, for sample size from 1-shot to 30-shot, HFT-Last1/HFT-All achieves an average performance improvement of 1.78% ∼ 2.85% over FT-Last1/FT-All, and the gains are relatively close, indicating that the proposed algorithm has good generalization ability for different sample sizes. Lastly, we can see that the accuracy of the layer frozen policy “All” is always higher or not less than its corresponding accuracy of the layer frozen policy “Last1,” which is consistent with the conclusions of [33, 34].

For tiered-ImageNet dataset, the category correlations between the training set and the test set are weak, and thus, it is more suitable for testing the generalization ability to novel few-shot classification tasks. The comparison results are shown in Table 3. Overall, we can see an average performance improvement of 2.78% for HFT-Last1/HFT-All, surpassing the average gain of 2.30% on mini-ImageNet. This shows that the proposed algorithm has strong generalization ability and can better adapt to novel few-shot classification scenarios. For layer frozen policies “Last1” and “All”, HFT-Last1/HFT-All achieves an average performance improvement of 2.66% ∼ 3.58% and 1.45% ∼ 1.77%, respectively, which is slightly larger than that on mini-ImageNet. For different sample sizes, HFT-Last1/HFT-All achieves an average performance improvement of 2.13% ∼ 3.37%. The average gains in 1-shot and 5-shot are larger than those in 10-shot, 20-shot, and 30-shot, which further illustrates that FSLDA plays an essential role when the sample size is less than 5. As for the comparison of different fine-tuning policies under the same pretraining method and the same finetuning strategy, the policy “All” is always better or not less than the policy “Last1,” which is the same as the conclusion on mini-ImageNet.

4.4. Ablation Experiments. In this section, we analyze the effects of FSLDA and AFT modules in our HFT, respectively. The experiments are carried out on mini-ImageNet under the two layer frozen policies “Last1” and “All,” employing the Meta-Baseline pretraining method [10]. The results are shown in Table 4. For the layer frozen policy “Last1,” HFT is indeed FSLDA; thus, AFT is useless (✓/×) when FSLDA is employed (✓). For the layer frozen policy “All,” the acquired hands-on hybrid fine-tuning strategy is built on FSLDA; thus, AFT cannot be run separately.

We can see that using FSLDA alone can perform consistently better than traditional fine-tuning methods under different sample sizes and layer frozen policies. For the layer frozen policy “Last1,” FSLDA alone achieves 2.26%, 4.35%, 4.03%, 2.82%, and 1.88% gains under the sample sizes of 1-shot, 5-shot, 10-shot, 20-shot, and 30-shot, respectively. Overall, it has an average performance improvement of 3.07%. For the layer frozen policy “All,” FSLDA also achieves gains of 3.34%, 5.47%, 4.13%, 1.54%, and 0.45% under the corresponding sample sizes though FSLDA is only designed for the last layer. Moreover, it obtains an average increase of 2.99% on the whole, which is close to that under the layer frozen policy “Last1.” A common explanation for this is that fine-tuning the classifier of the model using few-shot samples in the support set usually converges to a suboptimal solution, leading to the fine-tuned model’s poor performance. FSLDA gives the classifier an optimal solution by fully excavating the professional knowledge of the novel classes, which means the FSLDA model outperforms the model with the experience-based fine-tuning method, even without fine-tuning. For the layer frozen policy “All,” AFT brings 0.40%, 0.99%, and 0.79% performance improvements over individual FSLDA under the sample sizes of 10-shot, 20-shot, and 30-shot, respectively, and the average gain reaches 0.72%. This is because the adaptive epoch obtained by AFT can predictably help the FSLDA model update parameters through backpropagation while preventing the model from underfitting and overfitting, which enables the model to achieve better performance than the FSLDA model alone. One interesting thing is that the accuracies of the policy “All” under sample sizes of 1-shot, 5-shot, and 10-shot are lower than those of the policy “Last1” for the traditional fine-tuning method, which is not consistent with the conclusions of [33, 34] and brings uncertainty to the choice of the layer frozen policy.

Table 3: Comparison results under different pretraining methods on tiered-ImageNet. "Pre-tra" and "Lay-fro" are short for the pretraining method and the layer frozen policy, respectively. We report the mean accuracy of 600 episodes and the 95% confidence intervals.

Pre-tra	Lay-fro	1-Shot	5-Shot	10-Shot	20-Shot	30-Shot	Average gain
R2D2	FT-Last1	52.10 ± 0.70	68.99 ± 0.70	73.21 ± 0.30	76.82 ± 0.89	80.38 ± 1.24	**2.66↑**
	HFT-Last1	**55.18 ± 0.72**	**72.26 ± 0.66**	**75.19 ± 0.62**	**80.35 ± 0.63**	**81.82 ± 0.62**	
	FT-all	52.90 ± 0.78	70.87 ± 0.69	75.02 ± 0.28	80.04 ± 0.72	84.69 ± 0.96	**1.45↑**
	HFT-all	**55.18 ± 0.72**	**72.26 ± 0.66**	**76.57 ± 0.24**	**81.50 ± 0.91**	**85.24 ± 0.22**	
SKD-GEN0	FT-Last1	60.51 ± 0.75	76.28 ± 0.80	80.54 ± 0.71	83.84 ± 0.67	86.10 ± 0.61	**3.58↑**
	HFT-Last1	**64.17 ± 0.82**	**79.42 ± 0.61**	**83.75 ± 0.53**	**87.60 ± 0.42**	**90.25 ± 0.34**	
	FT-all	61.05 ± 0.76	76.37 ± 0.79	83.46 ± 0.50	87.01 ± 1.35	90.45 ± 1.26	**1.58↑**
	HFT-all	**64.17 ± 0.82**	**79.42 ± 0.61**	**83.79 ± 0.89**	**87.76 ± 0.98**	**91.09 ± 0.93**	
RFS-simple	FT-Last1	60.45 ± 0.98	74.09 ± 0.79	78.86 ± 0.58	83.36 ± 0.61	83.51 ± 1.52	**2.86↑**
	HFT-Last1	**63.76 ± 0.88**	**77.74 ± 0.57**	**81.27 ± 0.53**	**85.35 ± 0.50**	**86.45 ± 0.57**	
	FT-all	60.56 ± 0.97	75.39 ± 0.80	80.18 ± 0.43	83.90 ± 0.47	88.04 ± 1.35	**1.77↑**
	HFT-all	**63.76 ± 0.88**	**77.74 ± 0.57**	**81.42 ± 0.81**	**84.98 ± 0.65**	**89.01 ± 0.81**	
Average gain		**3.37↑**	**3.37↑**	**2.41↑**	**2.51↑**	**2.13↑**	**2.78↑**

Table 4: Ablation experiments on mini-ImageNet employing the meta-baseline pretraining method. We report the mean accuracy of 600 episodes and the 95% confidence intervals.

	FSLDA	AFT	1-Shot	5-Shot	10-Shot	20-Shot	30-Shot
Last1	×	×	48.14 ± 0.95	69.32 ± 0.75	74.02 ± 0.81	81.66 ± 0.40	86.39 ± 0.91
	✓	✓/×	**50.40 ± 0.35**	**73.67 ± 0.64**	**78.05 ± 0.31**	**84.48 ± 0.89**	**88.27 ± 0.77**
All	×	×	47.06 ± 0.96	68.20 ± 0.75	73.92 ± 0.73	82.94 ± 0.11	87.82 ± 0.94
	✓	×	**50.40 ± 0.35**	**73.67 ± 0.64**	78.05 ± 0.31	84.48 ± 0.89	88.27 ± 0.77
	✓	✓	—	—	**78.45 ± 0.92**	**85.47 ± 1.05**	**89.06 ± 0.88**

5. Conclusion

In this study, we have introduced a hybrid fine-tuning strategy (HFT) for FSC, including the FSLDA and AFT modules. FSLDA constructs the optimal linear classifier, and AFT outputs the hybrid fine-tuning strategy based on the FSLDA model. HFT solves the problem that the linear classifier is suboptimal under few-shot conditions and prevents the model from overfitting and underfitting by using the acquired hands-on hybrid finetuning strategy. By conducting extensive experiments, we find HFT achieves consistent performance improvements compared to traditional finetuning methods under different sample sizes, layer frozen policies, and few-shot classification frameworks. Intuitively, our HFT has enormous potential for FSC and even for few-shot learning. In the future, we will try to explore automatic learning methods of more hyperparameters for the fine-tuning stage.

Appendix

LDA classifier: LDA is a classical optimal linear classifier using Bayes' theorem. For a C-way K-shot classification task, let X and Y be the random variables for data samples and labels, respectively. The posterior probability of an observation x that belongs to the c^{th} class can be written as

$$P(Y = c|X = x) = \frac{\pi_c f_c(x)}{\sum_{i=1}^{C} \pi_c f_i(x)}, \tag{A.1}$$

where π_c is the prior probability which can be easily calculated by simply computing the fraction of the training observations that belong to c^{th} class, $f_c(x)$ is the conditional probability that an observation x belongs to c^{th} class, and $\sum_{i=1}^{C} \pi_c f_i(x)$ is a normalization constant.

To simplify the problem, LDA assumes that $f_c(x)$ obeys multivariate Gaussian distribution and the covariance matrix Σ of all classes is the same:

$$f_c(x) = \frac{1}{2\pi^{p/2}|\Sigma|^{0.5}} e^{-1/2\,(x-\mu_c)^T \Sigma^{-1} (x-\mu_c)}, \tag{A.2}$$

$$\Sigma = \frac{1}{C(K-1)} \sum_{c=1}^{C} \sum_{i=1}^{K} \left(x_c^i - \mu_c\right)\left(x_c^i - \mu_c\right)^T, \tag{A.3}$$

$$\mu_c = \frac{1}{K} \sum_{i=1}^{K} x_c^i. \tag{A.4}$$

Thus, the posterior probability can be written as

$$P(Y = c|X = x) = A \cdot \pi_c e^{-1/2\,(x-\mu_c)^T \Sigma^{-1} (x-\mu_c)}, \tag{A.5}$$

where $A = 1/\sum_{i=1}^{C} \pi_c f_i(x) \cdot 1/2^{p/2}|\Sigma|^{0.5}$ is a constant.

Then, LDA takes the logarithm of the posterior probability (ignores the constant item):

$$\begin{aligned}
P(Y = c|X = x) \\
&= \log\pi_c - \frac{1}{2}\big(x^T\Sigma^{-1}x - x^T\Sigma^{-1}\mu_c \\
&\quad -\mu_c^T\Sigma^{-1}x + \mu_c^T\Sigma^{-1}\mu_c\big), \\
&= \log\pi_c - \frac{1}{2}\big(x^T\Sigma^{-1}x - 2\mu_c^T\Sigma^{-1}x + \mu_c^T\Sigma^{-1}\mu_c\big),
\end{aligned} \tag{A.6}$$

where $x^T\Sigma^{-1}x$ is independent of the category of x. Therefore, the linear score function can be represented as

$$P(Y = c|X = x) = \log\pi_c + \mu_c^T\Sigma^{-1}x - \frac{1}{2}\mu_c^T\Sigma^{-1}\mu_c. \tag{A.7}$$

For a C-way K-shot classification task, π_c is also an irrelevant item and the final linear classifier function becomes

$$P(Y = c|X = x) = \mu_c^T\Sigma^{-1}x - \frac{1}{2}\mu_c^T\Sigma^{-1}\mu_c. \tag{A.8}$$

Equations (A.3), (A.4), and (A.8) form the LDA classifier as used in Section 3.2.

Data Availability

The data used to support the findings of this study can be obtained from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partly supported by the National Natural Science Foundation of China (Grant nos. U19B2033 and 62076020).

References

[1] S. Bai, S. Song, S. Liang, J. Wang, B. Li, and E. Neretin, "Uav maneuvering decision-making algorithm based on twin delayed deep deterministic policy gradient algorithm," *Journal of Artificial Intelligence and Technology*, vol. 2, no. 1, pp. 16–22, 2022.

[2] M. Saqlain, S. Rubab, M. M. Khan, N. Ali, and S. Ali, "Hybrid approach for shelf monitoring and planogram compliance (hyb-smpc) in retails using deep learning and computer vision," *Mathematical Problems in Engineering*, vol. 2022, Article ID 4916818, 18 pages, 2022.

[3] Y. Yang and X. Song, "Research on face intelligent perception technology integrating deep learning under different illumination intensities," *Journal of Computational and Cognitive Engineering*, vol. 1, pp. 32–36, 2022.

[4] Z. Lu, S. He, X. Zhu, L. Zhang, Y.-Z. Song, and T. Xiang, "Simpler is better: few-shot semantic segmentation with classifier weight transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8741–8750, Montreal, Canada, October 2021.

[5] L. Yang, W. Zhuo, L. Qi, Y. Shi, and Y. Gao, "Mining latent classes for few-shot segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8721–8730, Montreal, Canada, October 2021.

[6] D. Hendrycks, K. Lee, and M. Mazeika, "Using pre-training can improve model robustness and uncertainty," in *Proceedings of the International Conference on Machine Learning*, pp. 2712–2721, Da Lat, Vietnam, January 2019.

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, Columbus, OH, USA, June 2014.

[8] P. Agrawal, R. Girshick, and J. Malik, "Analyzing the performance of multilayer neural networks for object recognition," in *Proceedings of the European Conference on Computer Vision*, pp. 329–344, Springer, Berlin, Germany, September 2014.

[9] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the Devil in the Details: Delving Deep into Convolutional Nets," 2014, https://arxiv.org/abs/1405.3531.

[10] Y. Chen, X. Wang, Z. Liu, H. Xu, and T. Darrell, "A New Meta-Baseline for Few-Shot Learning," 2020, https://arxiv.org/abs/2003.04390.

[11] P. Ma, Z. Zhang, J. Wang et al., "Review on the application of metalearning in artificial intelligence," *Computational Intelligence and Neuroscience*, vol. 2021, Article ID 1560972, 12 pages, 2021.

[12] P. Bateni, R. Goyal, V. Masrani, F. Wood, and L. Sigal, "Improved few-shot visual classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA, June 2020.

[13] D. Chen, Y. Chen, Y. Li, F. Mao, Y. He, and H. Xue, "Self-supervised learning for few-shot image classification," in *Proceedings of the ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1745–1749, IEEE, Toronto, ON, Canada, June 2021.

[14] J. Hong, P. Fang, W. Li et al., "Reinforced attention for few-shot learning and beyond," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 913–923, Nashville, TN, USA, June 2021.

[15] A. Li, T. Luo, T. Xiang, W. Huang, and L. Wang, "Few-shot learning with global class representations," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9715–9724, Seoul, Korea (South), October 2019.

[16] H.-J. Ye, H. Hu, D.-C. Zhan, and F. Sha, "Few-shot learning via embedding adaptation with set-to-set functions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8808–8817, Seattle, WA, USA, June 2020.

[17] J. Zhang, C. Zhao, B. Ni, M. Xu, and X. Yang, "Variational few-shot learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1685–1694, Seoul, Korea (South), October 2019.

[18] D. Zhang and T. Yang, "Visual object tracking algorithm based on biological visual information features and few-shot learning," *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 3422859, 8 pages, 2022.

[19] Z.-M. Wang, J.-Y. Tian, J. Qin, H. Fang, and L.-M. Chen, "A few-shot learning-based siamese capsule network for intrusion detection with imbalanced training data," *Computational Intelligence and Neuroscience*, vol. 2021, Article ID 7126913, 17 pages, 2021.

[20] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of the International Conference on Machine Learning*, pp. 1126–1135, Sydney, Australia, August 2017.

[21] C. Finn, K. Xu, and S. Levine, "Probabilistic model-agnostic meta-learning," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[22] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, and K. Kavukcuoglu, "Matching networks for one shot learning," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[23] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[24] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: relation network for few-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1199–1208, Salt Lake City, UT, USA, June 2018.

[25] B. Hariharan and R. Girshick, "Low-shot visual recognition by shrinking and hallucinating features," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3018–3027, Venice, Italy, October 2017.

[26] P. Mangla, N. Kumari, A. Sinha, M. Singh, B. Krishnamurthy, and V. N. Balasubramanian, "Charting the right manifold: manifold mixup for few-shot learning," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2218–2227, Snowmass, CO, USA, March 2020.

[27] C. Liu, Y. Fu, C. Xu et al., "Learning a few-shot embedding model with contrastive learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 10, pp. 8635–8643, New York, NY, USA, February 2021.

[28] C. Xu, Y. Fu, C. Liu et al., "Learning dynamic alignment via meta-filter for few-shot learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5182–5191, Nashville, TN, USA, June 2021.

[29] Y. Tian, Y. Wang, D. Krishnan, J. B. Tenenbaum, and P. Isola, "Rethinking few-shot image classification: a good embedding is all you need?" in *Proceedings of the European Conference on Computer Vision*, pp. 266–282, Springer, Glasgow, UK, August 2020.

[30] N. Lai, M. Kan, C. Han, X. Song, and S. Shan, "Learning to learn adaptive classifier–predictor for few-shot learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 8, pp. 3458–3470, 2021.

[31] J. Liu, L. Song, and Y. Qin, "Prototype rectification for few-shot learning," in *Proceedings of the European Conference on Computer Vision*, pp. 741–756, Springer, Glasgow, UK, August 2020.

[32] J. P. Miller, R. Taori, A. Raghunathan et al., "Accuracy on the line: on the strong correlation between out-of-distribution and in-distribution generalization," in *Proceedings of the International Conference on Machine Learning*, pp. 7721–7735, PMLR, Ghaziabad, India, December 2021.

[33] S. Kornblith, J. Shlens, and Q. V. Le, "Do better imagenet models transfer better?" in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2661–2671, Long Beach, CA, USA, June 2019.

[34] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, Seattle, WA, USA, June 2020.

[35] X. Wang, T. Huang, J. Gonzalez, T. Darrell, and F. Yu, "Frustratingly simple few-shot object detection," in *Proceedings of the International Conference on Machine Learning*, pp. 9919–9928, Vienna, Austria, July 2020.

[36] C. Zhu, F. Chen, U. Ahmed, Z. Shen, and M. Savvides, "Semantic relation reasoning for shot-stable few-shot object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8782–8791, Nashville, TN, USA, June 2021.

[37] Z. Shen, Z. Liu, J. Qin, M. Savvides, and K.-T. Cheng, "Partial is better than all: revisiting fine-tuning strategy for few-shot learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 11, pp. 9594–9602, 2021.

[38] M. Ren, E. Triantafillou, S. Ravi et al., "Meta-learning for semi-supervised few-shot classification," *Training*, vol. 1, no. 2, p. 3, 2018.

[39] S. Ravi and H. Larochelle, *Optimization as a Model for Few-Shot Learning*, International Conference on Learning Representations, Toulon, France, 2017.

[40] Y. Guo, N. C. Codella, L. Karlinsky et al., "A broader study of cross-domain few-shot learning," in *Proceedings of the European Conference on Computer Vision*, pp. 124–141, Springer, Glasgow, UK, August 2020.

[41] J. Rajasegaran, S. Khan, M. Hayat, F. S. Khan, and M. Shah, "Self-supervised Knowledge Distillation for Few-Shot Learning," 2020, https://arxiv.org/abs/2006.09785.

[42] L. Bertinetto, J. F. Henriques, P. Torr, and A. Vedaldi, "Meta-learning with differentiable closed-form solvers," in *Proceedings of the International Conference on Learning Representations*, Vancover, Canada, May 2018.