

Adaptive Additive Parameter Updates of Vision Transformers for Few-Shot Continual Learning

Kyle Stein¹, Andrew Arash Mahyari^{1,2}, Guillermo Francia, III³, Eman El-Sheikh³

¹ Department of Intelligent Systems and Robotics, University of West Florida, Pensacola, FL, USA

² Florida Institute For Human and Machine Cognition (IHMC), Pensacola, FL, USA

³ Center for Cybersecurity, University of West Florida, Pensacola, FL, USA

ks209@students.uwf.edu, amahyari@ihmc.org, gfranciaiii@uwf.edu, eelsheikh@uwf.edu

Abstract—Integrating new class information without losing previously acquired knowledge remains a central challenge in artificial intelligence, often known as catastrophic forgetting. Few-shot class incremental learning (FSCIL) addresses this by first training a model on a robust set of base classes and then incrementally adapting it in successive sessions using few labeled examples per novel class. However, this approach is prone to overfitting on limited new data, which can compromise performance and exacerbate forgetting. In this work, we propose a simple yet effective FSCIL framework that leverages a frozen Vision Transformer (ViT) backbone augmented with parameter-efficient additive updates. Our approach freezes the pre-trained ViT parameters and selectively injects trainable weights into the self-attention modules via an additive update mechanism. This design updates a small subset of parameters to accommodate new classes without sacrificing representations learned during the base session. By fine-tuning only a few parameters, our method preserves the generalizable features in the frozen ViT while reducing overfitting risk. Furthermore, as most parameters remain fixed, the model avoids overwriting knowledge when novel data batches are introduced. Extensive experiments on benchmark datasets demonstrate that our approach yields state-of-the-art performance compared to baseline FSCIL methods. Our results confirm improvements in robustness and accuracy.

Index Terms—Class-incremental learning, few-shot learning, vision transformers, parameter-efficient fine tuning

I. INTRODUCTION

Deep neural networks (DNNs) have shown robust performance in replicating human brain abilities, from object recognition [1], [2] to complex language analysis [3], [4]. However, a key assumption for their success is the availability of abundant annotated training data. In contrast, humans can rapidly recognize and retain new objects after observing them once or a few times. This discrepancy becomes more evident when models are required to incrementally learn new classes over time, a process that frequently leads to *catastrophic forgetting*. Catastrophic forgetting is a paradigm that occurs when previously learned knowledge is overwritten by newly acquired information [5]. Class-incremental learning (CIL) [6], [7] aims to address this challenge by enabling models to learn new classes over time while still retaining knowledge from previously encountered classes.

Although CIL has shown promise in mitigating catastrophic forgetting, existing methods still require a substantial number of samples for each new incremental class [8], [9]. Few-shot class-incremental learning (FSCIL) seeks to overcome this limitation by enabling models to incrementally learn new

classes from only a few labeled samples, while still preserving the knowledge of previously encountered classes [10], [11]. FSCIL consists of a base session and subsequent incremental sessions. During the base session, the model is trained on large amounts of data, enabling it to learn fine-grained features and establish robust prior knowledge that can be effectively transferred to new classes. In contrast, incremental sessions introduce new classes with only a few samples each, which may cause the model to overfit, since these limited examples might not accurately represent the true distribution of the new classes.

Earlier works [12], [13], [14], [15] have used various ResNet models [16] to mitigate overfitting; however, these models often struggle to transfer knowledge from the base session to incremental sessions. Recent studies [17], [18], [19] have adopted the more powerful Vision Transformers (ViT) [20] to enhance base session performance. Nonetheless, since ViTs are pre-trained on large-scale datasets like ImageNet [21], they are particularly susceptible to catastrophic forgetting when fine-tuned on only a few novel examples. In few-shot and incremental learning scenarios, the limited and distributionally different fine-tuning data can easily lead to overwriting the robust pre-trained features, thereby amplifying this risk of catastrophic forgetting.

To prevent performance degradation in incremental sessions, a promising solution lies in parameter-efficient fine-tuning (PEFT). PEFT is a strategy for adapting a large, pre-trained model to a new task by making only minimal changes to its parameters [22]. A prominent submethod, prompt tuning [23], [24] [25], adapts the model by prepending context-specific learnable tokens into the model’s input layer. Previous FSCIL methods have employed prompt tuning techniques [24]; however, prompt tuning has been shown to converge slower and more difficult to optimize than other PEFT methods [22], [26]. In contrast, alternative PEFT methods [27] operate on the internal layers of the backbone model, such as the self-attention modules within transformer blocks, by injecting learnable matrices while keeping the pre-trained weights frozen. This approach avoids the overhead of processing additional input tokens and mitigates the optimization issues commonly encountered with prompt tuning.

Motivated by these insights, we propose a novel parameter-efficient approach to address catastrophic forgetting and overfitting in FSCIL. We leverage a pre-trained ViT with a **key**

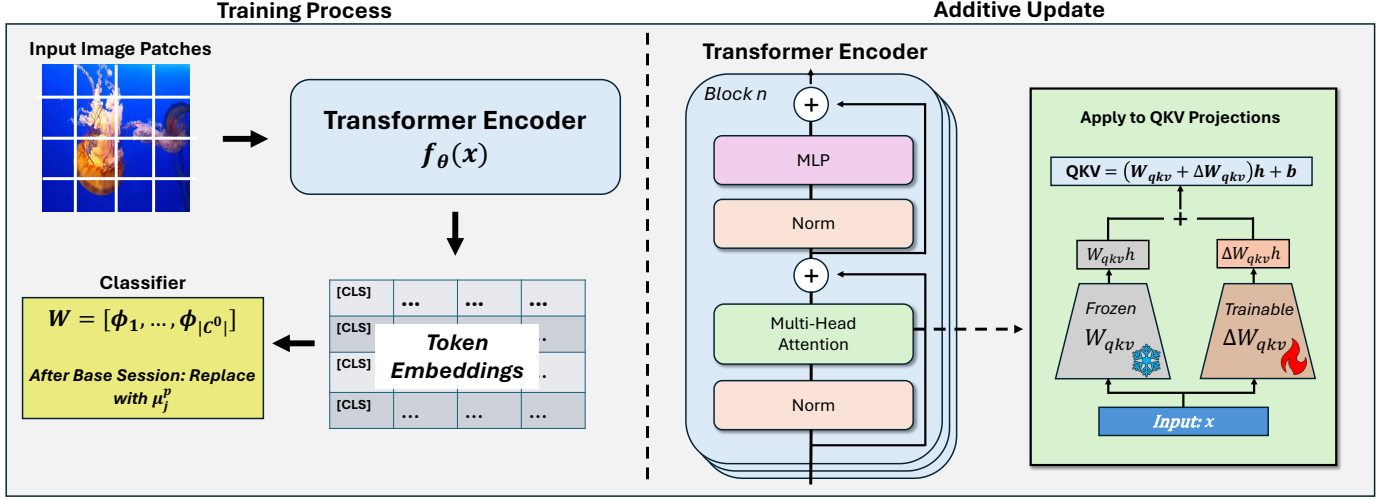


Fig. 1. The overall architecture of the proposed FSCIL approach.

contribution: an additive update mechanism. Unlike extensive fine-tuning or traditional prompt tuning, which require re-training large portions of the model or adding extra modules that are difficult to optimize, our method selectively updates only the self-attention layers within the frozen ViT backbone. This targeted adaptation preserves the robust pre-trained representations while enhancing computational efficiency and ease of integration. Our approach shares the same additive parameters across all blocks of the transformer, which reduces the number of additive parameters significantly while preserving the performance. In addition, we refine the classifier by recalibrating each class-specific weight vector to better align with the relevant feature distribution. Together, these components enable our approach to effectively mitigate catastrophic forgetting and overfitting in FSCIL, achieving state-of-the-art performance over baseline methods.

II. RELATED WORK

A. Few-Shot Class-Incremental Learning

Classic CIL [6], [9] continually learns new classes using ample data per incremental session, while FSCIL [10] mimics human learning by adapting to new classes with only a few samples after extensive base training. One common perspective divides methods into metric-based and fine-tuning based strategies. The metric-based methods [13], [28], [29] focus on robust prototype representation and similarity metrics, similar to traditional few-shot learning techniques [30], [31], [32]. Fine-tuning based approaches [33], [34] leverage extensive base class training to initialize the model, followed by careful fine-tuning on novel classes. Recently, the focus has shifted from shallower models such as ResNet for feature extraction to leveraging the more powerful ViT [17], [24], [35]. However, fine-tuning powerful models like ViT for incremental sessions is challenging because of catastrophic forgetting. Existing methods mitigate catastrophic forgetting by freezing the pre-trained ViT backbone to preserve its learned representations

and incorporating prompt tuning-based approaches [17], [24]. However, prompt tuning is known to converge slower and is more difficult to optimize [22], [26]. In contrast, our approach shares updates to the self-attention layers within transformer blocks of the frozen backbone, effectively preventing catastrophic forgetting while keeping the model adaptation strategy simple.

B. Parameter-Efficient Fine-Tuning

The goal of PEFT is to adjust a limited number of large model parameters while keeping the remainder of the parameters unchanged [36]. PEFT can be summarized into reparameterization-based methods, specification-based methods, and addition-based methods [22]. Reparameterization-based methods operate on insight that the adaptations required for many tasks reside in a low-dimensional subspace of the full parameter space [27], [37]. Specification-based methods do not add additional parameters to the model and optimize a pre-selected subset of the model's existing parameters, such as updating strictly bias terms in a pre-trained network [26]. Addition-based methods introduce extra parameters into the network without modifying the original pre-trained weights [25], [38]. In our work, we propose a targeted addition-based approach to the self-attention layers of transformer blocks. This approach adapts the model by adding small adjustments only in the self-attention layers, leaving the main pre-trained parameters intact. Moreover, we share a single additive parameter across all selected transformer blocks, further reducing the number of additional parameters required for adaptation.

III. PRELIMINARIES

In FSCIL, a model is trained sequentially over multiple sessions, denoted as $D = \{D^{(0)}, D^{(1)}, \dots, D^{(T)}\}$, where the base session is denoted as $D^{(0)}$ and T is the total number of incremental tasks. Each incremental session t provides a training dataset: $D^{(t)} = \{(x_i, y_i)\}_{i=1}^{|D^{(t)}|}$, where x_i is an input sample and y_i is its corresponding label. In the base

session $D^{(0)}$, a comprehensive label space $C^{(0)}$ is provided along with ample data for each class, thereby establishing a robust foundation for base model training. However, for each incremental session $D^{(t)}$, with $1 \leq t \leq T$, only a few training samples per class are available, arranged in an N -way K -shot format, where N new classes are introduced with K examples each. Notably, classes introduced in different sessions are entirely distinct: for any $t \neq t'$, the sets $C^{(t)}$ and $C^{(t')}$ are disjoint. During inference, the model is evaluated on all classes encountered up to that point. Specifically, after training session t , testing is performed on a dataset $D_{\text{test}}^{(\leq t)}$, which consists solely of examples whose labels belong to the cumulative set $C^{(\leq t)} = \bigcup_{j=0}^t C^{(j)}$. The model's goal is to incrementally incorporate new classes from only a few examples, while maintaining its ability to accurately classify all previously learned classes.

IV. PROPOSED METHOD

To address the limitations of existing approaches for FSCIL, we propose a novel framework that adapts a pre-trained ViT backbone to new classes by fine-tuning few parameters in base session, *i.e.*, PEFT. Our method leverages additive updates to the self-attention layers within each transformer block, while keeping pre-trained parameters frozen. Specifically, we apply updates to the query, key, and value (QKV) projection matrices. These projection matrices capture the contextual relationships among patches in each input image through positional embeddings and self-attention. This design enables the model to effectively incorporate new class information from only a few labeled examples without compromising the robust feature representations learned during the base session. In addition, we refine the classifier by recalibrating each class-specific weight vector to better align with the evolving feature distribution. The overall architecture of the proposed design is shown in Figure 1.

A. Feature Extraction

Following prior works [17], [24], [35], our classifier employs a pre-trained ViT model $f_{\theta}(\cdot)$ as the feature extractor. The ViT is initialized with weights pre-trained on large-scale datasets (*i.e.*, ImageNet-21K), and its original classification head is removed to yield raw feature representations. Specifically, given an input image $x \in \mathbb{R}^{C \times H \times W}$, the image is first divided into patches and embedded into a sequence. A classification token ([CLS]) is prepended to this sequence, and positional encodings are added. The transformer encoder then processes this sequence to output $h = f_{\theta}(x) \in \mathbb{R}^{N \times D}$, where h denotes the sequence of token embeddings, N is the total number of tokens, and D is the embedding dimension. The [CLS] token, which captures the global image representation, is extracted and denoted as $z = h_{\text{cls}} \in \mathbb{R}^D$, and is subsequently normalized to obtain the final representation from the transformer encoder.

B. Additive Update Mechanism in Self-Attention Layers

To facilitate controlled adaptation during the base session, we introduce an additive update mechanism that specifically

targets the QKV projections within the self-attention modules of each transformer block. In a standard ViT, these projections are computed as follows:

$$\begin{aligned} Q &= W_q h + b_q, \\ K &= W_k h + b_k, \\ V &= W_v h + b_v. \end{aligned} \quad (1)$$

Here, W_q , W_k , and W_v are the pre-trained weight matrices, and b_q , b_k , and b_v are the corresponding bias terms. In our approach, each projection is modified by adding a trainable update to the corresponding weight matrix while retaining the original trainable bias:

$$\begin{aligned} Q &= (W_q + \Delta W_q) h + b_q, \\ K &= (W_k + \Delta W_k) h + b_k, \\ V &= (W_v + \Delta W_v) h + b_v. \end{aligned} \quad (2)$$

Specifically, ΔW_q , ΔW_k , and ΔW_v are the trainable additive parameters (initialized to zero). Notably, we **share a single ΔW_q , ΔW_k , and ΔW_v across all selected transformer blocks**, which significantly reduces the number of parameters being updated throughout training process [39].

C. Training Procedure

Base Session. During the base session, the network is trained on a set of base classes using a softmax cross-entropy loss computed on logits derived from cosine similarity. For each training sample x_i , the feature extractor $f_{\theta}(\cdot)$ produces a final feature vector $z = h_{\text{cls}} \in \mathbb{R}^D$, from the [CLS] token which is then normalized. The classifier is implemented as a linear layer with a weight matrix:

$$W = [\phi_1, \phi_2, \dots, \phi_{|C^{(0)}|}], \quad (3)$$

where each $\phi_j \in \mathbb{R}^D$ is a learned weight vector of class j . The classifier maps an input's normalized feature z_i to a set of logits by computing the cosine similarity between z_i and each normalized classifier weight $\hat{\phi}_j$ (where $\hat{\phi}_j = \phi_j / \|\phi_j\|_2$). During base training, these classifier weights are learned via gradient descent to align the feature embeddings with the correct class. The cosine similarity is defined as:

$$\text{sim}(z_i, \phi_j) = z_i^\top \hat{\phi}_j. \quad (4)$$

To obtain the logit for class j , we scale this similarity by a temperature parameter τ :

$$\ell_{i,j} = \tau \text{sim}(z_i, \phi_j). \quad (5)$$

In the softmax cross-entropy loss, $\exp(\ell_{i,y_i})$ corresponds to the exponential of the logit for the ground-truth class y_i of sample i . The loss over a batch of size B is then defined as:

$$\mathcal{L}_{\text{ce}} = \frac{1}{B} \sum_{i=1}^B -\log \left(\frac{\exp(\ell_{i,y_i})}{\sum_{j=1}^{|C^{(0)}|} \exp(\ell_{i,j})} \right). \quad (6)$$

Minimizing \mathcal{L}_{ce} encourages each feature vector z_i to align with its corresponding classifier weight while distancing it from those of other classes.

In addition to updating the classifier weights, the shared additive parameters W_q , W_k , and W_v , as well as the associated bias terms are also updated. During base training, the gradients from the cross entropy loss are backpropagated through all adapted blocks. Furthermore, since we share the same additive updates in each block, gradients are aggregated to update this single parameter via gradient decent. Meanwhile, the original pre-trained weight matrices W_q , W_k , and W_v remain frozen. In incremental sessions, all parameters of the feature extractor, including the additive updates, are frozen to preserve the representations learned during the base session. Only the classifier is updated for new classes.

Classifier Replacement. After the base session training is completed, we refine the classifier via *classifier replacement* [14]. Recall that the classifier is implemented as a linear layer with a weight matrix from Eq. 3, where each $\phi_j \in \mathbb{R}^D$ is the learned weight vector for class j . Previous works [14], [17], [24] have shown that reinitializing these weight vectors to better capture the underlying feature distributions can avoid overfitting and catastrophic forgetting. To reinitialize these weight vectors for each base class j , we define a learnable parameter μ_j . Let $\mathcal{S}_j = \{i \mid y_i = j\}$ denote the set of indices for image samples belonging to class j . The parameter μ_j for each class j is learned by solving the following optimization problem:

$$\min_{\mu_j} \frac{1}{2} \sum_{i \in \mathcal{S}_j} \|\mu_j - z_i\|^2. \quad (7)$$

Once the optimization is solved, we normalize μ_j and update the corresponding classifier weight ϕ_j with the normalized solution. This step ensures that the classifier’s weights accurately represent the aggregate features of their respective classes.

Incremental Sessions. In the incremental sessions, new classes are introduced in a N -way K -shot manner, meaning that each session presents a small set of new classes (way) that contain only a few labeled examples (shots). To mitigate overfitting and catastrophic forgetting, all parameters of the feature extractor remain frozen. Consequently, each few-shot sample is passed through the feature extractor to obtain its feature representation. For each new class j' , we define its weight vector $\mu_{j'}^P$ analogously to Eq. 7 using the available few-shot samples. We normalize $\mu_{j'}^P$ and append each new class to the classifier’s weight matrix:

$$\mu_{j'} \leftarrow \frac{\mu_{j'}^P}{\|\mu_{j'}^P\|_2}. \quad (8)$$

By updating the classifier in this manner, we incorporate new classes with minimal disruption to the existing feature representations, thus maintaining robust performance on previously learned classes.

V. EXPERIMENTAL RESULTS

Datasets and Metrics. Our method is evaluated on three widely used datasets in the FSCIL community: CUB-200

[43], CIFAR-100 [44], and miniImageNet [31]. We follow the same splits as previously published manuscripts [12], [13], [17], [18], [40], [42], [41]. For CUB-200, the base session contains 100 classes, and each incremental session introduces 10 new classes (ways) with 5 examples per class (shots), resulting in one base session plus 10 incremental sessions. For CIFAR-100 and miniImageNet, the base session consists of 60 classes, while the incremental sessions adopt a 5-way 5-shot configuration, yielding one base session followed by 8 incremental sessions. The base session accuracy is reported as S_{base} , the final session as S_{last} , the average accuracy across all sessions as S_{avg} , and the performance drop (PD) as the difference between S_{base} and S_{last} (lower the better).

Implementation. In our model, we implement two ViT backbones. The first backbone is ViT-B/16 pre-trained on ImageNet-21k and fine-tuned on ImageNet-1K (denoted as \dagger). The second backbone is the more powerful ViT-B/16 pre-trained on ImageNet-21K (denoted as \ddagger). We focus solely on shared additive updates to the self-attention layers of all 12 transformer blocks, following [27]. For all datasets, images are uniformly resized to 224×224 pixels to meet the input requirements of the ViT architecture. Our method was trained for 10 epochs on both CUB-200 and CIFAR-100, and 3 epochs on miniImageNet. We used the SGD optimizer on a single NVIDIA L4 GPU and maintained a batch size of 128 across all datasets.

A. Main Experimental Results

In this section, we discuss our main experimental results compared against baseline FSCIL methods. Session-wise results on CUB-200, CIFAR-100, and miniImageNet are shown in Tables I, II, and Figure 2, respectively.

On CUB-200, our method improves the base accuracy (S_{base}) by 3.25% and the final session accuracy (S_{last}) by 9.18% over the best competing methods, while reducing the performance drop (PD) by 0.66%. These improvements show that our selective parameter updates not only preserve base knowledge but also better integrate new classes over incremental sessions. On CIFAR-100, we achieve state-of-the-art accuracies during both the base and all incremental sessions. Although our PD (5.17%) is marginally higher than that of PriViLege (4.82%), our method avoids the extra complexity and parameter overhead associated with prompt tuning and word embeddings required by PriViLege. For miniImageNet, our approach attains stellar performance across all sessions, setting a new benchmark PD of 1.29%, a new base session accuracy of 98.13% and final session accuracy of 96.84% when using the ViT backbone fine-tuned on ImageNet-1K. Interestingly, the less powerful backbone (with ImageNet-1K fine-tuning) outperforms its counterpart, likely due to better alignment with the miniImageNet distribution. Overall, these results validate that our FSCIL framework, with its targeted additive updates and refined classifier weights, effectively maintains robust representations while accommodating new classes, leading to significant improvements over existing methods.

TABLE I
THE PERFORMANCE OF SESSIONS ON CUB200.

Method	S_{base}	1	2	3	4	5	6	7	8	9	S_{last}	S_{avg}	PD
CEC [†] [40]	75.40	73.23	72.00	68.70	69.35	67.78	67.01	66.40	65.78	65.57	65.70	72.41	9.70
L2P [†] [41]	44.97	30.28	27.21	24.44	22.41	20.81	19.47	18.19	17.16	16.26	15.41	24.99	29.56
DualPrompt [†] [42]	53.37	45.99	41.15	37.33	34.32	31.57	29.44	27.58	25.92	24.55	23.25	36.30	30.12
NC-FSCIL [†] [13]	78.49	71.52	65.54	60.30	55.81	51.96	48.72	45.78	43.18	40.92	38.80	57.92	39.69
WaRP [†] [12]	67.74	64.21	61.06	57.80	55.78	53.81	52.82	51.61	50.13	50.02	49.36	55.85	18.38
PL-FSCIL [†] [18]	85.16	85.40	82.75	75.22	77.22	73.25	72.39	70.24	67.97	68.33	69.86	75.25	15.30
PriViLege [†] [17]	82.21	81.25	80.45	77.76	77.78	75.95	75.69	76.00	75.19	75.19	75.08	77.50	4.71
Proposed Method[†]	87.47	85.61	84.46	82.70	82.70	81.14	80.60	81.02	80.47	80.67	80.86	82.52	6.61
Proposed Method[‡]	88.41	87.02	86.27	84.90	85.06	83.45	83.46	83.98	83.81	84.02	84.26	84.97	4.15

Note: Columns correspond to the base session (S_{base}), incremental sessions (1–9), the final session (S_{last}), the average accuracy (S_{avg}), and the performance drop (PD) between S_{base} and S_{last} .

TABLE II
THE PERFORMANCE OF SESSIONS ON CIFAR-100.

Method	S_{base}	1	2	3	4	5	6	7	S_{last}	S_{avg}	PD
CEC [†] [40]	74.20	71.49	70.11	67.34	65.96	65.14	64.74	63.48	61.48	67.10	12.72
L2P [†] [41]	83.29	76.81	71.29	66.53	62.38	58.68	55.42	52.49	49.87	64.08	33.42
DualPrompt [†] [42]	85.11	78.42	72.81	67.92	63.69	59.92	56.60	53.62	50.93	65.45	34.18
NC-FSCIL [†] [13]	89.51	82.62	76.72	71.61	67.13	63.18	59.67	56.53	53.70	68.96	35.81
WaRP [†] [12]	86.20	82.58	79.30	75.57	73.46	71.07	69.58	67.70	65.48	74.55	20.72
PL-FSCIL [†] [18]	85.73	74.54	74.77	72.42	72.98	72.87	72.49	71.62	75.13	74.73	10.60
PriViLege [†] [17]	90.88	89.39	88.97	87.55	87.83	87.35	87.53	87.15	86.06	88.08	4.82
CKPD-FSCIL [†] [35]	91.57	90.03	89.84	88.44	88.58	87.74	87.82	87.36	86.22	88.62	5.35
Proposed Method[†]	93.43	91.42	90.83	89.17	89.30	88.89	88.91	88.63	87.42	89.78	6.01
Proposed Method[‡]	93.92	92.12	91.67	90.48	90.60	90.21	90.16	89.83	88.75	90.86	5.17

Note: The columns represent the base session (S_{base}), incremental sessions (1–7), the final session (S_{last}), the average accuracy (S_{avg}), and the performance drop (PD) between S_{base} and S_{last} .

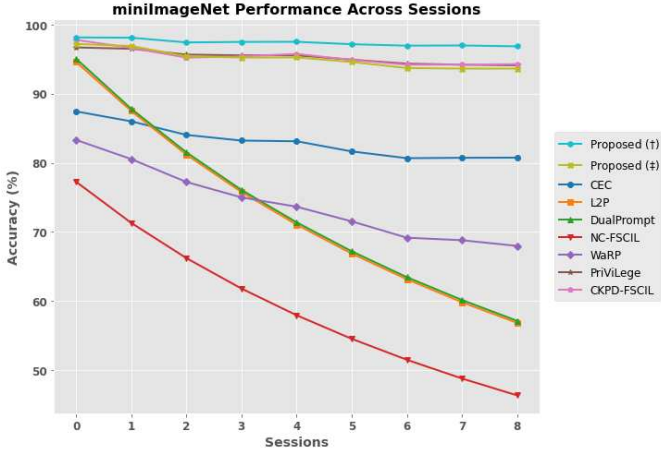


Fig. 2. miniImageNet performance across sessions.

B. Ablation Study

Number of Updated Self-Attention Layers. In the first ablation experiment, we study the effect of applying the additive update only to the final n self-attention layers of each transformer block in the ViT architecture. Note that when

TABLE III
ABLATION STUDY: IMPACT OF APPLYING ADDITIVE UPDATES TO ONLY THE FINAL n SELF-ATTENTION BLOCKS

Blocks	CUB-200		CIFAR-100	
	S_{avg} (%)	PD (%)	S_{avg} (%)	PD (%)
0	77.37	8.54	60.48	14.43
4	79.11	6.63	77.65	13.66
8	82.74	4.52	88.73	7.33
12	84.97	4.15	90.86	5.17

the additive update is applied to 0 blocks, the model remains identical to the frozen, pre-trained ViT, and only the classifier is updated during training.

Table III summarizes our findings on the CUB-200 and CIFAR-100 datasets. As we gradually increase the number of self-attention layers receiving additive updates, the model exhibits improved performance. On CUB-200, the average accuracy (S_{avg}) increases from 77.37% (with 0 updated blocks) to 84.97% (with all 12 blocks updated), resulting in a gain of 7.60 percentage points, while the PD is reduced by over 4 percentage points. Similarly, on CIFAR-100, S_{avg} improves

TABLE IV
COMPARISON OF SELF-ATTENTION AND MLP UPDATE STRATEGIES ON CUB-200 AND CIFAR-100

Metric	CUB-200		CIFAR-100	
	Self-Attn	MLP	Self-Attn	MLP
S_{base}	88.41 (+0.56)	87.85	93.92 (-0.55)	94.37
S_{last}	84.26 (+0.85)	83.41	88.75 (+0.56)	88.19
S_{avg}	84.97 (+0.80)	84.17	90.86 (+0.01)	90.85
PD	4.15 (+0.29)	4.44	5.17 (+1.01)	6.18

significantly from 60.48% to 90.86%, and PD decreases from 14.43% to 5.17%. The more pronounced improvements on CIFAR-100 can be attributed to its dataset split and incremental session configuration, which features fewer training base classes and a 5-way 5-shot format in each session, thereby making it more susceptible to catastrophic forgetting. These results suggest that updating a greater number of self-attention layers significantly enhances the model’s adaptability while preserving the robust representations learned during the base session.

Impact of Additive Updates on MLP Layers. In this experiment, we evaluate the impact of applying additive updates to the feed-forward (MLP) layers within each transformer block. Converse to our baseline, we instead freeze the self-attention layers and apply the additive update only on the MLP layers, then compare the resulting performance differences. Table IV presents a comparison between self-attention and MLP update strategies on CUB-200 and CIFAR-100 datasets.

On CUB-200, the self-attention update achieves a base accuracy of 88.41% (0.56 points higher), a final session accuracy of 84.26% (0.85 points higher), and a PD of 4.15% (0.29 points lower) compared to the MLP update. On CIFAR-100, although the MLP update shows a slightly higher base accuracy (94.37% vs. 93.92%), the self-attention update attains a higher final accuracy (88.75% vs. 88.19%) and a notably lower PD (5.17% vs. 6.18%). Notably, the self-attention update strategy also introduces fewer trainable parameters, $3d^2$, since it only modifies the QKV projection matrices. However, updating both fully connected layers in the MLP module introduces approximately $8d^2$ parameters, significantly increasing the parameter count.

VI. CONCLUSION

In this paper, we proposed a novel FSCIL framework that leverages a frozen pre-trained ViT backbone augmented with parameter-efficient additive updates. By selectively updating only the self-attention layers while keeping the pre-trained weights frozen, our method provides a simple strategy for effectively integrating new classes with only few-shot samples while preserving robust base representations. Extensive experiments on CUB-200, CIFAR-100, and miniImageNet demonstrate that our approach significantly outperforms previous state-of-the-art methods in terms of base accuracy, incremental accuracy, and performance drop. Our findings underscore

the advantages of targeted parameter tuning in mitigating catastrophic forgetting and reducing overfitting in FSCIL.

VII. ACKNOWLEDGMENT

This work is partially supported by the UWF Argo Cyber Emerging Scholars (ACES) program funded by the National Science Foundation (NSF) CyberCorps® Scholarship for Service (SFS) award under grant number 1946442. Any opinions, findings, and conclusions or recommendations expressed in this document are those of the authors and do not necessarily reflect the views of the NSF.

REFERENCES

- [1] Y. Li, C.-Y. Wu, H. Fan, K. Mangalam, B. Xiong, J. Malik, and C. Feichtenhofer, “Mvitv2: Improved multiscale vision transformers for classification and detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 4804–4814.
- [2] F. Boutros, N. Damer, F. Kirchbuchner, and A. Kuijper, “Elasticface: Elastic margin loss for deep face recognition,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 1578–1587.
- [3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of the North American Chapter of the Association for Computational Linguistics*, 2019.
- [4] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, “Learning transferable visual models from natural language supervision,” in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.
- [5] R. M. French, “Catastrophic forgetting in connectionist networks,” *Trends in cognitive sciences*, vol. 3, no. 4, pp. 128–135, 1999.
- [6] D.-W. Zhou, Q.-W. Wang, Z.-H. Qi, H.-J. Ye, D.-C. Zhan, and Z. Liu, “Class-incremental learning: A survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.
- [7] D.-W. Zhou, Z.-W. Cai, H.-J. Ye, D.-C. Zhan, and Z. Liu, “Revisiting class-incremental learning with pre-trained models: Generalizability and adaptivity are all you need,” *International Journal of Computer Vision*, pp. 1–21, 2024.
- [8] Z. Meng, J. Zhang, C. Yang, Z. Zhan, P. Zhao, and Y. Wang, “Diffclass: Diffusion-based class incremental learning,” in *European Conference on Computer Vision*. Springer, 2024, pp. 142–159.
- [9] W. Liu, X.-J. Wu, F. Zhu, M.-M. Yu, C. Wang, and C.-L. Liu, “Class incremental learning with self-supervised pre-training and prototype learning,” *Pattern Recognition*, vol. 157, p. 110943, 2025.
- [10] X. Tao, X. Hong, X. Chang, S. Dong, X. Wei, and Y. Gong, “Few-shot class-incremental learning,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 12 183–12 192.
- [11] S. Tian, L. Li, W. Li, H. Ran, X. Ning, and P. Tiwari, “A survey on few-shot class-incremental learning,” *Neural Networks*, vol. 169, pp. 307–324, 2024.
- [12] D.-Y. Kim, D.-J. Han, J. Seo, and J. Moon, “Warping the space: Weight space rotation for class-incremental few-shot learning,” in *The Eleventh International Conference on Learning Representations*, 2023.
- [13] Y. Yang, H. Yuan, X. Li, Z. Lin, P. Torr, and D. Tao, “Neural collapse inspired feature-classifier alignment for few-shot class incremental learning,” *arXiv preprint arXiv:2302.03004*, 2023.
- [14] J. Oh, S. Baik, and K. M. Lee, “Closer: Towards better representation learning for few-shot class-incremental learning,” in *European Conference on Computer Vision*. Springer, 2024, pp. 18–35.
- [15] Y. Zou, S. Zhang, H. Zhou, Y. Li, and R. Li, “Compositional few-shot class-incremental learning,” *arXiv preprint arXiv:2405.17022*, 2024.
- [16] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [17] K.-H. Park, K. Song, and G.-M. Park, “Pre-trained vision and language transformers are few-shot incremental learners,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 23 881–23 890.

- [18] J. Yuan, H. Chen, S. Tian, W. Li, L. Li, E. Ning, and Y. Zhang, "Prompt-based learning for few-shot class-incremental learning," *Alexandria Engineering Journal*, vol. 120, pp. 287–295, 2025.
- [19] Y.-S. Liang and W.-J. Li, "Inflora: Interference-free low-rank adaptation for continual learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 23 638–23 647.
- [20] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.
- [21] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.
- [22] N. Ding, Y. Qin, G. Yang, F. Wei, Z. Yang, Y. Su, S. Hu, Y. Chen, C.-M. Chan, W. Chen *et al.*, "Parameter-efficient fine-tuning of large-scale pre-trained language models," *Nature Machine Intelligence*, vol. 5, no. 3, pp. 220–235, 2023.
- [23] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," *arXiv preprint arXiv:2101.00190*, 2021.
- [24] C. Liu, Z. Wang, T. Xiong, R. Chen, Y. Wu, J. Guo, and H. Huang, "Few-shot class incremental learning with attention-aware self-adaptive prompt," in *European Conference on Computer Vision*. Springer, 2024, pp. 1–18.
- [25] K. Stein, A. Mahyari, G. Francia, and E. El-Sheikh, "Visual adaptive prompting for compositional zero-shot learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, 2025, pp. 4137–4146.
- [26] E. B. Zaken, S. Ravfogel, and Y. Goldberg, "Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models," *arXiv preprint arXiv:2106.10199*, 2021.
- [27] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen *et al.*, "Lora: Low-rank adaptation of large language models," *ICLR*, vol. 1, no. 2, p. 3, 2022.
- [28] Z. Chi, L. Gu, H. Liu, Y. Wang, Y. Yu, and J. Tang, "Metafsil: A meta-learning approach for few-shot class incremental learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 14 166–14 175.
- [29] D.-W. Zhou, H.-J. Ye, L. Ma, D. Xie, S. Pu, and D.-C. Zhan, "Few-shot class-incremental learning by sampling multi-phase tasks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 11, pp. 12 816–12 831, 2022.
- [30] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," *Advances in neural information processing systems*, vol. 30, 2017.
- [31] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra *et al.*, "Matching networks for one shot learning," *Advances in neural information processing systems*, vol. 29, 2016.
- [32] K. Stein, A. A. Mahyari, G. Francia, and E. El-Sheikh, "Transductive one-shot learning meet subspace decomposition," in *2025 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2025, pp. 576–581.
- [33] C. Peng, K. Zhao, T. Wang, M. Li, and B. C. Lovell, "Few-shot class-incremental learning from an open-set perspective," in *European Conference on Computer Vision*. Springer, 2022, pp. 382–397.
- [34] M. Hersche, G. Karunaratne, G. Cherubini, L. Benini, A. Sebastian, and A. Rahimi, "Constrained few-shot class-incremental learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 9057–9067.
- [35] X. Li, Y. Yang, J. Wu, D. A. Clifton, Y. Yu, B. Ghanem, and M. Zhang, "Continuous knowledge-preserving decomposition for few-shot continual learning," *arXiv preprint arXiv:2501.05017*, 2025.
- [36] Z. Han, C. Gao, J. Liu, J. Zhang, and S. Q. Zhang, "Parameter-efficient fine-tuning for large models: A comprehensive survey," *arXiv preprint arXiv:2403.14608*, 2024.
- [37] Q. Zhang, M. Chen, A. Bukharin, N. Karampatziakis, P. He, Y. Cheng, W. Chen, and T. Zhao, "Adalora: Adaptive budget allocation for parameter-efficient fine-tuning," *arXiv preprint arXiv:2303.10512*, 2023.
- [38] T. Gao, A. Fisch, and D. Chen, "Making pre-trained language models better few-shot learners," *arXiv preprint arXiv:2012.15723*, 2020.
- [39] S. Takase and S. Kiyono, "Lessons on parameter sharing across layers in transformers," *arXiv preprint arXiv:2104.06022*, 2021.
- [40] C. Zhang, N. Song, G. Lin, Y. Zheng, P. Pan, and Y. Xu, "Few-shot incremental learning with continually evolved classifiers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 12 455–12 464.
- [41] Z. Wang, Z. Zhang, C.-Y. Lee, H. Zhang, R. Sun, X. Ren, G. Su, V. Perot, J. Dy, and T. Pfister, "Learning to prompt for continual learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 139–149.
- [42] Z. Wang, Z. Zhang, S. Ebrahimi, R. Sun, H. Zhang, C.-Y. Lee, X. Ren, G. Su, V. Perot, J. Dy *et al.*, "Dualprompt: Complementary prompting for rehearsal-free continual learning," in *European conference on computer vision*. Springer, 2022, pp. 631–648.
- [43] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds-200-2011 dataset," 2011.
- [44] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.