This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

MetaFSCIL: A Meta-Learning Approach for Few-Shot Class Incremental Learning

Zhixiang Chi[1], Li Gu[1], Huan Liu[1,2], Yang Wang[1,3], Yuanhao Yu[1], Jin Tang[1]
[1]Noah's Ark Lab, Huawei Technologies [2]McMaster University, Canada
[3]University of Manitoba, Canada
{zhixiang.chi, li.gu, huan.liu3, yang.wang3, yuanhao.yu, tangjin}@huawei.com

Abstract

In this paper, we tackle the problem of few-shot class incremental learning (FSCIL). FSCIL aims to incrementally learn new classes with only a few samples in each class. Most existing methods only consider the incremental steps at test time. The learning objective of these methods is often hand-engineered and is not directly tied to the objective (i.e. incrementally learning new classes) during testing. Those methods are sub-optimal due to the misalignment between the training objectives and what the methods are expected to do during evaluation. In this work, we proposed a bi-level optimization based on meta-learning to directly optimize the network to learn how to incrementally learn in the setting of FSCIL. Concretely, we propose to sample sequences of incremental tasks from base classes for training to simulate the evaluation protocol. For each task, the model is learned using a meta-objective such that it is capable to perform fast adaptation without forgetting. Furthermore, we propose a bi-directional guided modulation, which is learned to automatically modulate the activations to reduce catastrophic forgetting. Extensive experimental results demonstrate that the proposed method outperforms the baseline and achieves the state-of-the-art results on CIFAR100, MiniImageNet, and CUB200 datasets.

1. Introduction

With the unprecedented increase in computational budget and data availability, deep models have achieved superior performance in the recognition tasks [10, 26]. Typically, those methods are offline trained on some pre-defined image categories and then deployed to target applications with fixed parameters. Such systems are not flexible enough since they cannot handle new classes that might emerge after deployment. In contrast, humans are able to learn new concepts incrementally throughout their lifetime.

Recently, class incremental learning has been an active area of research [2, 22, 25]. However, large-scale annotated data for the new classes are required, which leads to the well-known notorious catastrophic forgetting [19]. The forgetting issue becomes more severe when all the data for old classes are unavailable. Moreover, it is unrealistic for the end-users to collect and annotate numerous data. Thus, in this paper, we focus on a more practical and challenging setting: *few-shot class incremental learning* (FSCIL) [27].

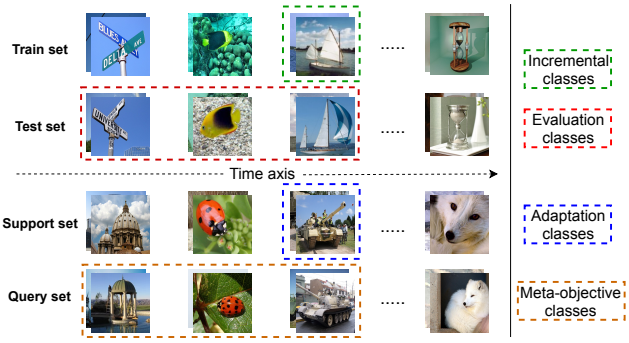

Figure 1. **Illustration of the evaluation protocol and our meta-training process.** During the evaluation of FSCIL, at each incremental session, the model is trained only on new classes but is evaluated on all classes encountered so far. Our MetaFSCIL follows the same rule where the model adapted to new classes is validated using a meta-objective based on all encountered classes.

FSCIL consists of an offline training stage and an online incremental learning stage. In the offline training stage, we have access to a large-scale dataset for some *base* classes. FSCIL learns a model on these *base* classes. During the online incremental learning (i.e. evaluation) stage, we will encounter *novel* classes in a sequential manner, where a few novel classes are presented at each time step (called *incremental session*). For each novel class, we only have a few training examples. In addition, we can only access training examples corresponding to the novel classes at the current time step. In other words, we cannot store training examples from previous time steps (e.g. due to limited storage

of deployment environment) during evaluation. The evaluation protocol is defined such that at each incremental session, after learning the novel classes, the model is evaluated on all encountered classes (including bases classes), as shown in Fig. 1. FSCIL is challenging due to two main reasons, namely catastrophic forgetting of old classes and adaptation ability of new classes. To mitigate the forgetting issue, some works [3, 7] use knowledge distillation-based methods. These methods typically require extra space to store exemplars from previous sessions. This is not realistic for memory-constrained devices when the incremental steps increase. Besides, the incremental learning process is only involved in the "evaluation phase" in [3, 7]. Thus, the model parameters are not directly optimized to handle forgetting and adaptation. In other words, these methods are sub-optimal due to the *misalignment* between their learning objective and the evaluation protocol.

Some recent works simulate the incremental process using the available *base* classes to better match the evaluation protocol. A random task selection strategy is proposed to enhance the extensibility of representation for the *novel* classes [33]. However, during training, only one of the *base* classes is selected for sampling query images. The work in [32] randomly samples an incremental task to meta-train a classifier refiner such that it can incorporate the classifiers for old and new classes when learning new classes at deployment. During training, two non-overlapping and equal-sized subsets are sampled from the *base* classes as *pseudo base* and incremental classes. However, the imbalance between the many-shot (*base*) and the few-shot (*novel*) classes is not considered. Furthermore, for both methods [32, 33], only one incremental session is considered. As a result, the model is not learned to incrementally learn in a longer horizon. The mismatch between the their sampling strategies for training and the incremental scenario at online evaluation leads to non-optimal solutions.

In addition, the aforesaid approaches involve hand-engineered heuristics (e.g. saving exemplars [3, 7], decoupled learning [32], prototype refinement [33]) which are defective compared to the learned solutions [6]. For example, the backbones in [32, 33] are manually-designed to be fixed during incremental learning at deployment. The adaptation and generalization to *novel* classes is greatly restricted under distribution shift, as the backbone is not task-agnostic and is biased to seen (*base*) classes [8].

In this work, we propose a fully learned solution based on meta-learning (e.g. MAML [9]) to directly formulate forgetting alleviation and adaptation as the optimization objective. We allow the model to learn how to incrementally learn through a nested optimization-based incremental learning. Concretely, different from [32], we propose a longer sequential task sampling scheme to mimic the increase in catastrophic forgetting as time goes during evaluation, as shown in Fig. 1 (learning addition to a 100-way classifier causes more forgetting than a 10-way classifier). For each task, the model first performs quick adaptation to the new classes via a few gradient updates. Then the meta-objective is defined by validating the adapted model on the query images of previous encountered classes (test forgetting) and current classes (test adaptation). Our meta-objective follows the evaluation protocols at test time. The goal is to learn a model initialization such that it can adapt fast to new classes sequentially and is less prone to catastrophic forgetting.

To further facilitate the optimization process, we build upon the *selective activation mechanism* [1] and propose a **B**i-directional **G**uided **M**odulation (BGM). BGM is meta-trained to automatically gate the activations of the classification module conditioned on the current state (e.g. weights, learned knowledge) of the classification module and incoming images for new classes. The gated activations during the forward pass indirectly affect the back-propagation when learning new classes. BGM is learned to accommodate the parameter update process such that adapting to new knowledge causes less forgetting of old knowledge. Code will be available upon approval. The contributions of this paper are manifold:

- We propose a sequential task sampling scheme to mimic the incremental learning process at evaluation.
- We propose a bi-directional guided modulation to strengthen the back-propagation such that the model can better preserve old knowledge while adapting to the new classes.
- A bi-level meta-learning-based optimization is proposed to directly optimize the model towards forgetting alleviation and adaptation. Our method is fully learned with minimal manually designed components.
- Extensive experiments on standard benchmarks CIFAR100, MiniImageNet, and CUB200 demonstrate that our method outperforms the baselines and achieves state-of-the-art.

2. Related Work

In this section, we review several lines of prior research related to our work.

Meta-learning. Meta-learning is an active area of research. Existing meta-learning approaches include model-based [24], metric-based [13, 28] and optimization-based [9, 21]. The proposed method builds upon one of the most popular meta-learning algorithms, namely model agnostic meta-learning (MAML) [9]. MAML learns the model using a nested optimization, where the inner loop performs task-level optimization, while the outer loop performs a global model update via meta-objective. The goal of MAML is to learn a model initialization such that it can quickly adapt

to any new tasks. MAML has also been used in multi-task settings [5, 17]. In MAXL [17], an auxiliary task is trained alongside the primary task to improve the generalization. To reduce the cost of manual labeling for the auxiliary task, another network is used to generate the auxiliary labels. MAXL uses meta-learning to force the network to automatically discover optimal auxiliary labels. In this paper, we propose to use the bi-level optimization of MAML to allow the model to automatically learn the optimal trade-off between two competing factors, namely adapting new knowledge and remembering old knowledge. The idea is to directly formulate them as the meta-objective, and force the optimization towards solving them both.

Few-shot class incremental learning. FSCIL is recently proposed by [27] to continually learn a sequence of few-shot tasks. Tao *et al.* [27] use a neural gas network to preserve the topology of features for different classes. Dong *et al.* [7] construct an exemplar relation graph to represent the learned knowledge. Zhu *et al.* [33] propose a dynamic relation projection module to constrain the update of new prototypes. Zhang *et al.* [32] propose a continually evolved classifier based on graph attention network to incorporate the global context information among previous tasks. A pseudo incremental training is also proposed to optimize the graph module. However, the optimization objectives of the existing methods are not aligned with the evaluation protocol. In addition, these approaches are heavily hand engineered. The hand-engineered modules limit the performance compared with fully learned solutions.

FSCIL is closely related to *online class incremental learning* setting (OC-IL) [1, 11]. Both settings have an offline stage where we can learn a model from base classes. Both of them have an online learning stage with a sequence of sessions, where each session is a few-shot problem for some new classes. But there are key differences. OC-IL focuses on the streaming of data at the instance level. In contrast, our FSCIL setting incrementally learns new classes via separate tasks. FSCIL also has the data imbalance challenge, where the *base* classes sometimes dominate the learning process. In addition, in OC-IL, during the online learning stage, the evaluation is performed only at the end of all sessions. And the model is not evaluated on the base classes, so the model will not be penalized for catastrophic forgetting of the base classes. But in our FSCIL setting, our model is evaluated at each session during the online learning stage, and the evaluation involves the base classes as well. Obviously, FSCIL is a more challenging problem since we need to deal with both catastrophic forgetting and incremental adaptation to new classes with few examples.

MAML has also been adopted to OC-IL [1, 11]. To simulate the testing scenario, the meta-update in OC-IL is performed at the end of each sampled sequence during training. It makes the learning process unstable when the sequence length increases. In addition, OML [11] and ANML [1] only conducted experiments on simple datasets (e.g. Omniglot [15]). When the variation between classes increase (e.g. MiniImageNet), the performance drops significantly. For example, as shown in [11], the accuracy is halved when 20 *novel* classes are learned.

Network modulation. To avoid catastrophic forgetting, a promising solution is to modulate the plasticity of the learned weights [18, 30]. The modulation mechanism selects and constrains the important weights for previous tasks when learning new tasks [12]. Kirkpatrick *et al.* [12] propose a regularization to determine the weight importance base on the Fisher information. Similarly, Zenke *et al.* [31] use synaptic states to estimate the importance of weights. Learning new tasks is then regularized based on the states. However, most of the methods involve manually designed modules. To avoid that, ANML [1] proposes a *selective activation* mechanism by training another parallel modulation network to modulate the last activation map. It is trained using meta-learning to automatically discover the optimal modulation given the new tasks. However, we have found that the modulation capability performs poorly for deeper networks. In addition, the lack of interaction between modulation and prediction network restricts the performance. In our work, we propose a more effective modulation mechanism that works well for deep networks.

3. Proposed Method

3.1. Problem Setting

FSCIL aims to incrementally learn through a sequence of disjoint classes with a few samples in each class [27]. Specifically, we define a sequence of labeled training datasets $\{\mathcal{D}^0, \mathcal{D}^1, \cdots, \mathcal{D}^T\}$ and their corresponding label set $\mathcal{C}^t$ at session t $(t = 0, 1, ..., T)$. It should be noted that the label sets are disjoint among different sessions, such that $\mathcal{C}^i \cap \mathcal{C}^j = \varnothing (i \neq j)$. Only the classes in the first session $\mathcal{D}^0$ contains large-scale training data. We refer $\mathcal{C}^0$ as *base* classes. We can perform an offline training stage using the *base* classes to learn a model. Once the offline training stage is done, we need to perform online incremental learning to adapt the model to handle new classes in each subsequent incremental session. Each subsequent incremental session $\mathcal{D}^t$ $(t > 0)$ only contains a few training samples for new classes in $\mathcal{C}^t$. We refer them as *novel* classes. For example, in the case of 5-way 5-shot FSCIL, each incremental session $\mathcal{D}^t$ contains 5 new classes where each class has 5 training examples. At the t^{th} incremental session, we only have access to $\mathcal{D}^t$. After learning on $\mathcal{D}^t$, the model is evaluated on test images of all encountered classes so far, i.e. $\mathcal{C}^0 \cup \mathcal{C}^1 \cdots \cup \mathcal{C}^t$.

FSCIL is a realistic setting for many real-world applica-

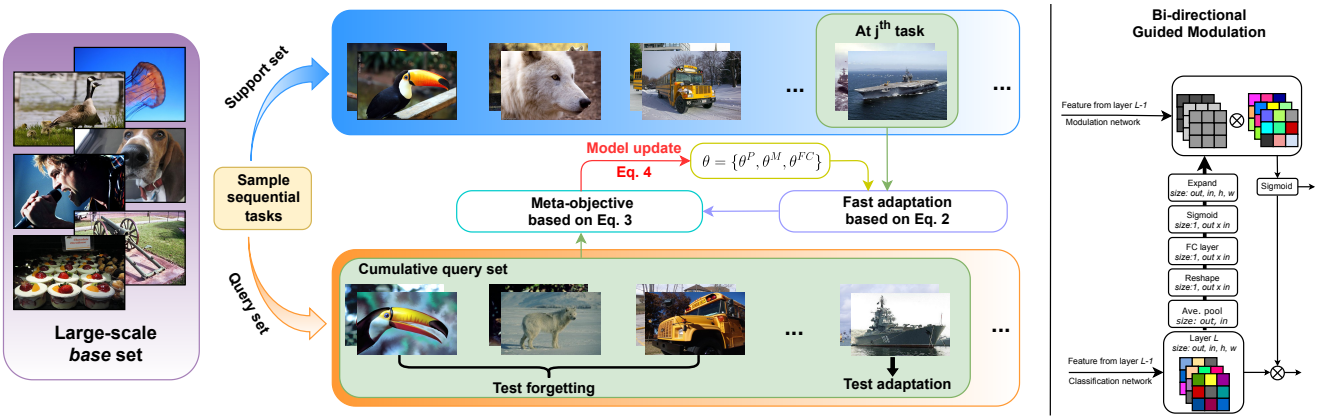

Figure 2. **Overview of our approach.** (Left) Illustration of Alg. 1. We first sample a sequence of sessions to mimic the evaluation protocol. At j^{th} session, the model adapts to new classes via a few gradient steps. Then, we evaluate the adapted model using a meta-objective based on a cumulative query set. The goal is to directly optimize the model such that it learns new classes with less forgetting. (Right) Illustration of the BGM module. We first process the weights of F^C to yield a weight attention map which is applied to the weights of F^M. The weights of F^C reveal its current learning capabilities. F^M then accepts the new images to generate an activation mask to assist reducing the catastrophic forgetting.

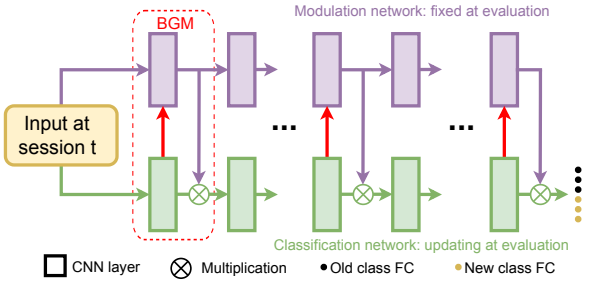

Figure 3. **Illustration of the proposed architecture.** After training, F^M is frozen while F^C keeps updating to learn new classes.

tions. Let us consider an image classification application where a company trains a model on the cloud. It is reasonable to assume that we will have a large dataset for some classes (equivalent to *base* classes in FSCIL) on the cloud. Once the model is trained and deployed to different users, each user may want to incrementally add new object classes to recognize over time. Due to the cost of acquiring training examples, the client will likely only have few-shot examples for those new classes. This is equivalent to incremental sessions with few-shot examples in FSCIL. Since the client device often has limited memory and computing power, it is not realistic to store training examples from previous sessions. FSCIL can be used to solve this practical scenario.

3.2. Bi-directional Guided Modulation

Training a deep model on a specific task normally yields non-uniform importance of learned parameters for that task [16]. In order for the deep model to accommodate new knowledge, a promising solution is to reduce the plasticity of vital parameters for previous tasks and allow the deemed unimportant parameters to learn the new knowledge [12]. The work in [1] follows this idea and adopts a modulation network (denoted as F^M) to produce a mask to modulate the last activation map of the classification module (denoted as F^C) in the online setting. However, there are two main drawbacks that hinder directly adopting [1] in FSCIL. First, only the last activation map is modulated in [1], which scales poorly to deeper networks, e.g. ResNet [10]. We have found that it is more effective to also modulate early layers in a deep network. Second, F^M and F^C are decoupled in [1], such that the F^M is only conditioned on the new data while discarding the learning capacity of F^C at the current incremental session.

To address the aforementioned drawbacks, we have empirically found that gating more activations throughout F^C is beneficial. On the other hand, F^C has different learning capabilities at each incremental session depending on the amount of knowledge it has learned so far. Thus, we propose a bi-directional guided modulation (BGM) to guide F^M using the learned knowledge encoded in the weights of F^C [12]. The right part of Fig. 2 illustrates the structure of BGM module and Fig. 3 illustrates the overall proposed architecture. Let us consider the weights of a certain layer in F^C with shape (out, in, h, w) where out, in, h, w denote the input/output channel number and height/width of each kernel. We follow the Built-in Attention [20] to generate an attention map for each of the $out \times in$ weight kernels as:

$$Z_{i,j} = Sigmoid(FC(Ave(W_{i,j})), \quad (1)$$

where $Z_{i,j}$ and $W_{i,j}$ are the attention map and weight for i^{th} output and j^{th} input channel. FC is the fully connected

layer, and Ave denotes the average pooling operation to average the weights along the (h, w) dimension to one value. The attention map Z is then applied to the weights of F^M via multiplication. Finally, F^M accepts the new input at the current incremental session to generate gating masks. In order to match the sizes of weights, we use the same network architecture for both F^M and F^C.

The attention map Z generated from the weights of F^C reveals the importance of the weights at the current incremental session. It guides F^M to generate masks along with the images for new classes. The gating masks are then applied back to the activations of F^C to modulate the learning of new knowledge with less forgetting.

3.3. Learning to Incrementally Learn

Our approach is inspired by MAML [9] for few-shot learning (FSL). During the meta-training stage, MAML learns from a set of tasks. Each task is constructed as a FSL problem to mimic the scenario during meta-testing.

In FSCIL, the online incremental stage is analogous to the "meta-testing" stage in MAML. The online incremental stage involves adapting the model to a sequence of incremental sessions, where each session involves several novel classes with few-shot examples. This suggests that we should try to mimic this scenario during the offline training stage as well. During the offline training stage, we use a meta-learning approach to learn a model from the $base$ classes. The high-level idea of our method is to use $base$ classes to mimic the incremental learning scenario that we will encounter during the online incremental learning (i.e. evaluation), so that the model is learned in a way that it can effectively adapt to new classes during evaluation.

Sequential task sampling. Having F^M and F^C coupled via BGM, we need to train it to adapt new concepts and reduce forgetting. We mimic the evaluation process using the $base$ classes. Specifically, we separate the training images of $base$ classes as support and query pools without overlapping. At each epoch, we first sample a sequence of $N+1$ tasks (each session contains one task), $\mathcal{D}^s = \{(\mathcal{S}^j, \mathcal{Q}^j)\}_{j=0}^N$, where $\mathcal{S}^j$, $\mathcal{Q}^j$ are the support and query sets for the j^{th} task. Unlike [32,33], where only one incremental task is sampled (e.g. $N = 1$), we allow $N \gg 1$ to simulate the increase in catastrophic forgetting at evaluation. We also set $(\mathcal{S}^0, \mathcal{Q}^0)$ as the *pseudo base* set with more classes and training examples. The subsequent tasks (e.g. $j > 0$) follow the few-shot setting as evaluation. To prevent the model from over-fitting to a certain sequence, we randomly sample the classes and corresponding images.

Meta-training. For each sampled sequence $\mathcal{D}^s$, we propose a bi-level optimization based on Meta-Learning [9] to directly formulate adapting with less forgetting as the meta-objective. We first denote $\theta = \{\theta^C, \theta^M, \theta^{FC}\}$ as the parameter for the whole network, where θ^C, θ^M, θ^{FC} denote the parameters for F^C, F^M and final fully connected layer, respectively. Note that θ^M includes the FC layers in BGM.

Algorithm 1 MetaFSCIL

Require: α, β: learning rates
Require: θ^C, θ^M: pre-trained weights
Require: $\mathcal{D}^0$: training set of base classes
Initialize the models with pre-trained weights
while *not converged* **do**
  $\mathcal{D}^s = \{(\mathcal{S}^j, \mathcal{Q}^j)\}_{j=0}^N$ ▷ Sample a pseudo incremental sequence
  $\mathcal{Q}^c = \varnothing$ ▷ Empty cumulative query set
  Discard θ_{fc} ▷ Discard FC layer from previous sequence
  for $j = 0, 1, ..., N$ **do** ▷ Loop through the whole sequence
    Warm-up θ_{fc}^{new} ▷ Train only new FC nodes for a few steps
    θ^{fc} = Concatenate(θ_{old}^{fc}, θ_{new}^{fc}) ▷ Merge the FC nodes
    $\tilde{\theta}^{C,fc} = \theta^{C,fc} - \alpha \nabla_{\theta^{C,fc}} \mathcal{L}_{CE}(\mathcal{X}_j^s, \mathcal{Y}_j^s; \theta)$
    ▷ Adapt θ^C and θ^{fc} to new session
    $\mathcal{Q}^c = \mathcal{Q}^c \cup \mathcal{Q}^j$ ▷ Accumulate the query set at session j
    $\theta^{C,M} \leftarrow \theta^{C,M} - \beta \nabla_{\theta^{C,M}} \sum_{(\mathcal{X}^q, \mathcal{Y}^q) \in \mathcal{Q}^c} \mathcal{L}_{CE}(\mathcal{X}^q, \mathcal{Y}^q$
    $; \tilde{\theta}^C, \tilde{\theta}^{fc}, \theta^M)$ ▷ Update meta parameters $\theta^{P,M}$.
  end for
end while

We first conduct supervised training of θ on the *base* classes using cross-entropy loss ($\mathcal{L}_{CE}$). After that, θ^{FC} is discarded. The meta-training procedure is illustrated in Alg. 1 and left of Fig. 2. At the beginning of training on each sequence, we define an empty cumulative query set $\mathcal{Q}^c$ to store the query sets from previous tasks. At the j^{th} task, we first randomly initialize new FC nodes as θ_{new}^{fc}. To reduce the impact from randomness, we train θ_{new}^{fc} alone for 20 iterations to bring it closer to local optima. We refer to this as a warm-up operation. θ_{new}^{fc} is then concatenated with θ_{old}^{fc} from previous tasks. Then, we start to perform fast adaptation to new classes and update θ^C and θ^{fc} via a few L gradient steps:

$$\tilde{\theta}^{C,fc} = \theta^{C,fc} - \alpha \nabla_{\theta^{C,fc}} \mathcal{L}_{CE}(\mathcal{X}_j^s, \mathcal{Y}_j^s; \theta). \quad (2)$$

$\mathcal{X}_j^s, \mathcal{Y}_j^s$ are the images and labels for support set in the j^{th} task. The loss term $\mathcal{L}_{CE}(\mathcal{X}, \mathcal{Y}; \theta)$ means that the loss is computed on the output of θ (given $\mathcal{X}$ as input) and the target label $\mathcal{Y}$. Note, for *pseudo base* set ($j = 0$), as it contains more images, we set the batch size larger so that after L iterations, the model sees all images once.

The adaptation process mimics how the model learns new classes at test time. Ideally, we would like the adapted parameters to perform well on the classes from the previous and current tasks. The query sets from previous task reflect how the updated model performs to resist the catastrophic forgetting while the current query set validates model adaptation to new classes. Thus, we append $\mathcal{Q}^j$ to $\mathcal{Q}^c$, and accordingly, the meta-objective is defined as:

$$\min_{\theta^C, \theta^M} \sum_{(\mathcal{X}^q, \mathcal{Y}^q) \in \mathcal{Q}^c} \mathcal{L}_{CE}(\mathcal{X}^q, \mathcal{Y}^q; \tilde{\theta}^C, \tilde{\theta}^{fc}, \theta^M). \quad (3)$$

Note that $\mathcal{L}(\cdot)$ is a function of $\tilde{\theta}^C$ but the optimization is

Methods	Venue	Sessions (MiniImageNet w/ ResNet18)									Average	Final
		0	1	2	3	4	5	6	7	8	Acc	Impro.
TOPIC [27]	CVPR2020	61.31	50.09	45.17	41.16	37.48	35.52	32.19	29.46	24.42	39.64	+24.77
Zhu *et.al* [33]	CVPR2021	61.45	63.80	59.53	55.53	52.50	49.60	46.69	43.79	41.92	52.75	+7.27
Cheraghian *et.al* [4]	ICCV2021	61.40	59.80	54.20	51.69	49.45	48.00	45.20	43.80	42.1	50.63	+7.09
CEC [32]	CVPR2021	72.00	66.83	62.97	59.43	56.70	53.73	51.19	49.24	47.63	57.75	+1.56
MetaFSCIL (ours)	-	**72.04**	**67.94**	**63.77**	**60.29**	**57.58**	**55.16**	**52.9**	**50.79**	**49.19**	**58.85**	

Methods	Venue	Sessions (CIFAR100) w/ ResNet20									Average	Final
		0	1	2	3	4	5	6	7	8	Acc	Impro.
TOPIC [27]	CVPR2020	64.10	55.88	47.07	45.16	40.11	36.38	33.96	31.55	29.37	42.62	+20.6
Zhu *et.al* [33]	CVPR2021	64.10	65.86	61.36	57.34	53.69	50.75	48.58	45.66	43.25	54.51	+6.72
Cheraghian *et.al* [4]	ICCV2021	62.00	57.00	56.7	52.00	50.60	48.8	45.00	44.00	41.64	50.86	+8.33
CEC [32]	CVPR2021	73.07	68.88	65.26	61.19	58.09	55.57	53.22	51.34	49.14	59.53	+0.83
MetaFSCIL (ours)	-	**74.50**	**70.10**	**66.84**	**62.77**	**59.48**	**56.52**	**54.36**	**52.56**	**49.97**	**60.79**	

Methods	Venue	Sessions (CUB200) w/ ResNet18											Average	Final
		0	1	2	3	4	5	6	7	8	9	10	Acc	Impro.
TOPIC [27]	CVPR2020	68.68	62.49	54.81	49.99	45.25	41.40	38.35	35.36	32.22	28.31	26.28	43.92	+26.36
Zhu *et.al* [33]	CVPR2021	68.68	61.85	57.43	52.68	50.19	46.88	44.65	43.07	40.17	39.63	37.33	49.32	+15.31
Cheraghian *et.al* [4]	ICCV2021	68.78	59.37	59.32	54.96	52.58	49.81	48.09	46.32	44.33	43.43	43.23	51.84	+9.41
CEC [32]	CVPR2021	75.85	71.94	68.50	63.50	62.43	58.27	57.73	55.81	54.83	53.52	52.28	61.33	+0.36
MetaFSCIL (ours)	-	**75.90**	**72.41**	**68.78**	**64.78**	**62.96**	**59.99**	**58.30**	**56.85**	**54.78**	**53.82**	**52.64**	**61.92**	

Table 1. **Comparison with the state-of-the-art methods on MiniImageNet, CIFAR100 and CUB200 datasets.** Other results are copied from the corresponding papers. Our method yields superior results among all incremental sessions from all datasets.

performed on θ^C. We then minimize the objective in Eq. 3 using gradient decent:

$$\theta^{C,M} \leftarrow \theta^{C,M} - \beta \nabla_{\theta^{C,M}} \sum_{(\mathcal{X}^q, \mathcal{Y}^q) \in \mathcal{Q}^c} \mathcal{L}_{CE}(\mathcal{X}^q, \mathcal{Y}^q; \tilde{\theta}^C, \tilde{\theta}^{fc}, \theta^M). \tag{4}$$

When all $N+1$ tasks in an epoch are done, $\mathcal{Q}^c$ is reset to empty. θ_{fc} is also discarded because we are not able to pre-define the length of θ_{fc} after deployment, as it can be dynamically extended to any length by the users. Therefore, for every new epoch, we start θ^{fc}_{new} from random initialization followed by warm-up operation and adaptation process.

Meta-testing. The meta parameter θ^C is learned to perform fast adaptation via a few examples of new classes. And θ^M is trained to facilitate the learning process with less forgetting according to the new data and current state of θ^C. During the online incremental learning stage, we perform Line 6-9 of Alg. 1 to learn novel classes at evaluation. Note that after offline training, θ^M is fixed. The procedure in Alg. 1 matches the evaluation protocol: at each incremental session, the model is evaluated on all encountered classes after training on the current session. Our meta-objective optimizes the model towards what it is supposed to do at evaluation. We name the proposed Alg. 1 as MetaFSCIL.

4. Experiments

In this section, we conduct experiments on three well-know FSCIL datasets: CIFAR100 [14], MiniImageNet [23] and CUB200 [29]. We first discuss the datasets, evaluation protocol, and implementation details. Then, we compare with the state-of-the-art methods and conduct thorough ablation studies to analyze the proposed method.

4.1. Datasets and Implementation Details

Datasets. CIFAR100 [14] and MiniImageNet [23] contain 100 classes. Each class has 500 images for training and 100 images for testing. The images resolutions are 32×32 and 84 × 84, respectively. CUB200 [29] contains 6,000 images with resolution of 224×224 for both training and testing for 200 bird categories.

Evaluation protocol. We follow the splits in [27]. For CIFAR100 and MiniImageNet, 60 classes are chosen as *base* classes and the rest are divided into 8 incremental sessions with *5-way 5-shot* settings. As for CUB200, 100 classes are chosen as *base* set. The rest classes are formed as *10-way 5-shot* tasks for a total of 10 sessions.

Network. Following [27,32], we use ResNet20 as the backbone for CIFAR100 and ResNet18 for MiniImageNet and CUB200. The same structure is used for F^C and F^M. We uniformly distribute the BGM modules to the layers where there is a downsampling operation. We also apply it to the last activation as in [1]. Therefore, there are a total of 4 and 5 BGM modules for ResNet20 and ResNet18, respectively.

Pre-training. We follow [32] to perform supervised training on *base* classes for 100 epochs using SGD with a batch size of 128. The initial learning rate is set to 0.1 with scheduled decays by a factor of 0.1 at 60^{th} and 70^{th} epoch.

Sequential task sampling. We use the training set of *base* classes to sample the sequences. We first split them into non-overlapping support and query pools (250/250 for CIFAR100/MiniImageNet, 20/10 for CUB200). For each se-

Methods	Sessions (CIFAR100 w/ ResNet20)								
	0	1	2	3	4	5	6	7	8
Baseline (Rep.)	74.33	67.23	63.18	59.24	56.03	53.05	50.66	48.69	46.47
Baseline (Init.)	74.33	66.78	62.30	57.18	54.33	51.68	48.73	46.67	43.80
+Meta-learning (Rep.)	74.45	70.03	65.75	61.69	58.68	55.81	53.68	51.68	49.30
+Meta-learning (Init.)	74.45	70.05	65.97	61.76	58.78	55.92	53.80	51.77	49.41
+Modulation (Last)	74.46	70.08	66.65	62.06	58.88	55.58	53.28	51.12	48.34
+Modulation (Uniform)	74.49	70.08	**67.00**	62.45	59.38	56.29	54.08	52.02	49.67
+BGM (Full model)	**74.50**	**70.10**	66.84	**62.77**	**59.48**	**56.52**	**54.36**	**52.56**	**49.97**
Meta-update (OML [11], ANML [1])	**74.50**	70.09	66.19	61.89	58.51	55.53	53.81	51.89	49.02
Meta-update (ours)	**74.50**	**70.10**	**66.84**	**62.77**	**59.48**	**56.52**	**54.36**	**52.56**	**49.97**

Table 2. **Overall ablation studies for the proposed method on CIFAR100.** With the proposed meta-learning, accuracy is boosted due to the alignment of learning objective and evaluation. Modulating the F^C (especially with BGM), further helps learning new classes. The proposed way of meta-update is also superior compared to the existing methods.

quence, we randomly sample *pseudo base* classes first (20 classes with 50 images each for CIFAR100/MiniImageNet and 15 images each for CUB200), followed by 8 sessions of *5-way 5-shot* tasks. For all classes, 50 images are randomly sampled as the query set for CIFAR100/MiniImageNet, and 5 images are sampled for CUB200.

Meta-training and testing. After pre-training, the model is further trained using Alg. 1 for 200 epochs with a fixed learning rate of 0.001 for both α and β. For the final evaluation in [27], we perform meta-testing using Line 6-9 of Alg. 1. For both meta-training and testing, we perform 5 gradient updates ($L = 5$) for learning new classes. We augment the data using random cropping, scale and horizontal flip for both pre-training, and meta-training. We also utilize Data Init. as in [32] for CUB200.

4.2. Main Results

In this section, we compare with recent state-of-the-art methods, including TOPIC [27], Zhu *et.al* [33], Cheraghian *et.al* [4], CEC [32]. We report top-1 accuracy for each incremental session and the average of all sessions. We also include the relative improvement for the final session. As reported in Table 1, the proposed method outperforms all the methods on all three dataset among all the incremental sessions. Specifically, we surpass the most recent method Cheraghian *et.al* [4] by 7.09%, 8.33% and 9.41% on MiniImageNet, CIFAR100 and CUB200 datasets for final accuracy. We also outperform the second-best method CEC [32] by 1.56%, 0.83% and 0.36%. It demonstrates the effectiveness of the proposed method where the model is directly optimized to incrementally learn with less forgetting.

Visualization. To intuitively show the effectiveness of the proposed method, we report the class-wise performance via the confusion matrices, as shown in Fig 4. As we can see, the baseline model performs poorly, especially when adapting to *novel* classes. However, the full model has significant gain in accuracy for adaptation on *novel* classes. On the other hand, less forgetting is also observed in our method, as more values are concentrated on the diagonal.

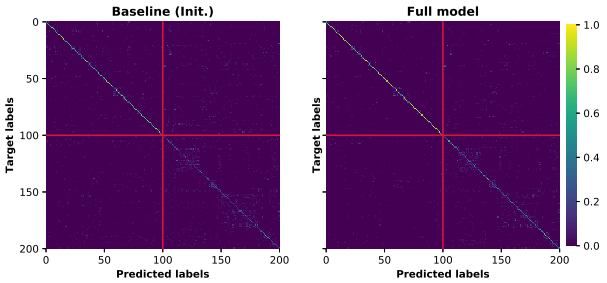

Figure 4. **Class-wise performance on CUB200 dataset.** The confusion matrices show that our method significantly improves the baseline for both *base* and *novel* classes (separated by red line).

4.3. Ablation Studies

In this section, we conduct ablation studies on CIFAR100 dataset to analyze various components of the proposed method. Table 2 reports the overall ablation results.

Baseline model. We consider two baseline models, which are pre-trained on the base classes without meta-learning and modulation. The pre-trained backbone is either fixed or updated during the incremental sessions. We refer to the fixed version as representation (**Rep.**) which is similar to the decouple learning in [32]. The non-fixed backbone serves as a model initialization (denoted as **Init.**) that is supposed to be updated for downstream sessions [9]. As reported in the first two rows of Table 2, the decoupled **Rep.** performs much better than **Init.**. Because the backbone is not trained on how to incrementally learn, updating it is more likely to overfit to new classes and suffers from catastrophic forgetting.

Meta-learning. As shown in the 3^{rd} and 4^{th} rows of Table 2, the meta-learned models greatly boost both baselines (+2.83% and +5.61% respectively). Notably, the **Init.** version outperforms the **Rep.** version. It demonstrates the effectiveness of the proposed method where the meta-objective specifically forces the model to learn new classes with less forgetting old classes.

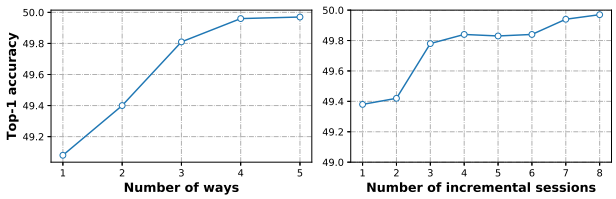

Figure 5. **Ablation studies on CIFAR100 dataset regarding various ways (left) and sampled incremental tasks (right)**. Increasing the ways or incremental tasks helps to stabilize the training process and improve performance.

Modulation network. The modulation network is another key to improving the performance. However, simply modulating the last activation as in [1] of F^C is undesirable when deeper network and complex datasets are employed. As shown in row 5 of Table 2, modulating only the last activation (denoted as **Last**) even drops the performance by 1.07% compared with the one without modulation (row 4). The experiments conducted in [1] only involve shallow networks and simple hand-written dataset [15]. It is possible that the modulation effect gradually diminishes through backpropagation to early layers for deeper networks. In contrast, our strategy of placing the modulation units throughout F^C (denoted as **Uniform**) can bring positive effect from modulation, as shown in row 6.

Bi-directional guided modulation: In addition to the modulation network, the BGM module also considers the learning capability of F^C at each session. According to the old knowledge and the upcoming images for new classes, BGM is able to modulate the F^C more accurately. It is noted that, as shown in row 7 of Table 2, without BGM, the model is able to perform well for the first two incremental sessions. However, starting with the 3^{rd} session, the performance drops faster than the model with BGM. Eventually, BGM further improves the final accuracy by 0.3%.

Meta-update. During training, the meta-update of the proposed method is performed on every incremental task in a sequence. This design matches the evaluation protocol. We compare with the methods in the online continual learning setting [1, 11], where the meta-update is performed only at the end of a sequence. Note that the meta-objectives of [1, 11] are defined on a set that contains both previous classes and randomly sampled classes that contain future classes. For fair comparison, we ignore the future classes and increase the number of epochs to match the total number of iterations. The last two rows of Table 2 demonstrate that our meta-update is more optimal for the FSCIL setting.

Number of classes in task sampling. We investigate the number of new classes for sampling the tasks during meta-training. We set this number in the range of $\{1, 5\}$ and train separate models. The left figure of Fig. 5 illustrates that a larger number of classes for each task is more optimal. With a smaller number of classes, the model is more likely over-fitted to particular classes instead of learning how to incrementally learn.

Methods	MiniImageNet	CIFAR100	CUB200
CEC [32]	47.63	49.14	52.28
MetaFSCIL + CEC	48.95	49.71	52.64

Table 3. **Integration of our meta-learned backbone with CEC.** Compare with the plain backbone that is naively trained on *base* classes; our meta-learned backbone significantly improves CEC.

Number of incremental tasks. The number of incremental tasks for each sequence during meta-training is vital for the training process. As shown in the right of Fig. 5, when only one incremental task is sampled, the sampling process degrades to the ones as in [32, 33]. A significant decrease in accuracy is observed, as the sampling process differs a lot compared to the evaluation protocol. At evaluation, catastrophic forgetting becomes more severe as time goes. Increasing the number of sessions can force the model to learn in a way that better matches the scenario it will encounter during evaluation.

Integration with CEC. Our proposed method aims to learn a backbone (coupled F^C and F^M) as an appropriate initialization for learning with less forgetting. CEC [32] has developed an advanced classifier with a fixed backbone that is pre-trained on *base* classes, which is obviously sub-optimal. To further show the effectiveness of the proposed method, we replace their backbone with our meta-trained one and re-train their classifier. Table 3 illustrates that our backbone is able to improve CEC by 1.32%, 0.57% and 0.36% on MiniImageNet, CIFAR100 and CUB200 datasets.

5. Conclusion

In this work, we have introduced meta-learning approach for few-shot class incremental learning. Previous work usually uses various hand-engineered approaches. In contrast, our model is specifically trained to effectively learn new classes using a few examples without forgetting old classes. A bi-directional guided modulation (BGM) is also proposed to automatically guide the adaptation process. BGM is trained to better achieve *activation selection* based on the data from novel classes and learned knowledge encoded in the classification network. Extensive experiments are conducted to show the effectiveness of the proposed method compared with the existing state-of-the-art methods.

Limitation and future works. Our FSCIL setting assumes the number of new classes and the number of shots is fixed in each incremental session. As future work, we would like to explore how to handle variable numbers of new classes and shots in different sessions since this is closer to real-world applications.

References

[1] Shawn Beaulieu, Lapo Frati, Thomas Miconi, Joel Lehman, Kenneth O Stanley, Jeff Clune, and Nick Cheney. Learning to continually learn. In *European Conference on Artificial Intelligence*, 2020. 2, 3, 4, 6, 7, 8

[2] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *European Confererence on Computer Vison*, 2018. 1

[3] Ali Cheraghian, Shafin Rahman, Pengfei Fang, Soumava Kumar Roy, Lars Petersson, and Mehrtash Harandi. Semantic-aware knowledge distillation for few-shot class-incremental learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 2

[4] Ali Cheraghian, Shafin Rahman, Sameera Ramasinghe, Pengfei Fang, Christian Simon, Lars Petersson, and Mehrtash Harandi. Synthesized feature based few-shot class-incremental learning on a mixture of subspaces. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 6, 7

[5] Zhixiang Chi, Yang Wang, Yuanhao Yu, and Jin Tang. Test-time fast adaptation for dynamic scene deblurring via meta-auxiliary learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 3

[6] Jeff Clune. Ai-gas: Ai-generating algorithms, an alternate paradigm for producing general artificial intelligence. *arXiv preprint arXiv:1905.10985*, 2019. 2

[7] Songlin Dong, Xiaopeng Hong, Xiaoyu Tao, Xinyuan Chang, Xing Wei, and Yihong Gong. Few-shot class-incremental learning via relation knowledge distillation. In *AAAI Conference on Artificial Intelligence*, 2021. 2, 3

[8] Zhibo Fan, Yuchen Ma, Zeming Li, and Jian Sun. Generalized few-shot object detection without forgetting. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 2

[9] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, 2017. 2, 5, 7

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 1, 4

[11] Khurram Javed and Martha White. Meta-learning representations for continual learning. In *Advances in Neural Information Processing Systems*, 2019. 3, 7, 8

[12] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017. 3, 4

[13] Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2. Lille, 2015. 2

[14] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 6

[15] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015. 3, 8

[16] Yann LeCun, John S Denker, and Sara A Solla. Optimal brain damage. In *Advances in neural information processing systems*, pages 598–605, 1990. 4

[17] Shikun Liu, Andrew J Davison, and Edward Johns. Self-supervised generalisation with meta auxiliary learning. In *Advances in Neural Information Processing Systems*, 2019. 3

[18] Davide Maltoni and Vincenzo Lomonaco. Continuous learning in single-incremental-task scenarios. *Neural Networks*, 116:56–73, 2019. 3

[19] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989. 1

[20] Niamul Quader, Md Mafijul Islam Bhuiyan, Juwei Lu, Peng Dai, and Wei Li. Weight excitation: Built-in attention mechanisms in convolutional neural networks. In *European Conference on Computer Vision*, 2020. 4

[21] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *International Conference on Learning Representations*, 2016. 2

[22] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 1

[23] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 6

[24] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International Conference on Machine Learning*, 2016. 2

[25] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems*, 2017. 1

[26] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2014. 1

[27] Xiaoyu Tao, Xiaopeng Hong, Xinyuan Chang, Songlin Dong, Xing Wei, and Yihong Gong. Few-shot class-incremental learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1, 3, 6, 7

[28] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, 2016. 2

[29] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 6

[30] LI Xuhong, Yves Grandvalet, and Franck Davoine. Explicit inductive bias for transfer learning with convolutional networks. In *International Conference on Machine Learning*, 2018. 3

[31] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*, 2017. 3

[32] Chi Zhang, Nan Song, Guosheng Lin, Yun Zheng, Pan Pan, and Yinghui Xu. Few-shot incremental learning with continually evolved classifiers. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 2, 3, 5, 6, 7, 8

[33] Kai Zhu, Yang Cao, Wei Zhai, Jie Cheng, and Zheng-Jun Zha. Self-promoted prototype refinement for few-shot class-incremental learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 2, 3, 5, 6, 7, 8