Sculpting [CLS] Features for Foundation Model-Based Class-Incremental Learning

Murat Onur Yildirim
TU Eindhoven
m.o.yildirim@tue.nl

Elif Ceren Gok Yildirim
TU Eindhoven
e.c.gok@tue.nl

Joaquin Vanschoren
TU Eindhoven
j.vancshoren@tue.nl

Abstract

Foundation model-based class-incremental learners achieve strong performance but still struggle with the stability-plasticity trade-off; excessive plasticity heavily modifies the general knowledge of the pre-trained model and causes forgetting, and excessive stability hinders adaptation to new classes. This necessitates an effective adaptation that introduces minimal yet functional modifications. To address this, we first introduce a new parameter-efficient fine-tuning module 'Learn and Calibrate', or LuCA, designed to acquire task-specific knowledge through an adapter-calibrator couple, enabling well-refined feature representations. Then, for each task, we deploy a sparse LuCA module on top of the last classification token [CLS] just before the classifier, which we refer to as 'Token-level Sparse Calibration and Adaptation', or TOSCA. By leaving the generalization capabilities of the foundation models intact and adapting exclusively via the last token, our approach achieves a harmonious balance between stability and plasticity while reducing both training and inference complexity. We demonstrate that TOSCA yields state-of-the-art performance while introducing ~8$\times$ fewer parameters.

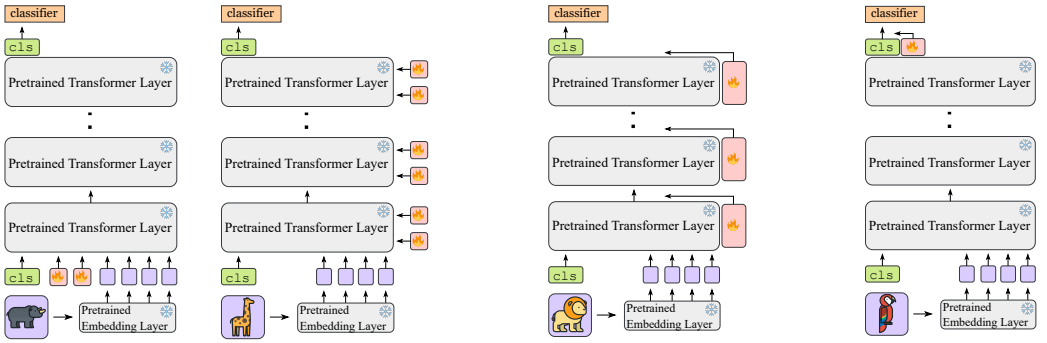
1 Introduction

Learning continuously from a series of concepts or classes using a unified model is a challenging problem due to catastrophic forgetting [1]. Class-incremental learning (CIL) addresses this issue by enabling models to acquire knowledge from new classes while preserving their ability to correctly classify previously learned categories [2]. With the rise of large Foundation Models (FMs) [3–5] such as Vision Transformers (ViTs) [6], many CIL methods now capitalize on the robust representations provided by FMs, marking a significant paradigm shift in the field. However, sequential fine-tuning of FMs alters the pre-trained representations, leading to substantial forgetting [7–9]. To tackle this, parameter-efficient fine-tuning (PEFT) methods such as learnable prompts [10–14] and lightweight adapters [15–17] restrict updates to small subsets of parameters. While this helps to mitigate forgetting, they introduce new trade-offs. Learnable prompts introduce small trainable embeddings, keeping the model's core parameters unchanged. This offers great stability but makes it challenging to adapt to specific tasks. In contrast, adapters provide localized feature refinement with high plasticity but this usually comes at the expense of quadratic parameter growth. This creates the stability-plasticity dilemma [18] and leads us to the core question in this work:

How can we efficiently tackle the stability-plasticity dilemma in FM-based CIL?

To address this question, we take inspiration from neuroscience, where the brain achieves continual learning by forming invariant representations in the ventral visual stream [19, 20], while flexibly adapting and modulating them through task-specific circuits in the prefrontal cortex [21–23]. In other words, the prefrontal cortex receives these stable visual representations and refines them through selective synaptic plasticity, enabling flexible adaptation to task demands and effectively guiding appropriate behavioral responses.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).



(a) Prompt-Based (b) Adapter-Based (c) TOSCA (ours)

Figure 1: Prompt-based methods influence the self-attention process of a FM, either from the input layer alone or across all layers. Adapter-based methods enable task-specific adaptations by inserting lightweight neural modules into the FM's layers. In contrast, we propose to train a single module that operates exclusively on the final `[CLS]` token representation, efficiently adapting and calibrating features just before classification. This design offers a streamlined and effective alternative to both prompt- and adapter-based methods.

Analogously, we aim to leverage a pre-trained model to emulate the ventral visual stream's role in extracting stable and invariant features, while introducing lightweight, task-adaptive modules just before the decision layer, mirroring the function of cortical circuits that flexibly tailor general representations to task-specific demands. This design minimizes redundant relearning of low-level features and aligns with principles of biological learning, promoting both efficiency and adaptability.

To this end, we first introduce a new PEFT module 'Learn and Calibrate', or *LuCA*, which comprises two components: (1) a residual adapter that applies task-specific feature transformations, and (2) a calibrator that reweighs and enhances the adapted features via attention-like gating. Then, to enable CIL, we train a single sparse LuCA module for each task, operating exclusively on the final `[CLS]` token representation of ViTs. We term this approach 'Token-level Sparse Calibration and Adaptation', or briefly *TOSCA*. By localizing adaptations at the final semantic aggregation point and preserving the low/mid-level feature hierarchy, TOSCA mirrors the harmony between the ventral visual stream and the prefrontal cortex. Our contributions are three-fold:

I. We introduce a novel PEFT module LuCA designed to learn task-specific residual transformations while refining features through additional calibration gating.

II. We propose TOSCA, a neuro-inspired FM-based CIL approach that integrates our LuCA module at the final semantic aggregation point in the network. This balances stability and plasticity while maintaining a model-agnostic parameter count, unlike many methods that scale linearly with the number of layers.

III. We validate TOSCA's advantages on six benchmarks where TOSCA yields (i) 7–21% higher accuracy than prompt-based methods and 4–12% higher than adapter-based methods on out-of-distribution datasets, (ii) ~2.5× faster runtime, and (iii) ~8× fewer parameters.

2 Related Work

FM-based CIL approaches aim to improve performance with minimal additions and modifications while freezing the FMs. L2P [12] borrows a technique from NLP by introducing a learnable prompt pool and selecting instance-specific prompts via a key-query matching selection mechanism to guide the FMs response. DualPrompt [13] extends L2P by designing G-Prompt and E-Prompt, which encode task-invariant and task-specific instructions respectively. CODA-Prompt [14] uses contrastive learning to decorrelate representations of the prompts to reduce interference and combine them by attention-based weighting method. APER [15] explores various PEFT methods including adapters and shows that simple prototypical classifier called SimpleCIL serve as a strong baseline. EASE [16] attaches adapters to each layer of FMs to create expandable subspaces, and during inference, it concatenates all feature representations from different sets of adapters to perform on a single classifier. MOS [17] adds replay generation for classifier alignment and an adapter merging over EASE to reduce mistakenly retrieving irrelevant modules during inference due to parameter drift.

3 Methodology

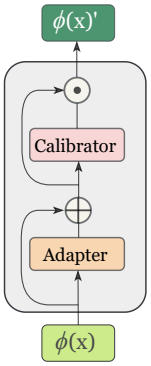


Figure 2: LuCA.

LuCA Module. It decouples feature transformation from discriminative feature enhancement with the dual adapter-calibrator architecture, allowing precise control over parameter updates. LuCA can process any intermediate representation $\mathbf{z} \in \mathbb{R}^d$ through two sequential operations:

$$L(\mathbf{z}) = C(A(\mathbf{z})), \tag{1}$$

where $A(\cdot)$ is a residual adapter that applies bottlenecked feature modulation to preserve original semantics via skip connections while learning task-specific offsets. The calibrator $C(\cdot)$ then reweights the adapter's output features through an attention-like gating, and refines more discriminative features with Eq. (2) where $\odot$ denotes the Hadamard product and σ indicates a sigmoid activation function.

$$C(\mathbf{z}) = \mathbf{z} \odot \sigma(\mathbf{z}V_{down})V_{up}, \quad V_{down} \in \mathbb{R}^{d\times r},\ V_{up} \in \mathbb{R}^{r\times d} \tag{2}$$

TOSCA: Specialization for CIL. In this work, to enable CIL with neuroscientific inspirations, we instantiate the LuCA module as TOSCA which is a strategic implementation operating exclusively on the final `[CLS]` token just before the classifier. Given an input $\mathbf{x}$, the frozen pre-trained backbone generates features $\phi(\mathbf{x})$ of the last `[CLS]` token, and TOSCA refines them through Eq. (3).

$$\phi(\mathbf{x})' = L(\phi(\mathbf{x})) = C(A(\phi(\mathbf{x}))) \tag{3}$$

The design of placing it at the last token is a deliberate architectural choice with three advantages: First, by adapting only the final `[CLS]` token, TOSCA preserves low- and mid-level features while flexibly adjusting high-level abstractions, balancing *stability* and *plasticity* similar to ventral stream representations modulated by prefrontal circuits [19–23]. Second, since the `[CLS]` token aggregates semantic information, it is the most effective site for task-specific refinement, unlike prompt-based methods that act indirectly through self-attention. Third, this design avoids modifying multiple layers and keeps parameter growth architecture-agnostic with a fixed $4dr$ footprint, compared to $N \times 2dr$ for layer-wise adapters, thus lowering training and inference cost.

Training Protocol. We completely freeze the parameters of FM and only train the TOSCA's parameters Θ together with the prototypical classifier $W^\top$. We utilize a new TOSCA module for each incremental stage b with the parameters Θ_b, which encodes task-specific information by optimizing an objective function that combines cross-entropy loss with ℓ_1-regularization as in Eq. (4), where λ controls the regularization strength. The ℓ_1 term induces to use of only a sparse subset of weights in the module, which encourages orthogonality, enabling each module to specialize on distinct feature dimensions, preventing interference [24, 25]. Please see Appendix A.1 for a detailed algorithm flow.

$$\min_{\Theta_b \cup W^\top} \sum_{(\mathbf{x},y)\in\mathcal{D}^b} \ell_{CE}\left(W^\top \phi(\mathbf{x})'_b, y\right) + \lambda\|\Theta_b\|_1, \quad \phi(\mathbf{x})'_b = L_b(\phi(\mathbf{x})) \tag{4}$$

Inference Protocol. We divide our inference protocol into a two-stage design for computational efficiency with minimal overhead. In the first stage, a single forward pass through the frozen backbone extracts a shared representation $\phi(\mathbf{x})$ for the input batch, up to the classifier. In the second stage, each TOSCA module independently processes this representation to produce task-specific predictions. This design avoids redundant computation and enables efficient reuse of features across all modules. Each transformed feature $\phi(\mathbf{x})'_b$ produces a task-specific probability distribution, and the module with the lowest output entropy is selected to make the final prediction over the union of all classes *without access to task labels* that can be formalized as in Eq. (5).

$$\hat{y} = \arg\min_{y\in\mathcal{Y}_b} H\left(\sum_{b=1}^{B} \pi_b p_b(y|\mathbf{x})\right), \quad p_b(y|\mathbf{x}) = softmax(W^\top \phi(\mathbf{x})'_b) \tag{5}$$

4 Experimental Setup

Datasets. Since FMs often exhibit substantial knowledge of upstream tasks, we adopt the evaluation framework proposed in [12–17] to assess their performance across a diverse set of benchmarks. These include CIFAR-100 [26], CUB-200 [27], ImageNet-R [28] ImageNet-A [29], OmniBenchmark [30], and VTAB [31]. These datasets encompass both standard CIL benchmarks and out-of-distribution datasets which exhibit significant domain shifts relative to the dataset used for pre-training, e.g. ImageNet [32]. Specifically, the datasets vary in scale: CIFAR-100 has 100 classes of natural images, each with 500 training images. CUB-200 dataset consists of images from 200 bird classes, with about 30 images per class for training. ImageNet-R includes 24000 for training images from 200 classes with abstract forms. ImageNet-A consists of 200 classes and 7500 training samples that are usually misclassified by ResNet models. Omnibenchmark with 300 classes and VTAB with 100 classes are designed to evaluate the generalization of visual representations.

Dataset Splits. Following [15–17], we use the notation 'B-m Inc-n' to represent the class splits, where m specifies the number of classes in the initial stage and n denotes the number of classes added at each incremental stage. Consistent with [33], we shuffle the class order randomly using the seed 1993 before splitting the data. To ensure a fair comparison, the training and testing sets for all methods are maintained the same.

Evaluation Metrics. We compare the methods based on the accuracy over all stages obtained after last stage and the accuracy across all stages. Formally, building on [33], we denote the Top-1 accuracy after the b-th stage as $\mathcal{A}_b$ and use $\mathcal{A}_B$ to represent the performance after the final stage. The average performance across all incremental stages then measured by $\bar{\mathcal{A}} = \frac{1}{B}\sum_{b=1}^{B}\mathcal{A}_b$.

Implementation Details. We conduct our experiments on an NVIDIA A100, and reproduce the compared methods using PyTorch [34] and Pilot [35]. For TOSCA, we train the model using the SGD optimizer with a batch size of 48 over 20 epochs. The learning rate starts at 0.025 and follows a cosine annealing schedule. The projection dimension r is set to 48 and the ℓ_1 contribution λ is set to 0.0005. We perform runs with five random seeds and report the mean and standard deviation.

5 Results

In this section, we compare our approach with state-of-the-art algorithms and share joint training performance as an upper bound against our method. Finally, we share a parameter and run-time analysis, along with an ablation study, to offer deeper insights.

State-of-the-Art Comparison. We compare TOSCA with leading state-of-the-art continual learning methods across six benchmark datasets and multiple foundation models. Table 1 summarizes the final-stage accuracy using ViT-B/16 pre-trained on IN21K for each method. TOSCA consistently achieves the highest performance across all six benchmarks, significantly surpassing existing approaches such as EASE and MOS. Notably, MOS relies on replay generation for classifier alignment and is therefore not strictly replay-free, whereas TOSCA achieves superior results without requiring a replay. To further examine learning dynamics, we report the incremental performance trends over successive training sessions in Figure 3, using ViT-B/16 pre-trained on IN1K. Across datasets, TOSCA outperforms the next-best methods by $4\% - 12\%$ on CIFAR100, ImageNet-R, and ImageNet-A, as indicated by the annotations at the end of learning sessions. These improvements are particularly pronounced on out-of-distribution datasets, highlighting TOSCA's robustness and capacity to generalize under distributional shifts.

Comparison to Joint Training. To provide a more complete picture, we also compare TOSCA to a traditional 100-epoch joint training, which represents an upper-bound for performance since it has access to all data at once. While joint training still holds the top spot, TOSCA demonstrates highly competitive performance despite only training a single, lightweight module for each new task. This is a crucial finding, as TOSCA delivers near-optimal performance while drastically reducing computational cost and training time. Unlike full joint training, it avoids the need to revisit all past data, making it a practical and scalable solution for real-world continual learning scenarios where memory and compute are limited.

Table 1: Average and last accuracy [%] with **ViT-B/16-IN21K** as the backbone. 'IN-R/A' stands for 'ImageNet-R/A,' and 'OmniBench' stands for 'OmniBenchmark.' We report the best performance in bold.

Method	CIFAR B0 Inc5		CUB B0 Inc10		IN-R B0 Inc20		IN-A B0 Inc20		OmniBench B0 Inc30		VTAB B0 Inc10	
	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$
Joint	–	96.21 ±1.0	–	92.62 ±1.1	–	81.92 ±1.4	–	67.97 ±1.9	–	85.44 ±1.2	–	94.96 ±1.2
Finetune	60.65 ±5.6	48.12 ±3.3	55.78 ±2.8	33.13 ±3.3	59.09 ±3.7	49.46 ±3.3	30.98 ±3.4	19.86 ±1.8	63.71 ±1.0	45.45 ±1.0	31.60 ±6.0	21.63 ±8.3
SimpleCIL	86.48 ±0.8	81.28 ±0.1	91.58 ±1.3	86.73 ±0.1	61.31 ±0.4	54.55 ±0.1	58.92 ±1.0	48.77 ±0.1	79.59 ±1.5	73.13 ±0.1	90.65 ±1.1	84.43 ±0.1
L2P	84.90 ±1.2	80.06 ±1.4	73.22 ±1.8	61.55 ±1.7	75.92 ±0.7	70.88 ±0.7	50.13 ±1.8	42.80 ±1.1	73.96 ±2.0	64.63 ±0.6	78.61 ±4.2	64.81 ±2.9
DualPrompt	85.61 ±1.3	79.92 ±0.4	81.36 ±1.8	70.51 ±1.1	71.48 ±0.5	66.09 ±1.3	51.57 ±0.4	40.56 ±1.6	75.58 ±1.4	66.46 ±0.8	86.86 ±2.8	75.86 ±3.7
CODAPrompt	87.64 ±0.4	81.46 ±0.3	77.65 ±1.0	68.44 ±1.0	76.25 ±0.3	71.39 ±0.3	58.82 ±0.78	47.18 ±0.9	73.73 ±0.5	69.46 ±0.7	87.60 ±0.5	86.71 ±0.8
APER-Adapter	89.57 ±0.9	84.91 ±0.2	91.62 ±1.2	86.72 ±0.2	74.81 ±0.8	66.97 ±0.8	59.57 ±1.6	49.46 ±0.4	80.48 ±1.2	74.04 ±0.3	90.59 ±1.0	84.28 ±0.2
EASE	90.79 ±0.8	85.97 ±0.6	92.51 ±1.3	86.49 ±1.2	80.35 ±1.0	75.74 ±0.8	64.00 ±1.5	54.99 ±1.0	81.11 ±0.8	74.16 ±2.0	90.26 ±3.6	82.07 ±3.0
MOS	93.45 ±0.9	90.04 ±0.6	93.42 ±1.2	90.07 ±0.9	82.26 ±1.0	77.62 ±0.9	63.57 ±2.0	54.60 ±0.8	84.73 ±1.1	79.97 ±0.9	92.75 ±1.0	92.74 ±0.9
TOSCA (ours)	**96.37** ±0.5	**95.64** ±0.8	**93.47** ±1.9	**91.09** ±1.8	**82.27** ±1.9	**79.28** ±1.9	**66.92** ±3.0	**65.37** ±2.9	**84.75** ±2.6	**82.35** ±1.0	**96.59** ±1.6	**93.87** ±2.0

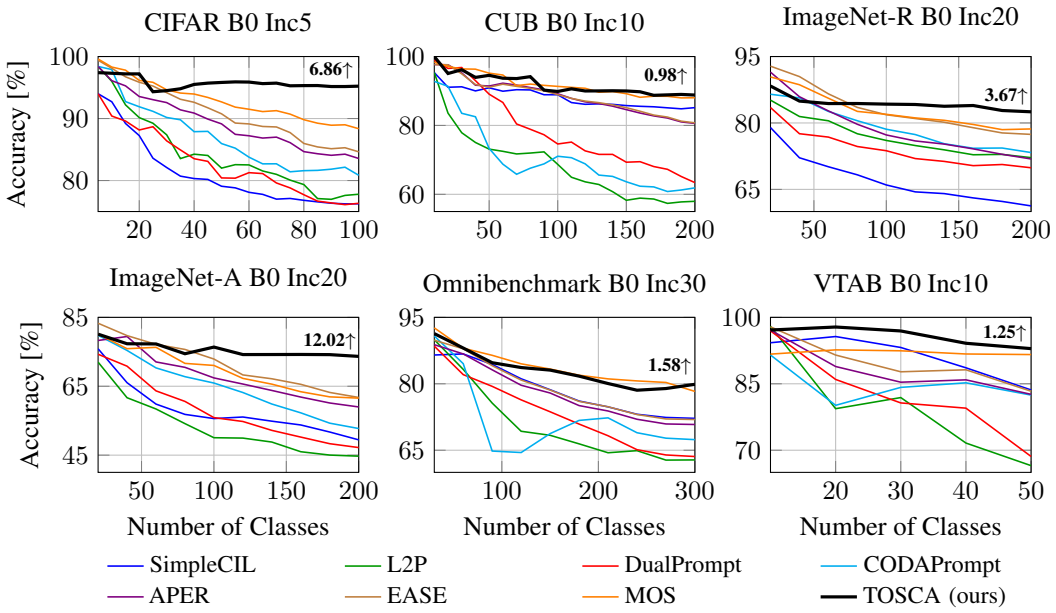
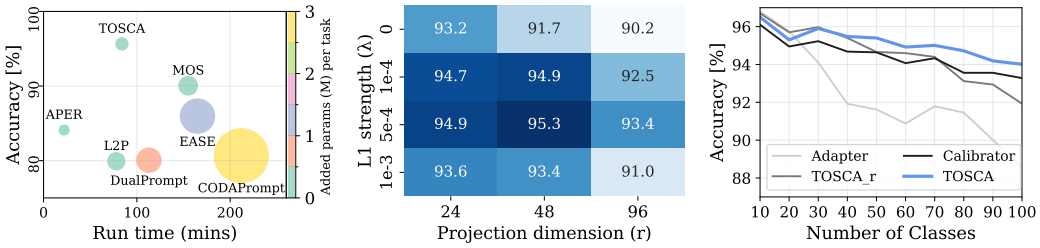


Figure 3: Performance curves of different methods with **ViT-B/16-IN1K**. We annotate the relative improvement of TOSCA above the runner-up method with numerical numbers at the last incremental stage.

Parameter and Run-Time Analysis. We further investigate FM-based CIL approaches in terms of accuracy, computational cost (overall run time), and parameter efficiency on CIFAR B0 Inc5 benchmark in Figure 4a. TOSCA achieves the top performance while maintaining a low computational cost and parameter overhead per task. In contrast, methods like CODA-Prompt and EASE require significantly more parameters and longer run times, making them less efficient. Notably, MOS also attains high accuracy, but it comes at a higher computational expense due to additional processes such as adapter merging and replay generation. Overall, TOSCA demonstrates its effectiveness in CIL with minimal parameter overhead and shorter run-time, striking a balance between efficiency and performance.

Ablation Study. We also perform an ablation study on CIFAR B0 Inc10, evaluating the incremental performance across different learning settings. First, we analyze the impact of ℓ_1-regularization strength (λ) and projection dimension (r) on performance, as shown in Figure 4b. Our findings indicate that moderate ℓ_1 regularization enhances accuracy, with performance peaking at $\lambda = 5e^{-4}$. This promotes the orthogonality among different modules, improving module selection during inference. However, excessive sparsity degrades performance by excessively constraining representations, thereby reducing expressiveness and learning capacity. Similarly, increasing the projection dimension (r) improves accuracy up to $r = 48$, beyond which performance deteriorates due to the larger bottleneck. Based on these observations, we identify the optimal configuration as $\lambda = 5e^{-4}$ and $r = 48$, achieving an accuracy of 95.3%. Additionally, we compare the performance of different components, alternative module designs and configurations against TOSCA in Figure 4c.



(a) Memory & computational cost (b) Hyperparameter analysis (c) Design and component ablation

Figure 4: Performance evaluation of TOSCA across different perspectives. (a) Memory & computational cost highlights TOSCA's efficiency, (b) Hyperparameter analysis illustrates effect of ℓ_1 strength (λ) and projection dimension (r) on accuracy, (c) Design and component ablation presents the impact of different components and flows on accuracy.

This includes a reversed variant, TOSCA_r, which integrates new information atop the calibrated pre-trained features, formulated as $\phi(\mathbf{x})' = A(C(\phi(\mathbf{x})))$. Our results highlight the crucial role of the calibrator while TOSCA surpasses other design variants by effectively harmonizing its two modules working together.

6 Conclusions

In this paper, we introduced LuCA, a new parameter-efficient fine-tuning module that combines a lightweight adapter with a calibrator to refine feature representations and facilitate the integration of new information. Building upon this, we proposed TOSCA, a neuro-inspired class-incremental learning framework based on foundation models, which leverages a single sparse LuCA module applied exclusively to the final `[CLS]` token before the classifier for each task. This targeted adaptation enables highly efficient, task-specific modulation while maintaining orthogonality between tasks, resulting in minimal memory and computational overhead. Extensive experiments across multiple benchmark datasets demonstrate that TOSCA consistently achieves state-of-the-art performance. It effectively balances the stability–plasticity trade-off, outperforming prior methods with substantially fewer additional parameters, while retaining a scalable and model-agnostic design suitable for advancing continual learning with foundation models.

Limitations and future works. Although TOSCA shows strong performance, it relies on pre-trained models, and its effectiveness depends on the generalizability of these models when adapting through a single token. Future work would explore extending TOSCA to additional modalities or multimodal models and diverse continual learning scenarios, including few-shot, blurry, and class-revisiting incremental settings.

Broader Impact

This paper aims to advance the field of machine learning, with a focus on replay-free class-incremental learning. By introducing a lightweight, trainable module that enables continual adaptation without the need to store past data, the proposed approach addresses key concerns related to privacy, memory efficiency, computational overhead, and scalability. These properties make it particularly suitable for deployment in privacy-sensitive or resource-constrained settings, such as personalized models in healthcare, adaptive vision systems for assistive technologies. While the method does not raise immediate ethical concerns, its application in high-stakes domains may require additional safeguards to ensure responsible use.

Acknowledgements

This work is supported by SYNERGIES, a project funded by the EU Horizon programme under GA No. 101146542; and Dutch e-infrastructure with the support of SURF Cooperative using GA no. EINF-10242; and a Turkish MoNE scholarship.

References

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

[2] Marc Masana, Xialei Liu, Bartłomiej Twardowski, Mikel Menta, Andrew D Bagdanov, and Joost Van De Weijer. Class-incremental learning: survey and performance evaluation on image classification. *TPAMI*, 2022.

[3] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021.

[4] Stéphane d'Ascoli, Hugo Touvron, Matthew L Leavitt, Ari S Morcos, Giulio Biroli, and Levent Sagun. Convit: Improving vision transformers with soft convolutional inductive biases. In *ICML*, 2021.

[5] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[7] Mark D McDonnell, Dong Gong, Amin Parvaneh, Ehsan Abbasnejad, and Anton van den Hengel. Ranpac: Random projections and pre-trained models for continual learning. In *NeurIPS*, 2024.

[8] Aristeidis Panos, Yuriko Kobe, Daniel Olmeda Reino, Rahaf Aljundi, and Richard E Turner. First session adaptation: A strong replay-free baseline for class-incremental learning. In *ICCV*, 2023.

[9] Gengwei Zhang, Liyuan Wang, Guoliang Kang, Ling Chen, and Yunchao Wei. Slca: Slow learner with classifier alignment for continual learning on a pre-trained model. In *ICCV*, 2023.

[10] Yabin Wang, Zhiwu Huang, and Xiaopeng Hong. S-prompts learning with pre-trained transformers: An occam's razor for domain incremental learning. 2022.

[11] Paul Janson, Wenxuan Zhang, Rahaf Aljundi, and Mohamed Elhoseiny. A simple baseline that questions the use of pretrained-models in continual learning. *NeurIPS Workshop on Distribution Shifts*, 2022.

[12] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *CVPR*, 2022.

[13] Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *ECCV*, 2022.

[14] James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, et al. Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In *CVPR*, 2023.

[15] Da-Wei Zhou, Zi-Wen Cai, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Revisiting class-incremental learning with pre-trained models: Generalizability and adaptivity are all you need. *IJCV*, 2024.

[16] Da-Wei Zhou, Hai-Long Sun, Han-Jia Ye, and De-Chuan Zhan. Expandable subspace ensemble for pre-trained model-based class-incremental learning. In *CVPR*, 2024.

[17] Hai-Long Sun, Da-Wei Zhou, Hanbin Zhao, Le Gan, De-Chuan Zhan, and Han-Jia Ye. Mos: Model surgery for pre-trained model-based class-incremental learning. In *AAAI*, 2024.

[18] Stephen T Grossberg. *Studies of mind and brain: Neural principles of learning, perception, development, cognition, and motor control*. Springer Science & Business Media, 2012.

[19] Irving Biederman and Peter C Gerhardstein. Recognizing depth-rotated objects: evidence and conditions for three-dimensional viewpoint invariance. *Journal of Experimental Psychology: Human perception and performance*, 19(6):1162, 1993.

[20] Irving Biederman. Recognition-by-components: a theory of human image understanding. *Psychological review*, 94(2):115, 1987.

[21] Ningyu Zhang and Ning-long Xu. Reshaping sensory representations by task-specific brain states: Toward cortical circuit mechanisms. *Current Opinion in Neurobiology*, 77:102628, 2022.

[22] Karl Friston and Stefan Kiebel. Cortical circuits for perceptual inference. *Neural Networks*, 22(8):1093–1104, 2009.

[23] Apoorva Bhandari, Haley Keglovits, and David Badre. Task structure tailors the geometry of neural representations in human lateral prefrontal cortex. *bioRxiv*, 2024.

[24] Kun-Peng Ning, Hai-Jian Ke, Yu-Yang Liu, Jia-Yu Yao, Yong-Hong Tian, and Li Yuan. Sparse orthogonal parameters tuning for continual learning. *arXiv:2411.02813*, 2024.

[25] Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing Huang. Orthogonal subspace learning for language model continual learning. *arXiv:2310.14152*, 2023.

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

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

[28] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *ICCV*.

[29] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *CVPR*, 2021.

[30] Yuanhan Zhang, Zhenfei Yin, Jing Shao, and Ziwei Liu. Benchmarking omni-vision representation through the lens of visual realms. In *ECCV*, 2022.

[31] Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, et al. A large-scale study of representation learning with the visual task adaptation benchmark. *arXiv:1910.04867*, 2019.

[32] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[33] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, 2017.

[34] Adam Paszke, Sam Gross, Francisco Massa, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*. 2019.

[35] Hai-Long Sun, Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Pilot: A pre-trained model-based continual learning toolbox. *arXiv:2309.07117*, 2023.

A Appendix

A.1 Compared Methods and TOSCA

Here, we provide an overview of the methods evaluated in the main paper. To ensure a fair and consistent basis for comparison, all methods utilize the same FM. This standardization allows us to isolate the contributions of each method's unique approach and compare their performance more accurately. Additionally, we present the pseudocode for TOSCA, providing a clear and detailed description of its working algorithm. This helps to better understand how TOSCA operates, offering insights into its efficiency and functionality within the context of class-incremental learning.

Joint: This method adheres to the traditional supervised batch learning paradigm, where all classes are presented simultaneously and trained over multiple epochs. It serves as the upper bound for class-incremental learning methods, as it does not experience forgetting.

Finetune: This method updates all parameters of the pretrained model when continually trained on new tasks. While it can achieve strong performance, it is susceptible to catastrophic forgetting, where previous knowledge is lost when learning new tasks.

SimpleCIL [15]: SimpleCIL uses the FM in its original form, combined with a prototypical classifier. It constructs a prototype for each class and utilizes a cosine classifier for classification, aiming for efficient task learning without additional adaptations.

L2P [12]: L2P integrates visual prompt tuning into class-incremental learning with a pre-trained Vision Transformer (ViT). The method places the prompt only in the initial embedding layer, ensuring that the prompt adjusts the features at the early stage of the model while maintaining the frozen structure of the rest of the pretrained model.

DualPrompt [13]: DualPrompt builds on L2P by introducing two types of prompts: general prompts (G-Prompt) and expert prompts (E-Prompt). The G-Prompts are applied to the earlier transformer blocks, allowing for broad task-specific adaptation. E-Prompts, on the other hand, are used in the latter blocks of the transformer, providing more specialized tuning for later stages of task processing. This separation allows for more efficient adaptation.

CODA-Prompt [14]: It addresses the challenges of selecting instance-specific prompts by introducing prompt reweighting. It enhances the selection process through an attention mechanism that dynamically weights prompts, improving task-specific performance.

APER [15]: This approach builds on SimpleCIL by introducing an adapter to each transformer layer, but only for the initial task. This adapter helps the pre-trained model to extract task-specific features during the first incremental phase, ensuring better adaptation to the new task while minimizing forgetting in subsequent tasks.

EASE [16]: This method adds adapters to each transformer layer for every task. This approach leads to good performance by concatenating the feature representations of multiple task-specific backbones, but it comes with an increase in model complexity due to the addition of task-specific adapters at every stage.

MOS [17]: It also trains adapters for each transformer layer for every task. However, MOS introduces the concept of adapter merging and replay generation for classifier correction. These processes increases computational complexity, particularly during training, as the model must handle the merging of multiple task-specific adapters with an increasing number of parameters.

Algorithm 1 TOSCA for FM-based CIL

Require: Incremental datasets: $\{\mathcal{D}^1, \mathcal{D}^2, \ldots, \mathcal{D}^B\}$, Pre-trained embedding: $\phi(\mathbf{x})$

1: **for** $b = 1, 2, \ldots, B$ **do**
2: Get the incremental training set $\mathcal{D}^b$
3: Initialize a new LuCA module L_b with parameters Θ_b on top of last `[CLS]` token
4: Optimize the parameters Θ_b of the module L_b and prototypical classifier $W^\top$ via Eq. (4)
5: Test the model with all classes seen so far via Eq. (5)
6: **end for**