IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, November 2026
Impact Factor: 8.2

Scalable Continual Learning with a Modular Dynamic Architecture and AutoML

Alhat Pratiksha Santosh[1] and Prof. Divya Ashok Rahane[2]
[1] Student, Department of Artificial Intelligence and Data Science
[2]Assistant Professor, Department of Artificial Intelligence and Data Science
Sahyadri Valley College of Engineering and Technology, Rajuri, India
[1] alhatpratiksha08@gmail.com, [2] divyarahane26@gmail.com

Abstract : *Machine learning systems deployed in dynamic real-world environments continuously encounter new tasks, classes, and shifting data distributions. Updating a conventional static neural network on sequential tasks degrades performance on previously learned tasks—a phenomenon termed catastrophic forgetting—while committing to an oversized fixed architecture wastes memory and compute. This paper presents a Scalable Continual Learning framework integrating an Opportunistic Modular Dynamic Architecture based on a Mixture of Experts (MoE) with resource-aware Automated Machine Learning (AutoML). The framework monitors incoming task streams, clusters similar classes via Gaussian Mixture Model (GMM) prototypes to instantiate compact expert modules, opportunistically fuses redundant experts using Singular Value Decomposition (SVD) weight similarity and norm alignment under memory limits, and employs an AutoML search controller to optimize preprocessing, routing policies, and hyperparameters across accuracy, retention, memory, and latency constraints. Experimental evaluations across CIFAR-100, CUB, and iNaturalist benchmarks achieve 76.03%, 87.53%, and 70.82% Final Average Accuracy while requiring only ~16% of the per-task training time of state-of-the-art replay baselines.*

Keywords: Continual Learning, Catastrophic Forgetting, Modular Neural Networks, Dynamic Architecture, AutoML, Neural Architecture Search, Hyperparameter Optimization.

I. INTRODUCTION

Machine learning has achieved strong performance across image recognition, natural language processing, recommendation, and industrial analytics [1], [2]. Most conventional training pipelines assume that all relevant training distributions and classes are available prior to deployment. In practical environments, however, data streams evolve continuously: new object categories appear, user behavior shifts, sensors are replaced, and input-output relationships drift [3]. Rebuilding monolithic models from scratch whenever changes occur is computationally prohibitive and storage-intensive.

Continual Learning (CL), also referred to as lifelong learning, aims to acquire knowledge sequentially from a stream of tasks while retaining previously learned information [1], [4]. Its central challenge is catastrophic forgetting, where gradient updates for a new task overwrite parameters critical to earlier tasks [5]. Existing regularization and replay strategies typically commit to a fixed, over-parameterized architecture at initialization, causing memory waste on early tasks and parameter saturation as task sequences grow [3], [6].

Conversely, adding a separate network for every incoming task causes unbounded parameter growth and high inference cost [7], [8]. A modular dynamic architecture offers a balanced solution by reusing, freezing, expanding, or merging modules according to task similarity and hardware limits [3]. Automated Machine Learning (AutoML) complements this design by automating preprocessing selection, model configuration, Neural Architecture Search (NAS), and Hyperparameter Optimization (HPO) [9], [10]. The objectives of this work are: (i) to design a modular continual task-learning framework; (ii) to define an opportunistic capacity controller for expert creation and SVD-based expert fusion;

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, November 2026
Impact Factor: 8.2

(iii) to integrate resource-aware AutoML search under accuracy, retention, memory, and latency constraints; and (iv) to validate scalability using standardized continual adaptation metrics.

II. RELATED WORK

A. Continual Learning: Regularization and Replay

Parisi et al. [1] and Wang et al. [2] review lifelong learning paradigms, categorizing methods into parameter regularization, rehearsal, and architectural expansion. Elastic Weight Consolidation (EWC) [5] and Learning without Forgetting (LwF) [11] penalize updates on weights deemed important for prior tasks. Rehearsal-based techniques such as Incremental Classifier and Representation Learning (iCaRL) [6], Bias Correction (BiC) [12], and Dark Experience Replay (DER++) [13] retain a bounded exemplar buffer of past samples and interleave them during training, though fixed-capacity backbones eventually saturate as class counts scale [3].

B. Dynamic and Modular Architectures

Progressive Neural Networks [7] allocate new columns per task with lateral transfer connections but require task identifiers at inference. Dynamically Expandable Networks (DEN) [8] and Dynamically Expandable Representations (DER) [14] selectively add neurons or feature extractors, often accumulating redundant parameters. Opportunistic Dynamic Architecture (ODA) [3] groups semantically similar classes into compact Mixture-of-Experts (MoE) modules and merges redundant experts when memory budgets are reached

C. Automated Machine Learning for Evolving Streams

AutoML reduces manual pipeline engineering by searching across data transformations, neural architectures, and hyperparameters via random search, Bayesian optimization, evolutionary algorithms, or successive halving [9], [10]. Applying AutoML to continual task streams requires multi-criteria evaluation that balances new-task accuracy against old-task retention, parameter growth, and update latency.

III. PROPOSED SYSTEM

A. System Overview and Architecture

The proposed system processes sequential task batches characterized by input features, label space, and arrival metadata. As illustrated in Fig. 1, the architecture integrates six modular stages: (1) Task/Data Stream Intake, (2) Preprocessing and Feature Representation via a Vision Transformer (ViT) backbone, (3) Task and Drift Monitoring using Gaussian Mixture Model (GMM) prototypes, (4) Resource-Aware AutoML Search Controller, (5) Module Router and Dynamic Capacity Controller, and (6) Updated Model Deployment with Bounded Replay Memory [3].

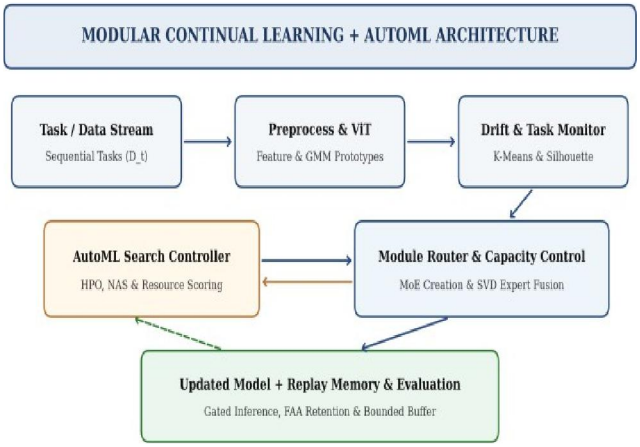


Fig. 1. Proposed modular dynamic continual learning and AutoML architecture combining task monitoring, expert routing, SVD fusion, and resource-aware search.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, November 2026
Impact Factor: 8.2

B. Module Library, Router, and Prototype Clustering

The module library stores reusable feature extractors, lightweight task experts, and classification mappers. For each incoming task T_t with dataset D_t = {(x_t^(j), y_t^(j)) | y_t^(j) in C_t}, the system fits a GMM per class to extract compact prototypes and applies K-means clustering (K in {2, 3}) guided by the Silhouette score [3]. A compact expert network E_i (one hidden layer of 128 neurons) is instantiated and trained independently per cluster. Once trained, experts are frozen and integrated via a Gating Network and Classification Mapper (CM) layer trained on replay exemplars and GMM pseudo-samples [3]:

$$y_hat = \Sigma_{i=1}^{M} g_i \cdot CM(z_i) \quad (1)$$

where z_i denotes logits from expert i, CM maps logits onto the global class space C_all, and g_i is the learned gating coefficient [3].

C. Dynamic Capacity Controller and SVD Expert Fusion

Expansion is triggered only when validation error or representation mismatch indicates insufficient capacity. Conversely, when active experts exceed hardware budget M_max, the controller identifies non-saturated expert pairs and computes weight similarity via Singular Value Decomposition (SVD), $W_{i,l} = U_{i,l} S_{i,l} V_{i,l}$ [3]:

$$d_{i,j} = \Sigma_{l} \| S_{i,l} - S_{j,l} \|_2 \quad (2)$$

The most similar experts are merged by averaging hidden-layer weights and concatenating classification heads after applying weight norm alignment γ = Mean(Norm_i) / Mean(Norm_j), followed by brief fine-tuning on buffered exemplars and GMM samples [3], [12].

IV. METHODOLOGY AND AUTOML OPTIMIZATION

A. Continual Learning Objective and AutoML Search Loop

Task streams are partitioned chronologically or sequentially into training, validation, and test splits without future data leakage. Training optimizes a composite continual learning objective balancing current-task loss, retention regularization, and architectural complexity:

$$L(t) = L_{current} + \lambda L_{retain} + \mu L_{complexity} \quad (3)$$

Concurrently, the AutoML controller searches across learning rates ([0.1, 0.01, 0.001, 0.0001]), batch sizes ([32, 64, 128]), replay buffer sizes, and expansion thresholds using Bayesian or grid search with early stopping to maximize a multi-criteria deployment score [3], [9]:

$$Score = \alpha \cdot Quality + \beta \cdot Retention - \gamma_m \cdot Memory - \delta \cdot Latency \quad (4)$$

Table I: Qualitative Comparison of Continual Learning and Automation Strategies

Approach	Primary Strength	Key Limitation
Replay Buffer	Revisits earlier task distributions	Memory and privacy overhead
Regularization	Low memory footprint	May constrain new-task plasticity
Dynamic Expansion	Adds capacity for novel tasks	Unbounded growth without fusion
AutoML Controller	Automates HPO and module routing	Search compute if unpruned

B. Training and Update Workflow

For each arriving task, the workflow executes nine deterministic steps: (1) validate data schema; (2) apply fitted preprocessing and ViT feature extraction; (3) compute GMM class prototypes and drift indicators; (4) cluster classes and retrieve matching modules; (5) run resource-bounded AutoML search (selecting optimal learning rate 0.001 and batch size 64); (6) reuse, expand, or SVD-merge experts; (7) train the gating and mapper layers with replay and GMM samples; (8) evaluate current and prior tasks; and (9) commit or roll back the model update based on retention thresholds [3].

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, November 2026
Impact Factor: 8.2

V. PERFORMANCE EVALUATION

A. Datasets and Evaluation Metrics

Experiments were conducted on three standard class-incremental benchmarks [3]: CIFAR-100 (100 classes across 10 tasks), Caltech-UCSD Birds CUB-200 (200 species across 10 tasks), and iNaturalist-2018 (1,011 long-tailed species across 20 tasks), with exemplar buffers of 10, 10, and 20 samples per class, respectively. Performance is measured via Final Average Accuracy (FAA = (1/T) Σ A_{T,i}), Final Forgetting (FF = (1/(T-1)) Σ (A_{i,i} - A_{T,i})), total parameter count (in millions), and per-task update latency (in seconds) [3].

Table II: Quantitative Benchmark Comparison (FAA % and Per-Task Training Time)

Technique	CIFAR-100	CUB-200	iNaturalist	Time / Task
Finetune	16.58%	22.34%	10.63%	~15 s
EWC [5]	18.39%	29.20%	20.15%	~42 s
ProgressNet [7]	14.92%	22.49%	11.06%	~68 s
iCaRL [6]	64.34%	79.76%	60.78%	105–562 s
DER++ [13]	71.01%	81.58%	58.63%	120–490 s
BiC [12]	73.64%	82.64%	68.28%	131–295 s
Proposed Modular + AutoML	76.03%	87.53%	70.82%	16–77 s

B. Results and Scalability Analysis

As shown in Table II, the proposed framework outperforms all regularization, replay, and static expansion baselines across every benchmark, achieving 76.03% FAA on CIFAR-100 (+2.39% over BiC), 87.53%
FAA on CUB (+4.89% over BiC), and 70.82% FAA on iNaturalist (+2.54% over BiC) [3]. Because existing experts remain frozen and only lightweight 100K-parameter modules and the gating network are updated, per-task training time drops to 17 s on CIFAR-100, 16 s on CUB, and 77 s on iNaturalist—approximately 16% of iCaRL's training time—while SVD fusion bounds total parameters to 0.76M on CIFAR-100 and 1.19M on CUB [3].

VI. APPLICATIONS AND FUTURE SCOPE

The framework supports personalized recommendation, evolving medical-image classification, industrial fault monitoring, autonomous driving perception, fraud detection, and multilingual domain adaptation where new tasks emerge continuously under strict memory budgets [1], [3].
Future extensions will explore task-boundary-free online continual learning with automated drift triggers, privacy-preserving federated expert sharing across edge devices, and neuron-activation saturation metrics to reduce post-merging expert forgetting [3].

VII. CONCLUSION

This paper presented a Scalable Continual Learning framework combining an Opportunistic Modular Dynamic Architecture with resource-aware AutoML. By clustering incoming task classes into compact Mixture-of-Experts modules, fusing redundant experts via SVD weight similarity and norm alignment, and optimizing pipeline configurations under memory and latency constraints, the system mitigates both catastrophic forgetting and unbounded parameter growth while achieving ~6x faster per-task adaptation.

ACKNOWLEDGMENT

S. S. Kharade thanks the Department of Artificial Intelligence and Data Science, Sahyadri Valley College of Engineering and Technology, Rajuri, for academic guidance, computing resources, and continuous encouragement.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, November 2026
Impact Factor: 8.2

REFERENCES

[1] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," Neural Netw., vol. 113, pp. 54–71, May 2019.
[2] L. Wang, X. Zhang, H. Su, and J. Zhu, "A comprehensive survey of continual learning: Theory, method and application," IEEE Trans. Pattern Anal. Mach. Intell., vol. 46, no. 8, pp. 5362–5383, Aug. 2024.
[3] F. Rahman, A. Rosales Sanabria, and J. Ye, "Opportunistic dynamic architecture for class-incremental learning," IEEE Access, vol. 13, pp. 59146–59156, Mar. 2025.
[4] D.-W. Zhou, Q. Wang, Z. Qi, H.-J. Ye, D. Zhan, and Z. Liu, "Class-incremental learning: A survey," IEEE Trans. Pattern Anal. Mach. Intell., vol. 46, no. 12, pp. 9851–9873, Jul. 2024.
[5] J. Kirkpatrick et al., "Overcoming catastrophic forgetting in neural networks," Proc. Nat. Acad. Sci. USA, vol. 114, no. 13, pp. 3521–3526, Mar. 2017.
[6] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "iCaRL: Incremental classifier and representation learning," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 5533–5542.
[7] A. A. Rusu et al., "Progressive neural networks," arXiv:1606.04671, 2016.
[8] J. Yoon, E. Yang, J. Lee, and S. J. Hwang, "Lifelong learning with dynamically expandable networks," in Proc. Int. Conf. Learn. Represent. (ICLR), 2018.
[9] X. He, K. Zhao, and X. Chu, "AutoML: A survey of the state-of-the-art," Knowl.-Based Syst., vol. 212, Jan. 2021, Art. no. 106622.
[10] I. Salehin et al., "AutoML: A systematic review on automated machine learning with neural architecture search," J. Inf. Intell., vol. 2, no. 1, pp. 52–81, 2024.
[11] Z. Li and D. Hoiem, "Learning without forgetting," IEEE Trans. Pattern Anal. Mach. Intell., vol. 40, no. 12, pp. 2935–2947, Dec. 2018.
[12] Y. Wu et al., "Large scale incremental learning," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 374–382.
[13] P. Buzzega, M. Boschini, A. Porrello, D. Abati, and S. Calderara, "Dark experience for general continual learning: A strong, simple baseline," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 33, 2020, pp. 15920–15930.
[14] S. Yan, J. Xie, and X. He, "DER: Dynamically expandable representation for class incremental learning," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 3013–3022.

AUTHOR PROFILE



Alhat Pratiksha Santosh is currently pursuing her undergraduate studies in the Department of Artificial Intelligence and Data Science at Sahyadri Valley College of Engineering and Technology, Rajuri, India. Her research interests include Continual Learning, Dynamic Neural Architectures, Automated Machine Learning (AutoML), and Deep Learning.



Divya Ashok Rahane is a Faculty member in the Department of Artificial Intelligence and Data Science at Sahyadri Valley College of Engineering and Technology, Rajuri, India. Her academic and research interests span Artificial Intelligence, Machine Learning, and Scalable Deep Learning Systems.