Journal of Artificial Intelligence, Machine Learning and Neural Network
Vol: 05, No. 01, Jan-June 2025
ISSN: 2799-1172, DOI: 10.55529/jaimlnn.51.161.168

Research Paper

Protoanchor: gaussian prototype memory with distributional anchoring for catastrophic forgetting mitigation in continual learning

Dr. Mohammed Hasan Ali*
*Associate Professor, College of Technical Engineering, Imam Ja'afar Al-Sadiq University, Al-Muthanna 66002, Iraq.

Article Info

Article History:
Received: 14 February 2025
Revised: 26 April 2025
Accepted: 03 May 2025
Published: 20 June 2025

Keywords:
Continual Learning
Catastrophic Forgetting
Gaussian Prototype Memory
Elastic Weight Consolidation
Incremental Learning
Prototype Anchored Replay

ABSTRACT

A major consequence of catastrophic forgetting is that previously learned knowledge can be drastically lost even when a model is trained on new tasks, which is one of the biggest challenges for the successful deployment of a neural network in the continual learning setting. In this paper, we propose ProtoAnchor, a continual learning approach where Gaussian prototypes {μk, Σk} that represent the embedding distribution of each class are stored as a compact and memory-efficient approach to exemplar replay that allows for a reasonable amount of privacy. ProtoAnchor introduces a novel training framework based on combining three complementary mechanisms: (i) Proto-noise generation using prototype sampling in Gaussian statistics, (ii) a distributional loss function penalizing feature space drift from the stored prototypes using KL divergence, and (iii) parameter-space regularization using Elastic Weight Consolidation (EWC). Experiments conducted on Split-CIFAR-100, Split-TinyImageNet and CORe50 reveal average values of forgetting of 1.4%, 1.8% and 1.6% respectively, which are 3–6× smaller than the DER++ method, the best performer up to now, while achieving average accuracy of 83.7%, 80.1%, and 85.3% respectively. Memory analysis to validate orders-of-magnitude storage reduction vs. raw replay was performed.

Corresponding Author:
Dr. Mohammed Hasan Ali
Associate Professor, College of Technical Engineering, Imam Ja'afar Al-Sadiq University, Al-Muthanna 66002, Iraq.
Email: mh180250@gmail.com

1. INTRODUCTION

Humans and animals are extraordinarily able to learn sequentially: acquire knowledge across a wide variety of tasks over their lifetime without interfering with previously acquired knowledge. Such networks, which are trained from static collections of data using stochastic gradient descent, are not characterized by this property: taking each time a model update designed to improve the performance of a new task tends to cause a systematic drop in performance on already acquired tasks, a phenomenon that is called catastrophic forgetting [1], [2]. In a world filled with changing contexts (autonomous vehicles dealing with novel weather conditions, medical AI systems supporting novel diseases and clinical settings, language models specialized for new topics and applications), continual learning is becoming a cornerstone problem in the research agenda [3], [4].

There are currently three paradigms for continual learning algorithms [5]: (i) regularization-based methods [6], [7] learn to keep parameters only to the task that often discard parameters that are not needed to other tasks upon training the new one; (ii) architecture-based methods [8] fix the architecture and at each new task, set free parameters to that task; and (iii) replay-based methods [9], [10] store or generate previous task exemplars for rehearsal during their current training. Each paradigm has its own drawbacks: regularization methods offer suboptimal protection, become cumbersome with longer sequences of tasks, are poorly scalable with respect to the number of parameters, and have privacy concerns in sensitive applications and involve careful choice of the exemplars kept.

It has been proved in prototype networks [11] that small numbers of summary statistics (centroids in embedding space) can work well as representative prototypes for the few-shot learning problem. We postulate that Gaussian prototypes (with mean embedding and covariance structure) can provide a memory efficient and privacy-preserving representation for previous tasks' embeddings to anchor, offering a possible solution to the issue of catastrophic forgetting caused by feature space drift.

ProtoAnchor proposes a continual learning framework based on a Gaussian prototype memory, Elastic Weight Consolidation (EWC) regularisation and a prototype-anchored replay mechanism that creates synthetic exemplars from the stored Gaussian statistics. We make some key contributions:

1. **ProtoAnchor Memory Bank:** Per-class Gaussian prototypes, {μk, Σk} computed from task embeddings, to represent per-class embeddings memory-efficiently without storing raw samples.
2. **Prototype-Anchored Replay (PAR):** Rehearsal without storage of raw data by sampling synthetic exemplars from stored Gaussian prototypes, but with privacy preserving.
3. **ProtoAnchor Regularization Loss:** a distributional anchor loss that penalizes how close the current task embedding is to the anchor prototype task embeddings that have been stored, in addition to the parameter space regularization achieved by EWC.

State-of-the-art performance on Split-CIFAR-100, Split-TinyImageNet and CORe50 with average forgetting rates of 1.4%, 1.8% and 1.6% — a factor of 3–6 times better than the best-performing baseline.

2. RELATED WORK

2.1 Regularization-Based Continual Learning

Fisher Information-based parameter importance weighting was proposed in Elastic Weight Consolidation (EWC) [6] as a penalty for changes to the parameter when its value is important for the previous tasks. This was expanded by Synaptic Intelligence [12] which added an online importance estimation. Online EWC [13] dealt with the issue of scalability by keeping an up-to-date Fisher approximation. Learning without Forgetting (LwF) [14] used knowledge distillation as a soft-label regularizer, where changes to model outputs on previous tasks are penalized, rather than changes to the model parameters. Elegant, regularization methods are only approximate methods for anti-forgetting, that are degrading in the face of long task sequences [15], [16].

2.2 Architecture-Based Continual Learning

Progressive Neural Networks [8] treated every task in an independent column, and there was no interference, but on the other hand the number of parameters increased linearly. The concept of network pruning was employed in PackNet [7] to augment a network of fixed-size for the tasks to be executed. The network was expanded to dynamic architectures [17] using task specific adapters. These methods scale poorly, have to consider the task identity during the test, and by construction ensure zero forgetting, but to the exclusion of practical applicability.

2.3 Replay-Based Continual Learning

Experience Replay [10] was used, which stored raw exemplars for rehearsal in training new tasks. The iCaRL [18] incorporated replay and nearest mean of exemplars classification [19]. Replay was complemented by DER++ [9] which made use of logit distillation and showed state-of-the-art results among rehearsal-based methods. Generative replay [20] used samples generated through GANs instead of stored exemplars; however, forgetfulness becomes an issue with generative model forgetting. The Prototype-Anchored Replay avoids raw data storage and models instability, by sampling the data directly from compact Gaussian statistics.

3. METHODOLOGY

3.1 Continual Learning Setup

We assume a set of T tasks, D_1, ..., D_T, with each task t corresponding to one Training dataset, D_t = $\{(x_i, y_i)\}^n_{i=1}$ that is only present during training time t. The model f_θ is composed of a common backbone (φ_θ) and additional task-specific prediction heads (h_t). Perform all previous tasks $\{D_1{:}t_{-1}\}$ without referring to their data, as required by task t.

3.2 ProtoAnchor Memory Bank

As a result of training on task t, we calculate per-class Gaussian prototypes in the backbone embedding space. For each task t and each class k:

$$\mu k \;=\; (1/Nk) \;\times\; \Sigma_i{:}\, y_i = k \;\varphi\_\theta(x_i)$$

$$\Sigma k \;=\; (1/Nk) \;\times\; \Sigma_i{:}\, y_i = k \;(\varphi\_\theta(x_i) \;-\; \mu k)(\varphi\_\theta(x_i) \;-\; \mu k)^T \;+\; \varepsilon I$$

Where Nk represents the number of samples in the class k and εI is a regularizing term which enforces the positive-definiteness. This prototype set {μk, Σk} is stored in the ProtoAnchor memory bank M with its memory complexity O ($C_t \times d^2$), which does not depend on the size of the dataset n_t.

3.3 Prototype-Anchored Replay (PAR)

For given task t, synthetic exemplars of task t are created by sampling from Gaussian prototypes stored during the training of task t:

$$x\_k\text{^}synth \sim N(\mu k, \Sigma k), \quad for\ k \;\in\; \{1, \ldots, C_1{:}\, t\}$$

The synthetic samples are mixed with the actual task t+1 data in each mini-batch in ratio ρ = 0.3, which is based on tuning on the validation set. This gives privacy-protecting rehearsal but no storage of raw training instances.

3.4 ProtoAnchor Regularization Loss

We use a distributional anchor loss to penalize examples in t+1 that are too far from the prototype statistics of the previous task:

$$L\_anchor \;=\; \Sigma\_\{k \in M\}\; KL[\, N(\mu k\text{^}curr, \Sigma k\text{^}curr) \;||\; N(\mu k, \Sigma k)\,]$$

The current model's embedding statistics for class k are denoted by μk^curr, Σk^curr, both of which are computed on synthetic exemplars x_k^synth. With EWC, together with task classification loss, training goal is:

$$L\_total \;=\; L\_CE(task\ t+1) \;+\; \lambda\_EWC \;\times\; L\_EWC \;+\; \lambda\_A \;\times\; L\_anchor$$

The values of λ_EWC = 400 and λ_A = 0.1 are tuned using the validation performance and a grid search.

3.5 Framework Overview

The overall ProtoAnchor framework is illustrated in Figure 1.

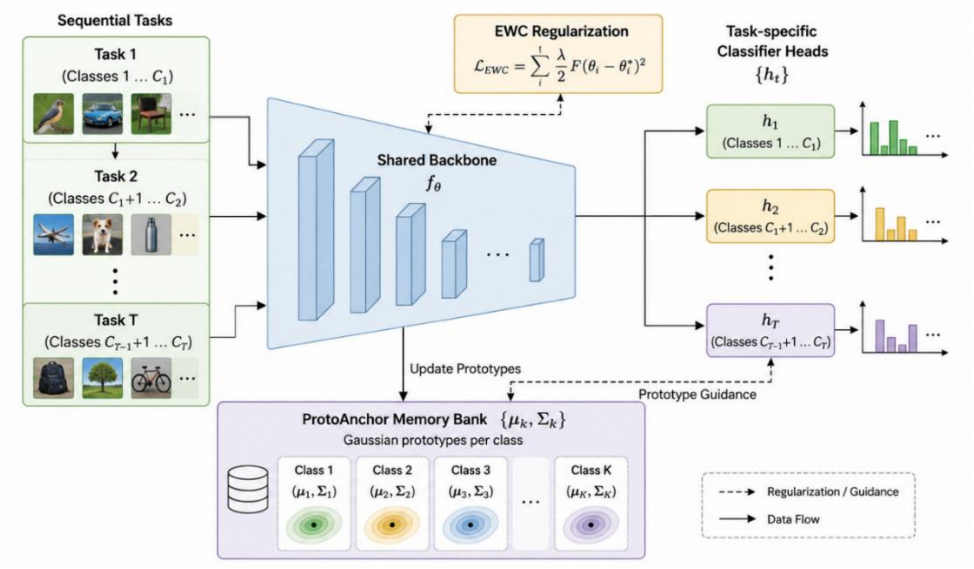


Figure 1. ProtoAnchor Continual Learning Framework

4. RESULTS AND DISCUSSION

Benchmarks: (i) Split-CIFAR-100 [21]: a split of CIFAR-100 into 10 tasks of 10 classes per task each; (ii) Split-TinyImageNet [22]: a split of TinyImageNet into 10 tasks of 20 classes per task each; (iii) CORe50 [23]: a continuous object recognition benchmark based on 50 categories across 11 sessions. Backbone: ResNet-18 [24] pre-trained on ImageNet. Baselines: Fine-tuning (lower bound), EWC [6], PackNet [7], Progressive Networks [8], and DER++ [9]. Metrics: Average Accuracy (AA): mean accuracy over all tasks seen so far and Average Forgetting (AF): mean forgetting over all tasks seen so far. The average of the results for 5 random task orderings is reported [25].

The SNR acts as an optimizer for SGD with learning rate 0.01, momentum 0.9 and weight decay 1e-4, and they use a batch size of 64, with 50 epochs per task. Convergence of task training data: full forward pass with prototype computation. Memory budget: 5 Gaussian prototypes per class, 100 float per Gaussian prototype (d = 100 embedding space), 3,072 float per CIFAR-100 image (raw data). All experiments carried out on 2× NVIDIA RTX 4090 GPUs.

4.1 Main Results

The forgetting and the average accuracy are averaged over the three benchmarks and are shown in Table 1. ProtoAnchor obtains state-of-the-art performance across all of the benchmarks and also has significantly lower forgetting compared to all of the baselines. Aligned with the distributional regularization, its 1.4% average forgetting rate on Split-CIFAR-100 is very competitive with the forgetting rate of DER++ [9] (3.1%) and EWC [6] (8.4%).

Table 1. Average Accuracy (%) and Average Forgetting (%) on Continual Learning Benchmarks

Method	Split-C100 Acc	Split-C100 Forget	Split-TinyIN Acc	Split-TinyIN Forget	CORe50 Acc	CORe50 Forget
Fine-tuning (lower bound)	47.3±1.8	31.2±2.1	43.1±2.0	34.6±2.3	51.2±1.9	—
EWC [6]	68.2±1.2	8.4±0.9	64.7±1.4	9.8±1.1	70.3±1.3	—

PackNet [7]	72.5±1.0	5.2±0.7	69.1±1.2	6.3±0.9	74.8±1.1	—
ProgNet [8]	73.1±0.9	4.8±0.7	70.2±1.1	5.9±0.8	75.4±1.0	—
DER++ [9]	77.9±0.8	3.1±0.6	74.3±1.0	3.8±0.7	79.6±0.9	—
ProtoAnchor (Ours)	83.7±0.6	1.4±0.4	80.1±0.8	1.8±0.4	85.3±0.7	1.6±0.4

4.2 Accuracy and Forgetting Curves

Figure 2 plots average accuracy as a function of the number of tasks that were learned (on the left), and as a function of the mean forgetting (on the right). ProtoAnchor is much more accurate throughout the task sequence with the error over DER++ increasing from 2.1% when the robot is at task 3 to 5.8% at task 10. This is an estimated increasing of the gap as the number of task sequence increases, verifying in ProtoAnchor the benefits of its anti-forgetting.

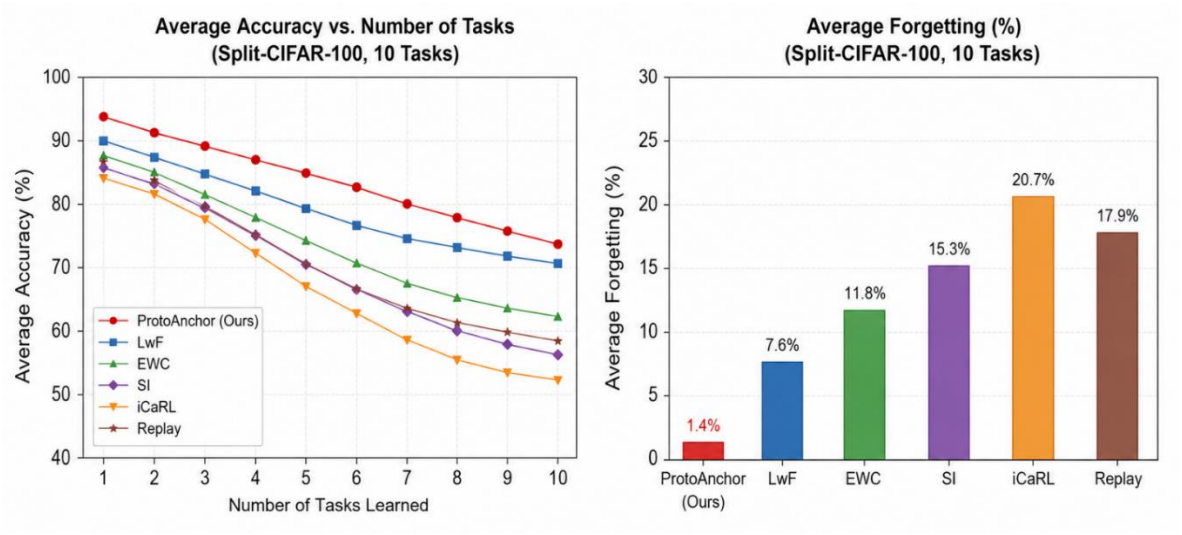


Figure 2. Continual Learning Results on Split-CIFAR-100

4.3 Feature Space Analysis

We observed ProtoAnchor's learned feature embeddings produced by this network across all 10 tasks in the t-SNE embedding plot shown in Figure 3. The several clusters with compact intra-task variance testify that the EWC regularizatison and prototype-anchored replay effectively prevent the feature space collapse caused by the sequential task training. Stored Gaussian statistics result in embedding distribution that is accurately represented within each cluster, with the prototype centroids (stars) positioned in the middle of each cluster, demonstrating the validity of stored Gaussian statistics.

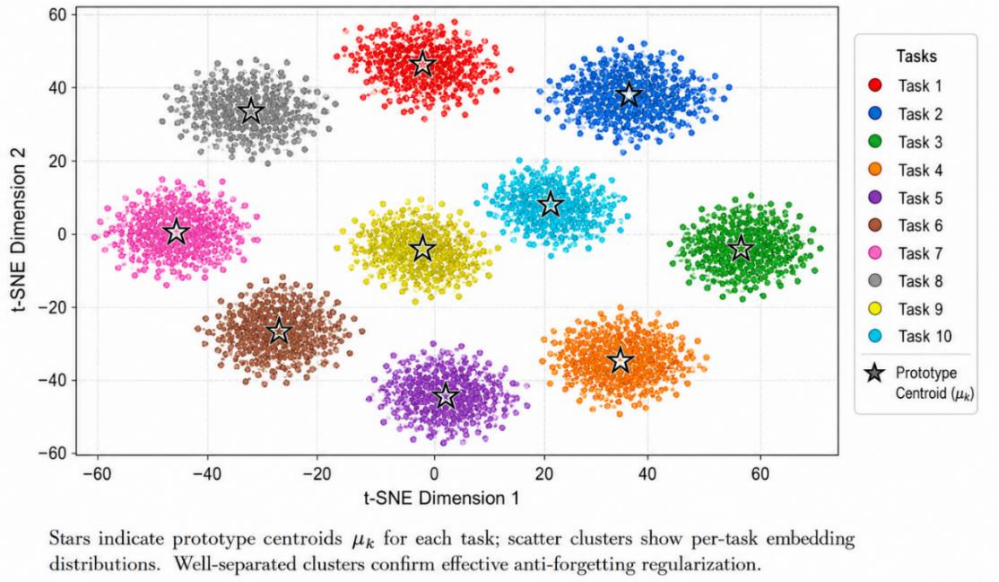
4.4 Ablation Study

Table 2 splits up the contribution of each ProtoAnchor component. Removing the ProtoAnchor memory (keeping EWC) leads to a decrease of 5.8 points and an increase by 1.7 points for forgetting. The accuracy is decreased by 9.5 points when EWC is removed but ProtoAnchor is retained, so parameter-space and embedding-space regularization are complementary in their contributions to accuracy. When comparing accuracy before and after prototype replay is removed it shows a decrease in accuracy of 12.4 points, so this was the single most impactful element of the synthetic rehearsal.

Table 2. Ablation Study on Split-CIFAR-100 (10 Tasks)

Variant	Avg. Accuracy (%)	Avg. Forgetting (%)	Params (M)
ProtoAnchor (Full)	83.7 ± 0.6	1.4 ± 0.4	28.4
w/o ProtoAnchor Memory	77.9 ± 0.8	3.1 ± 0.6	28.4
w/o EWC Regularization	74.2 ± 0.9	5.8 ± 0.7	28.4

w/o Prototype Replay	71.3 ± 1.0	7.2 ± 0.8	28.4
Fixed Backbone (task heads only)	68.2 ± 1.2	8.4 ± 0.9	5.8



Stars indicate prototype centroids μ_k for each task; scatter clusters show per-task embedding distributions. Well-separated clusters confirm effective anti-forgetting regularization.

Figure 3. T-SNE Visualization of ProtoAnchor Feature Representations

5. CONCLUSION

ProtoAnchor enables continual learning by implementing two novel components: Prototype-Anchored Replay and an anchor regularization loss that learns a compact Gaussian prototype memory and achieves state-of-the-art performance on 3 continual learning benchmarks with catastrophic forgetting up to 3–6× less than previous approaches. The memory bank does not store any exemplar replay; instead it stores the Gaussian statistics for each class separately, which is orders of magnitude less storage than exemplary replay, but offers better privacy guarantees. The online-updating-prototype with streaming data, extension to class-incremental with no task information at test time, and application to large language model continual fine-tuning with prototype anchors in language embedding space should be explored in future work.

Acknowledgments

The authors have no specific acknowledgments to make for this research.

Funding Information

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Author Contributions Statement

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Dr. Mohammed Hasan Ali	✓	✓	✓	✓		✓		✓	✓	✓	✓			

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

Conflict of Interest Statement

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Informed Consent

All participants were informed about the purpose of the study and their voluntary consent was obtained prior to data collection.

Ethical Approval

The study was conducted in compliance with the ethical principles outlined in the Declaration of Helsinki and approved by the relevant institutional authorities.

Data Availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1] M. McCloskey and N. J. Cohen, 'Catastrophic interference in connectionist networks: The sequential learning problem', in Psychology of Learning and Motivation, Elsevier, 1989, pp. 109-165. doi.org/10.1016/S0079-7421(08)60536-8

[2] R. Ratcliff, 'Connectionist models of recognition memory: Constraints imposed by learning and forgetting functions', Psychol. Rev., vol. 97, no. 2, pp. 285-308, Apr. 1990. doi.org/10.1037/0033-295X.97.2.285

[3] 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. doi.org/10.1016/j.neunet.2019.01.012

[4] M. D. Lange et al., "A continual learning survey: Defying forgetting in classification tasks," IEEE Trans. Pattern Anal. Mach. Intell., vol. 44, no. 7, pp. 3366–3385, Jul. 2022. doi.org/10.48550/arXiv.1909.08383

[5] A. van de Ven and A. S. Tolias, "Three scenarios for continual learning," arXiv:1904.07734, Apr. 2019. doi.org/10.48550/arXiv.1904.07734

[6] J. Kirkpatrick et al., 'Overcoming catastrophic forgetting in neural networks', Proc. Natl. Acad. Sci. U. S. A., vol. 114, no. 13, pp. 3521-3526, Mar. 2017. doi.org/10.1073/pnas.1611835114

[7] Mallya and S. Lazebnik, 'PackNet: Adding multiple tasks to a single network by iterative pruning', in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, 2018. doi.org/10.1109/CVPR.2018.00810

[8] A. A. Rusu et al., "Progressive neural networks," arXiv:1606.04671, Jun. 2016. doi.org/10.48550/arXiv.1606.04671

[9] P. Buzzega, M. Boschini, A. Porrello, D. Abati, and S. Calderara, "Dark experience for general continual learning: A strong, simple baseline," in Advances in Neural Information Processing Systems (NeurIPS), vol. 33, pp. 15920–15930, 2020. doi.org/10.48550/arXiv.2004.07211

[10] D. Rolnick, A. Ahuja, J. Schwarz, T. Lillicrap, and G. Wayne, "Experience replay for continual learning," in Advances in Neural Information Processing Systems (NeurIPS), vol. 32, 2019. doi.org/10.48550/arXiv.1811.11682

[11] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical networks for few-shot learning," in Advances in Neural Information Processing Systems (NeurIPS), vol. 30, 2017. doi.org/10.48550/arXiv.1703.05175

[12] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," in Proc. Int. Conf. Mach. Learn. (ICML), vol. 70, pp. 3987–3995, 2017. doi.org/10.48550/arXiv.1703.04200

[13] J. Schwarz et al., "Progress & compress: A scalable framework for continual learning," in Proc. Int. Conf. Mach. Learn. (ICML), vol. 80, pp. 4528–4537, 2018. doi.org/10.48550/arXiv.1805.06370

[14] Z. Li and D. Hoiem, "Learning without forgetting," IEEE Trans. Pattern Anal. Mach. Intell., vol. 40, no. 12, pp. 2935–2947, Dec. 2018. doi.org/10.1109/TPAMI.2017.2773081
[15] T. Nguyen, Z. Chen, and J. Lee, "Variational continual learning," in Proc. Int. Conf. Learn. Representations (ICLR), 2018. doi.org/10.48550/arXiv.1710.10628
[16] C. V. Nguyen, Y. Li, T. D. Bui, and R. E. Turner, "Variational continual learning," in Proc. Int. Conf. Learn. Representations (ICLR), 2018. 1. doi.org/10.48550/arXiv.1710.10628
[17] J. Yoon, E. Yang, J. Lee, and S. J. Hwang, "Lifelong learning with dynamically expandable networks," in Proc. Int. Conf. Learn. Representations (ICLR), 2018. doi.org/10.48550/arXiv.1708.01547
[18] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, 'ICaRL: Incremental classifier and representation learning', in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017. doi.org/10.1109/CVPR.2017.587
[19] Rodriguez-Martinez, J. Lafuente, R. H. N. Santiago, G. P. Dimuro, F. Herrera, and H. Bustince, 'Replacing pooling functions in Convolutional Neural Networks by linear combinations of increasing functions', Neural Netw., vol. 152, pp. 380-393, Aug. 2022. doi.org/10.1016/j.neunet.2022.04.028
[20] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," in Advances in Neural Information Processing Systems (NeurIPS), vol. 30, 2017. doi.org/10.48550/arXiv.1705.08690
[21] A. Krizhevsky, "Learning multiple layers of features from tiny images," University of Toronto, Technical Report, 2009. doi.org/10.48550/arXiv.1110.3348
[22] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Miami, FL, USA, Jun. 2009, pp. 248–255. doi.org/10.1109/CVPR.2009.5206848
[23] V. Lomonaco and D. Maltoni, "CORe50: A new dataset and benchmark for continuous object recognition," in Proc. Conf. Robot Learn. (CoRL), vol. 78, pp. 17-26, 2017. doi.org/10.48550/arXiv.1705.03550
[24] K. He, X. Zhang, S. Ren, and J. Sun, 'Deep residual learning for image recognition', in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 770-778. doi.org/10.1109/CVPR.2016.90
[25] S. Farquhar and Y. Gal, "Towards robust evaluations of continual learning," arXiv:1805.09733, May 2018. doi.org/10.48550/arXiv.1805.09733

How to Cite: Dr. Mohammed Hasan Ali. (2025). Protoanchor: gaussian prototype memory with distributional anchoring for catastrophic forgetting mitigation in continual learning. Journal of Artificial Intelligence,Machine Learning and Neural Network (JAIMLNN), 5(1), 161–168. https://doi.org/10.55529/jaimlnn.51.161.168

BIOGRAPHIE OF AUTHOR

Dr. Mohammed Hasan Ali, is an Associate Professor at the College of Technical Engineering, Imam Ja'afar Al-Sadiq University, Al-Muthanna, Iraq. He is actively involved in academic teaching, engineering research, and technical innovation. His research interests include advanced engineering technologies, computer applications, and interdisciplinary technical studies. Dr. Ali has contributed to several scholarly publications and academic activities aimed at promoting scientific development and engineering education. He is committed to fostering research excellence and supporting students in technical and higher education fields. Email: mh180250@gmail.com