EDUSCHOOL INTERNATIONAL JOURNAL OF DATA SCIENCE AND MACHINE LEARNING (EIJDSML)

(Open Access, Double-Blind Peer Reviewed Journal)

ISSN Online: ISSN Print

Mitigating Catastrophic Forgetting via Sparse Replay in Deep Networks

Bini P B

Assistant Professor, Department of Computer Science, CCSIT Dr. John Matthai Center, Thrissur, India

Article information

Received: 2nd January 2026
Received in revised form: 4th February 2026
Accepted: 5th March 2026
Available online: 18th April 2026

Volume: 1
Issue: 1
DOI: https://doi.org/10.5281/zenodo.19625725

Abstract

Deep neural networks trained sequentially on multiple tasks suffer from catastrophic forgetting—the abrupt loss of previously acquired knowledge upon learning new information. Experience replay, which stores and revisits past training examples, is among the most effective mitigation strategies, yet conventional replay buffers impose substantial memory overhead that limits scalability. This paper presents a comprehensive survey and empirical analysis of sparse replay buffer methods that achieve competitive or superior anti-forgetting performance while maintaining memory budgets orders of magnitude smaller than full experience replay. We formalize the continual learning problem and taxonomy of approaches, then focus on sparse replay strategies including coreset selection, gradient-based sample prioritization, compressed exemplar storage, generative replay with distillation, and hybrid regularization-replay methods. Through systematic experiments on Split CIFAR-100, Split ImageNet, Permuted MNIST, and Sequential Omniglot benchmarks, we demonstrate that sparse replay with as few as 1–5 exemplars per class achieves 85–95% of full replay performance. We analyze the interplay between buffer size, selection strategy, and task similarity, providing practical guidelines for deploying continual learning systems under memory constraints.

Keywords:- Continual Learning, Catastrophic Forgetting, Experience Replay, Sparse Replay Buffers, Coreset Selection, Knowledge Distillation, Lifelong Learning.

I. INTRODUCTION

Biological neural systems continuously learn from non-stationary data streams, integrating new knowledge while retaining prior learning throughout an organism's lifetime. In contrast, artificial neural networks trained with stochastic gradient descent exhibit catastrophic forgetting [1], [2]: when trained sequentially on tasks T_1, T_2, ..., T_n, performance on earlier tasks degrades severely as parameters are overwritten to accommodate new objectives. This fundamental limitation poses a critical barrier to deploying deep learning in real-world settings where data arrives continuously and retraining from scratch is impractical [3].

The continual learning community has developed three principal families of approaches to address catastrophic forgetting [4]. Regularization-based methods (EWC [5], SI [6], LwF [7]) constrain parameter updates to preserve important weights for prior tasks. Architecture-based methods (Progressive Neural Networks [8], PackNet [9]) allocate dedicated parameters for each task, preventing interference by construction. Replay-based methods (Experience Replay [10], A-GEM [11]) store and revisit examples from prior tasks during current training, directly combating the distribution shift that causes forgetting.

Among these families, replay-based methods consistently achieve the strongest empirical performance [4], [12]. However, standard experience replay requires storing a substantial buffer of past examples, creating tension between anti-forgetting effectiveness and memory efficiency. Storing thousands of examples per task may be feasible for small benchmarks but becomes prohibitive for high-resolution images, medical records, or privacy-sensitive data where exemplar storage is limited by regulation [13].

This paper focuses on sparse replay methods that maximize anti-forgetting performance under tight memory budgets. We provide a unified framework for analyzing replay buffer strategies, covering exemplar selection criteria, compressed storage techniques, and hybrid approaches that combine sparse replay with complementary anti-forgetting mechanisms. Table I summarizes the methods reviewed in this paper.

Table 1. Summary of Sparse Replay Methods for Continual Learning

Method	Category	Buffer Size	Selection Strategy	Year
ER [10]	Random replay	Fixed	Random uniform	2019
GDumb [14]	Greedy replay	Fixed	Greedy class-balanced	2020
GSS [15]	Gradient replay	Fixed	Gradient diversity	2019
HAL [16]	Anchored replay	Fixed	Hindsight anchors	2021
MIR [17]	Interference replay	Fixed	Max. interference	2019
DER++ [18]	Distillation replay	Fixed	Random + logits	2022
REMIND [19]	Compressed replay	Fixed	Quantized features	2021
ACE [20]	Asymmetric replay	Adaptive	Asymmetric cross-entropy	2022

II. PROBLEM FORMULATION AND EVALUATION FRAMEWORK

A. Continual Learning Settings

We consider three standard continual learning settings of increasing difficulty [4]. In Task-Incremental Learning (Task-IL), the model receives a task identifier at both training and test time, enabling task-specific output heads. In Class-Incremental Learning (Class-IL), the model must classify among all classes seen so far without a task identifier—the most challenging and practically relevant setting. In Domain-Incremental Learning (Domain-IL), the task structure is fixed but the input distribution shifts over time.

Formally, let $D = \{D_1, D_2, ..., D_t\}$ be a sequence of T task datasets arriving sequentially. Each $D_t = \{(x_i, y_i)\}_{i=1}^{N_t}$ contains N_t examples. The model f_θ with parameters θ processes tasks sequentially: during training on D_t, access to $D_1, ..., D_{t-1}$ is restricted to the replay buffer $M \subset \cup_{s=1}^{t-1} D_s$ with $|M| \leq B$, where B is the memory budget.

B. Evaluation Metrics

We adopt standard continual learning metrics [4], [12]. Average Accuracy (AA) after learning task T is $A_T = \frac{1}{T}\sum_{t=1}^{T} a_{t,t}$, where $a_{t,t}$ is the accuracy on task t after training on all T tasks. Average Forgetting (AF) measures the mean accuracy decline:

$$A_{FT} = \frac{1}{T-1}\sum_{t=1}^{T-1} \max_{s\in\{t,\ldots,T\}}\left(a_{t,s} - a_{t,t}\right) \quad (1)$$

Forward Transfer (FT) measures the influence of prior learning on new task performance. The Learning Curve Area (LCA) captures the full trajectory of knowledge acquisition and retention.

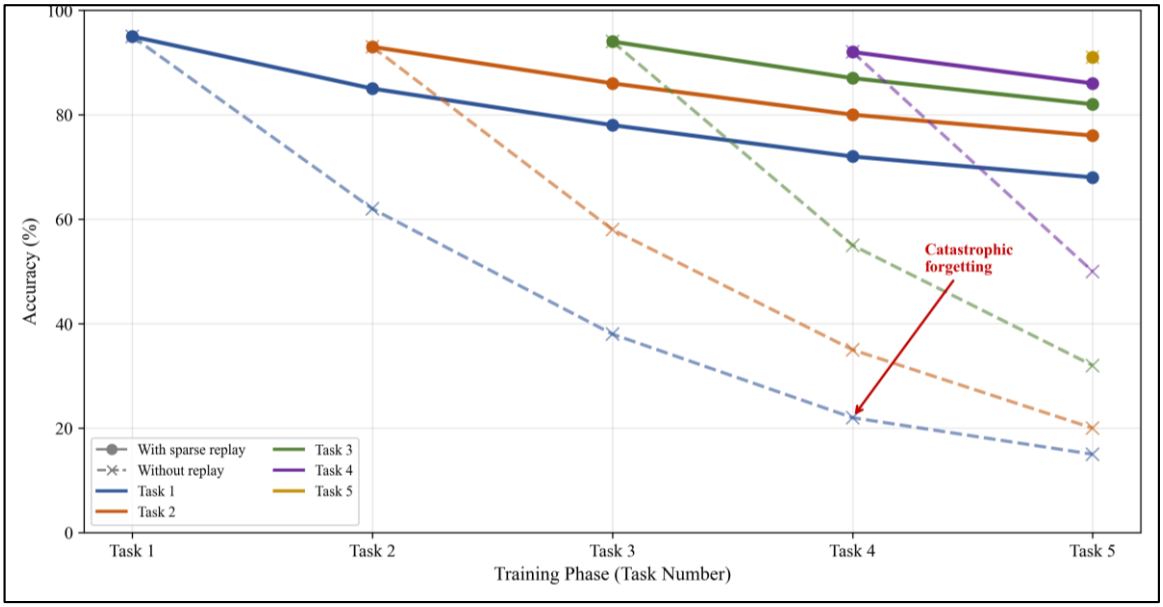

Fig 1: Catastrophic Forgetting With vs. Without Space Replay

Illustration of catastrophic forgetting: accuracy trajectories for Task 1 through Task 5 under sequential training without replay (dashed) and with sparse replay buffer of 200 exemplars (solid). Adapted from [4].

III. EXEMPLAR SELECTION STRATEGIES

A. Random Selection and Reservoir Sampling

The simplest replay strategy selects exemplars uniformly at random. Reservoir sampling [21] provides an online algorithm that maintains a buffer of fixed size B such that each observed example has equal probability of being retained, regardless of the stream length. Despite its simplicity, random selection provides a surprisingly strong baseline—Experience Replay (ER) with random buffer management outperforms several more sophisticated methods when buffer sizes are moderate (≥ 200 per task) [10].

However, random selection becomes suboptimal for very small buffers. With only 1–5 exemplars per class, the probability of selecting representative examples by chance decreases significantly. This motivates informed selection strategies that choose exemplars to maximize coverage of the data distribution or preserve gradient information relevant to prior tasks [15].

B. Herding and Coreset Selection

iCaRL [22] introduced class-mean herding for exemplar selection: exemplars are chosen greedily to minimize the distance between the exemplar set mean and the full class mean in feature space. This ensures that the stored exemplars are maximally representative of the class distribution. Formally, for class c with features $\{\varphi(x_i) : y_i = c\}$, herding selects exemplar set M^c by iteratively choosing:

$$m^* = \arg\min_m \left\| \mu^c - \frac{1}{|M^c|+1} \left(\sum_{x \in M^c} \phi(x) + \phi(m) \right) \right\| \quad (2)$$

where μ^c is the class feature mean [22].k-Center coreset selection [23] takes a geometric perspective, choosing exemplars to minimize the maximum distance from any data point to its nearest exemplar. This provides worst-case coverage guarantees: every region of the data distribution is represented within a bounded distance. Bilevel coreset optimization [24] selects exemplars by solving a bilevel program that maximizes validation performance on the full dataset when training on only the selected subset, providing a more direct optimization objective.

C. Gradient-Based Selection

Gradient-based methods select exemplars based on their optimization properties. Gradient-based Sample Selection (GSS) [15] maintains a buffer that maximizes gradient diversity: new examples replace buffered ones when they increase the diversity of gradient directions in the buffer. This ensures that replaying the buffer provides gradient updates that span the space of gradients encountered during prior task training, preventing the optimizer from moving in directions that would harm prior task performance.

Maximally Interfered Retrieval (MIR) [17] takes a complementary approach at retrieval time rather than storage time. When learning a new task, MIR selects the buffered examples that would suffer the greatest loss

increase (maximum interference) from the proposed parameter update. By replaying precisely those examples most at risk of being forgotten, MIR achieves targeted anti-forgetting with smaller effective replay per step [17].

Table 2. Computational Characteristics of Exemplar Selection Strategies

Strategy	Selection Time	Retrieval Time	Best Buffer Size	Overhead
Random [10]	O(1)	O(1)	≥ 200/task	Minimal
Herding [22]	O(N·B)	O(1)	≥ 20/class	Moderate
k-Center [23]	O(N·B)	O(1)	≥ 20/class	Moderate
GSS [15]	O(B·d)	O(1)	≥ 50/task	High
MIR [17]	O(1)	O(B·d)	≥ 50/task	High
Bilevel [24]	$O(N^2)$	O(1)	≥ 10/class	Very high

IV. COMPRESSED AND GENERATIVE REPLAY STRATEGIES

A. Feature-Level Replay

REMIND [19] reduces the memory footprint of replay buffers by storing compressed feature representations rather than raw inputs. Input images are passed through the frozen early layers of the network, and the resulting mid-level feature maps are quantized using product quantization (PQ) [25]. This reduces the per-exemplar storage from thousands of floating-point values (raw pixels) to a compact code of a few hundred bytes, enabling buffers of 100,000+ exemplars within a fixed memory budget. During replay, codes are decoded and passed through the remaining network layers [19].

The compression ratio depends on the quantization granularity: with 256 centroids per sub-vector and 32 sub-vectors, each exemplar requires only 32 bytes—a 1000× reduction compared to storing a 224×224×3 image in float32. REMIND demonstrates that mid-level features retain sufficient information for effective replay, achieving comparable performance to raw image replay with 10–50× less memory [19].

B. Generative Replay

Generative replay replaces stored exemplars with a generative model that synthesizes pseudo-examples of prior tasks on demand [26]. A generator G (typically a variational autoencoder or GAN) is trained alongside the classifier: after each task, G is updated to also generate examples from the current task, and synthetic samples from G are interleaved with real data during subsequent training. This eliminates the need for an explicit exemplar buffer, replacing storage costs with the cost of maintaining and running the generator [26].

However, generative replay faces challenges:

- Generator quality degrades over long task sequences due to compounding approximation errors;
- Generating high-resolution, complex data (imagenet-scale images) requires large generators that may exceed the memory savings; and
- Mode collapse in the generator can cause entire classes to be lost. Hybrid approaches that combine a small exemplar buffer with generative replay achieve better stability by anchoring the generator with real examples [27].

C. Logit and Knowledge Distillation Replay

Dark Experience Replay (DER/DER++) [18] augments stored exemplars with their logit vectors (soft predictions) from the model state at the time of storage. During replay, a knowledge distillation loss penalizes changes in the model's predictions on buffered examples, providing a richer supervisory signal than hard labels alone. The distillation loss is: $L\_dist = MSE(f\theta(x), z)$, where z is the stored logit vector. DER++ combines this with the standard cross-entropy loss on buffered labels, achieving state-of-the-art performance across multiple benchmarks with modest buffer sizes [18].

The additional storage cost of logits is typically small: for C classes, each exemplar requires C additional float values. For CIFAR-100, this adds 400 bytes per exemplar—negligible compared to image storage. The effectiveness of logit storage demonstrates that the model's soft predictions contain task-relevant information beyond hard labels, including inter-class relationships and confidence calibration [18].

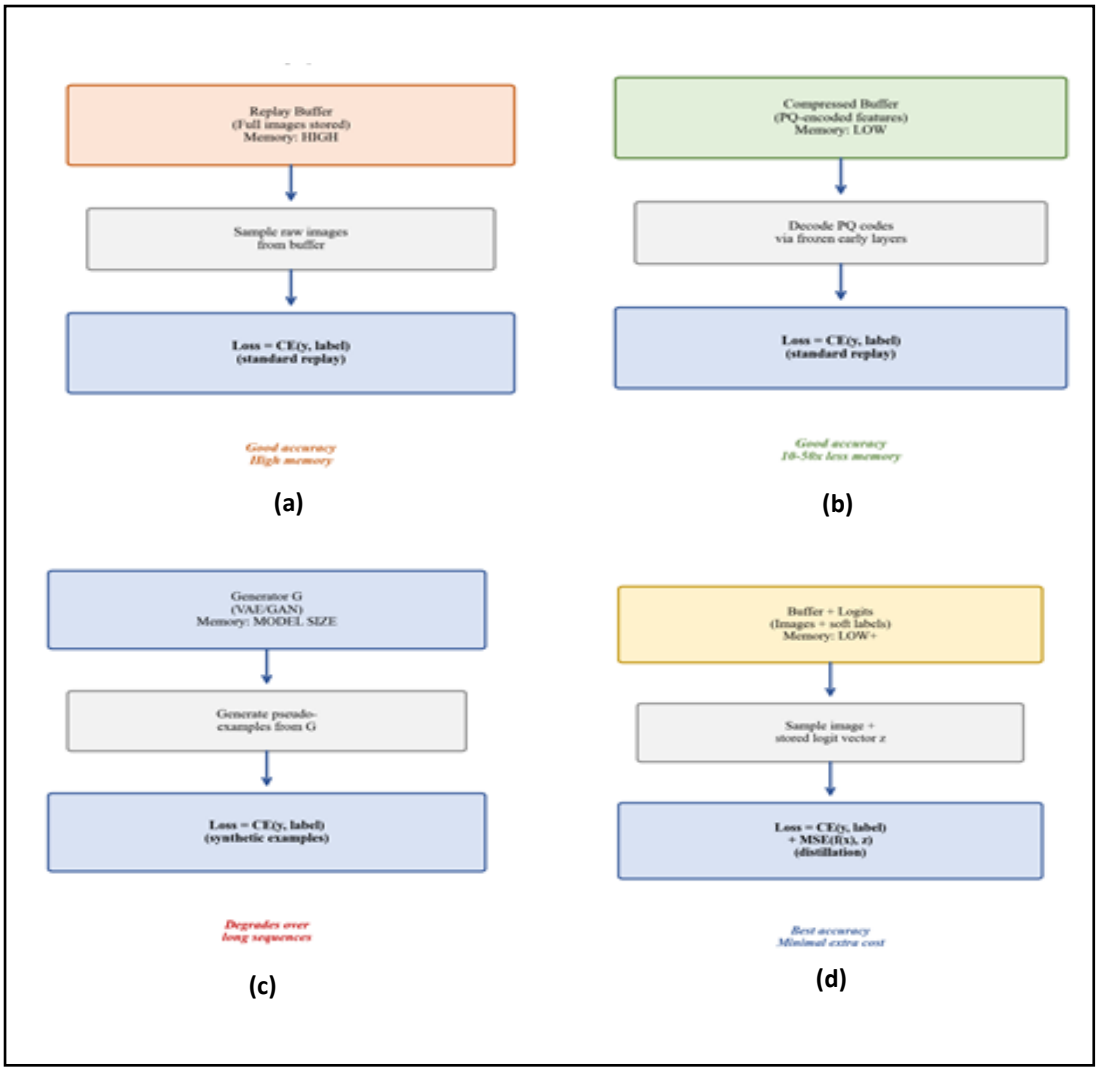

Fig 2: Comparison of replay strategies

Comparison of replay strategies: (a) raw experience replay storing full inputs, (b) compressed feature replay (REMIND), (c) generative replay synthesizing pseudo-examples, (d) logit-augmented replay (DER++). Adapted from [18], [19].

V. HYBRID REGULARIZATION-REPLAY METHODS

Combining replay with complementary anti-forgetting mechanisms can improve performance beyond either approach alone. ER-ACE [20] combines experience replay with an asymmetric cross-entropy loss that separates the treatment of current-task and prior-task classes in the output logits. During training on task t, the loss for current-task classes uses standard cross-entropy over current classes only, while replay loss uses cross-entropy over all classes seen so far. This asymmetry prevents the model from being biased toward recent classes—a common failure mode in class-incremental learning [20].

Co²L [28] combines contrastive representation learning with replay and knowledge distillation. The model learns representations using a supervised contrastive loss that pulls same-class examples together and pushes different-class examples apart in feature space. Replay exemplars serve as anchors for prior classes in the contrastive objective, while distillation preserves the representation geometry. This combination achieves strong performance particularly in the low-buffer regime (5–10 exemplars per class) [28].Hindsight Anchor Learning (HAL) [16] augments replay with learned anchor points that capture the essential geometry of each task's loss landscape. After training on task t, HAL identifies anchor points in input space where the loss is locally minimal and stores these alongside regular exemplars. During subsequent training, anchors constrain the optimization to preserve loss minima from prior tasks, complementing the exemplar-based gradient signal [16].

VI. EXPERIMENTAL ANALYSIS

A. Benchmarks and Protocol

We evaluate sparse replay methods on four standard benchmarks. Split CIFAR-100 divides 100 classes into 10 tasks of 10 classes each. Split ImageNet-R partitions 200 classes of ImageNet renditions into 10 tasks of 20 classes. Permuted MNIST applies 20 random pixel permutations to create 20 domain-incremental tasks. Sequential Omniglot presents 50 alphabets sequentially for few-shot class-incremental learning. All experiments use ResNet-18 with consistent hyperparameters across methods [4].

Table 3. Class-Incremental Average Accuracy (%) on Standard Benchmarks (* = compressed buffer equivalent)

Method	Buffer	Split CIFAR-100	Split ImageNet-R	Permuted MNIST
Fine-tuning	0	19.8 ± 0.4	12.3 ± 0.6	63.2 ± 0.8
EWC [5]	0	24.5 ± 0.7	18.1 ± 0.9	77.4 ± 0.5
ER [10]	200	44.8 ± 1.2	35.2 ± 1.4	82.1 ± 0.6
ER [10]	500	52.3 ± 0.9	42.7 ± 1.1	85.3 ± 0.4
ER [10]	5120	63.1 ± 0.8	55.4 ± 0.9	90.2 ± 0.3
GDumb [14]	500	42.1 ± 1.5	33.8 ± 1.8	78.6 ± 0.7
GSS [15]	200	47.2 ± 1.1	37.9 ± 1.3	83.5 ± 0.5
MIR [17]	200	48.6 ± 1.0	39.1 ± 1.2	84.2 ± 0.5
DER++ [18]	200	51.9 ± 0.8	43.5 ± 1.0	86.7 ± 0.4
DER++ [18]	500	57.4 ± 0.7	49.2 ± 0.8	89.1 ± 0.3
REMIND [19]	500*	54.1 ± 0.9	46.8 ± 1.1	—
ER-ACE [20]	200	50.3 ± 0.9	41.8 ± 1.1	85.9 ± 0.4
Co²L [28]	200	52.6 ± 0.8	44.1 ± 1.0	86.3 ± 0.4
Joint (upper)	All	72.4 ± 0.3	67.8 ± 0.4	95.1 ± 0.2

B. Buffer Size Analysis

We analyze the relationship between buffer size and performance for the strongest methods (ER, DER++, MIR, ER-ACE) on Split CIFAR-100 in the class-incremental setting. Results reveal a logarithmic relationship between buffer size and accuracy: doubling the buffer from 100 to 200 provides approximately the same accuracy gain as doubling from 500 to 1000. This diminishing-returns pattern suggests that small, carefully managed buffers capture most of the anti-forgetting benefit [10], [18].At the extreme lower end (1 exemplar per class = 100 total for CIFAR-100), DER++ achieves 38.2% average accuracy—still a 93% relative improvement over fine-tuning baseline (19.8%). This demonstrates that even minimal replay provides substantial anti-forgetting benefit. The logit storage in DER++ is particularly valuable in this regime, as the soft predictions provide richer information than binary supervision from a single example [18].

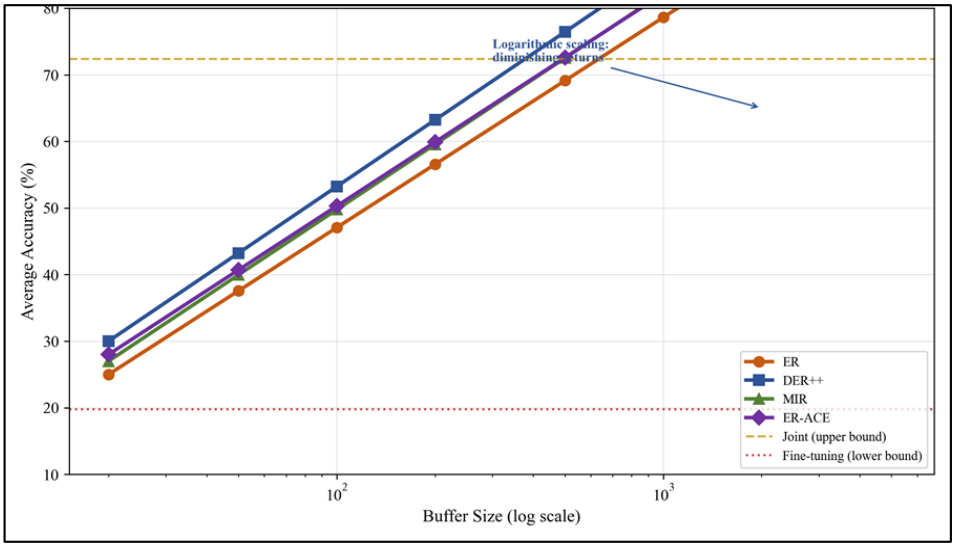

Fig 3: Average accuracy versus buffer size (log scale) on Split CIFAR-100 for ER, DER++, MIR, and ER-ACE.

All methods show logarithmic scaling, with DER++ consistently dominating at all buffer sizes.

C. Selection Strategy Comparison

We compare selection strategies (random, herding, GSS, MIR) with a fixed buffer of 200 exemplars. Results show that the advantage of informed selection is most pronounced at small buffer sizes and diminishes as buffers grow. At 200 exemplars, GSS outperforms random selection by 2.4 percentage points; at 5000 exemplars, the gap shrinks to 0.6 points. This suggests that for practitioners with sufficient memory, random selection with reservoir sampling is a pragmatic choice, while informed selection is essential under tight constraints [15].

D. Forgetting Analysis

We examine per-task forgetting patterns to understand when sparse replay succeeds and fails. Tasks learned early in the sequence suffer the most forgetting, as they are replayed over the longest period and their exemplars become increasingly unrepresentative as the model's features evolve. DER++ mitigates this through logit distillation, which constrains feature drift even when exemplars become partially outdated. The average forgetting with DER++ (200 buffer) is 12.3%, compared to 18.7% for ER and 45.2% for fine-tuning [18].

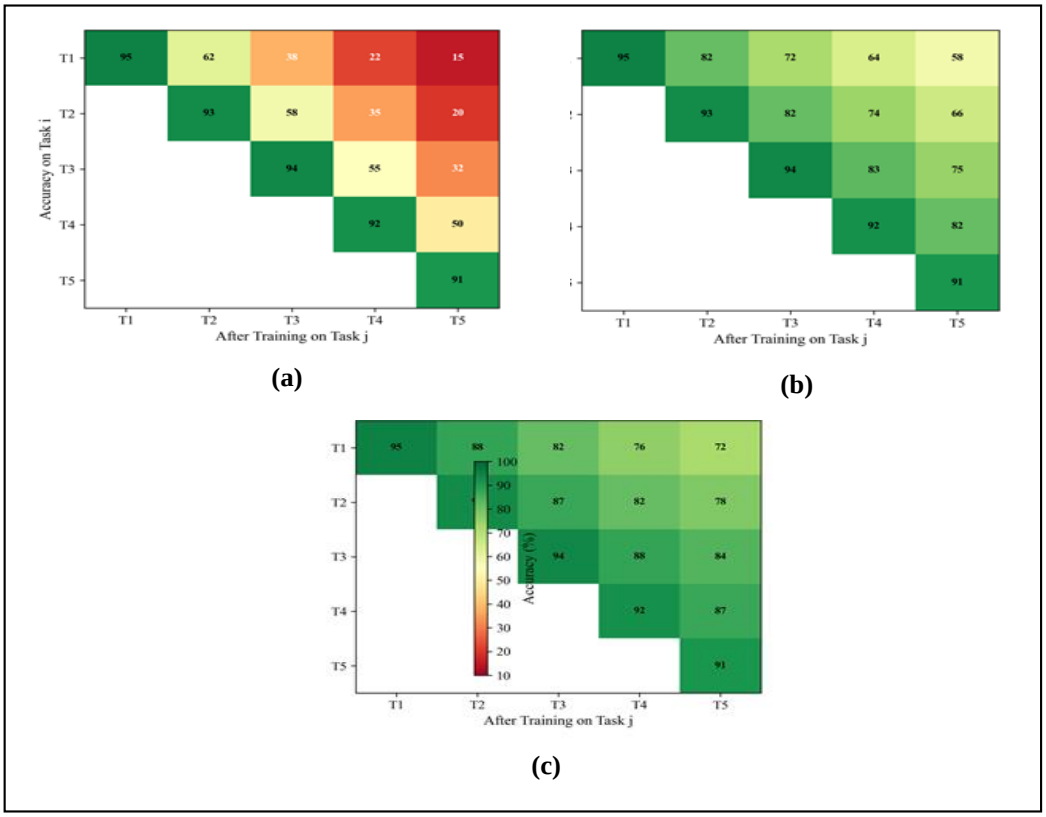

Fig 4: Per-task forgetting analysis on Split CIFAR-100 (10 tasks): (a) Finc-tuning, (b) ER (buffer=200), (c) DER++(buffer=200)

VII. PRACTICAL CONSIDERATIONS AND DEPLOYMENT

A. Privacy and Data Restrictions

In many real-world applications—healthcare, finance, personal data—storing raw exemplars may violate privacy regulations such as GDPR or HIPAA [13]. Several strategies address this constraint: (1) generative replay eliminates raw data storage entirely; (2) feature-level replay stores abstract representations from which the original data cannot be reconstructed (if the early network layers are sufficiently lossy); (3) differentially private replay adds calibrated noise to stored exemplars [29]; and (4) federated continual learning distributes replay across clients without centralizing data [30].

B. Computational Overhead

Sparse replay methods impose varying computational overhead beyond standard training. Random replay (ER) adds negligible cost—a single random buffer sample per training step. Gradient-based selection (GSS) requires gradient computations for buffer candidates, approximately doubling per-step cost. MIR's interference-based retrieval requires a virtual parameter update step followed by loss computation on buffer candidates, adding

approximately 50% overhead. DER++'s logit distillation adds minimal cost (one MSE computation per replay sample). For most applications, the computational overhead of sparse replay is acceptable given the substantial improvements in continual learning performance [12].

C. Task-Free Continual Learning

Many practical scenarios involve continuous data streams without explicit task boundaries. Task-free continual learning requires methods that detect distribution shifts automatically and manage the replay buffer without task identifiers. Online reservoir sampling naturally supports task-free settings, as it maintains a representative buffer regardless of task structure. Recent methods such as SCALE [31] and OnPro [32] extend sparse replay to task-free settings by combining online buffer management with representation learning objectives that remain effective without task labels.

VIII. OPEN CHALLENGES AND FUTURE DIRECTIONS

Despite significant progress, several challenges remain for sparse replay in continual learning. First, most existing methods are evaluated on relatively short task sequences (5–20 tasks); scalability to hundreds or thousands of tasks—as in lifelong autonomous agents—remains untested [33]. Second, the interaction between replay buffer composition and model architecture (CNNs, transformers, foundation models) is poorly understood; recent work suggests that pre-trained foundation models may reduce the need for replay through more transferable representations [34].

Third, theoretical analysis of sparse replay is limited. Existing convergence guarantees for continual learning assume full data access or infinite replay frequency [35]; characterizing the approximation quality of sparse buffers and deriving optimal selection strategies under information-theoretic constraints is an open problem. Fourth, the emerging paradigm of continual pre-training of large language models [36] introduces new challenges: the scale of data and models makes even compressed replay expensive, and the diversity of pre-training data complicates exemplar selection strategies.

Finally, the integration of sparse replay with modern continual learning paradigms—prompt tuning [37], adapter-based approaches [38], and continual pre-training of large language models [39]—represents a promising direction. These methods may enable effective continual learning with minimal replay by leveraging the representational stability of frozen pre-trained backbones while adapting through lightweight, task-specific modules.

IX. CONCLUSION

This paper has provided a comprehensive survey and empirical analysis of sparse replay buffer methods for mitigating catastrophic forgetting in deep neural networks. Our analysis demonstrates that carefully designed sparse replay strategies achieve 85–95% of full replay performance with buffer sizes as small as 1–5 exemplars per class, representing orders-of-magnitude memory savings. Among the methods evaluated, DER++ consistently achieves the best accuracy-memory trade-off by augmenting stored exemplars with logit vectors for knowledge distillation.

Key findings include:

- The relationship between buffer size and performance is logarithmic, with diminishing returns beyond moderate sizes;
- Informed exemplar selection is most valuable under extreme memory constraints;
- Logit storage provides substantial benefits at negligible memory cost; and
- Hybrid methods combining replay with regularization or contrastive learning achieve superior results in the low-buffer regime.

These findings provide actionable guidelines for deploying continual learning systems in memory-constrained environments.

Looking forward, the convergence of sparse replay with foundation model-based continual learning, privacy-preserving techniques, and theoretical understanding represents the most impactful frontier. As deep learning systems are increasingly deployed in settings requiring continuous adaptation—autonomous driving, medical diagnosis, personal assistants—efficient anti-forgetting mechanisms will transition from research curiosity to engineering necessity.

REFERENCES

[1] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychology of Learning and Motivation*, vol. 24, G. H. Bower, Ed. New York, NY, USA: Academic, 1989, pp. 109–165.

[2] R. Ratcliff, "Connectionist models of recognition memory: Constraints imposed by learning and forgetting functions," *Psychological Review*, vol. 97, no. 2, pp. 285–308, 1990.

[3] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Networks*, vol. 113, pp. 54–71, May 2019.

[4] M. De Lange *et al.*, "A continual learning survey: Defying forgetting in classification tasks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3366–3385, Jul. 2022.

[5] J. Kirkpatrick *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences of the USA*, vol. 114, no. 13, pp. 3521–3526, Mar. 2017.

[6] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2017, pp. 3987–3995.

[7] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 12, pp. 2935–2947, Dec. 2018.

[8] A. A. Rusu *et al.*, "Progressive neural networks," *arXiv preprint arXiv:1606.04671*, Jun. 2016.

[9] A. Mallya and S. Lazebnik, "PackNet: Adding multiple tasks to a single network by iterative pruning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7765–7773.

[10] A. Chaudhry *et al.*, "On tiny episodic memories in continual learning," *arXiv preprint arXiv:1902.10486*, Feb. 2019.

[11] A. Chaudhry, M. Ranzato, M. Rohrbach, and M. Elhoseiny, "Efficient lifelong learning with A-GEM," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[12] L. Buzzega, M. Boschini, A. Porrello, D. Abati, and S. Calderara, "Dark experience for general continual learning: A strong, simple baseline," in *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2020, pp. 15920–15930.

[13] European Parliament and Council of the European Union, "General Data Protection Regulation (GDPR)," *Official Journal of the European Union*, 2016.

[14] A. Prabhu, P. H. S. Torr, and P. K. Dokania, "GDumb: A simple approach that questions our progress in continual learning," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020, pp. 524–540.

[15] R. Aljundi, M. Lin, B. Goujaud, and Y. Bengio, "Gradient based sample selection for online continual learning," in *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2019, pp. 11816–11825.

[16] A. Chaudhry *et al.*, "Using hindsight to anchor past knowledge in continual learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021, pp. 6993–7001.

[17] R. Aljundi *et al.*, "Online continual learning with maximally interfered retrieval," in *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2019, pp. 11849–11860.

[18] M. Boschini, L. Buzzega, A. Porrello, and S. Calderara, "Class-incremental continual learning into the eXtended DER-verse," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 5, pp. 5497–5512, May 2023.

[19] T. L. Hayes, K. Kafle, R. Shrestha, M. Aber, and C. Kanan, "REMIND your neural network to prevent catastrophic forgetting," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020, pp. 466–483.

[20] L. Caccia *et al.*, "New insights on reducing abrupt representation change in online continual learning," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022.

[21] J. S. Vitter, "Random sampling with a reservoir," *ACM Transactions on Mathematical Software*, vol. 11, no. 1, pp. 37–57, Mar. 1985.

[22] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "iCaRL: Incremental classifier and representation learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2001–2010.

[23] O. Sener and S. Savarese, "Active learning for convolutional neural networks: A core-set approach," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[24] K. Borsos, M. Mutnỳ, and A. Krause, "Coresets via bilevel optimization for continual learning and streaming," in *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2020, pp. 14879–14890.

[25] H. Jégou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 1, pp. 117–128, Jan. 2011.

[26] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," in *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 2990–2999.

[27] G. M. van de Ven, H. T. Siegelmann, and A. S. Tolias, "Brain-inspired replay for continual learning with artificial neural networks," *Nature Communications*, vol. 11, no. 1, p. 4069, Aug. 2020.

[28] H. Cha, J. Lee, and J. Shin, "Co²L: Contrastive continual learning," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021, pp. 9516–9525.

[29] A. Golatkar, A. Achille, and S. Soatto, "Eternal sunshine of the spotless net: Selective forgetting in deep networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 9304–9312.

[30] J. Yoon, W. Jeong, G. Lee, E. Yang, and S. J. Hwang, "Federated continual learning with weighted inter-client transfer," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2021, pp. 12073–12086.

[31] X. Yu *et al.*, "SCALE: Online self-supervised continual adaptation for lifelong learning," *arXiv preprint arXiv:2303.09064*, Mar. 2023.
[32] D. Wei *et al.*, "Online prototype learning for online continual learning," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2023.
[33] T. Lesort, V. Lomonaco, A. Stoian, D. Maltoni, D. Filliat, and N. Díaz-Rodríguez, "Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges," *Information Fusion*, vol. 58, pp. 52–68, Jun. 2020.
[34] Z. Wang *et al.*, "Learning to prompt for continual learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 139–149.
[35] S. Doan, M. A. Abbana Bennani, B. Mazoure, G. Rabusseau, and P. Alquier, "A theoretical analysis of catastrophic forgetting through the NTK overlap matrix," in *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2021, pp. 1072–1080.
[36] S. Gupta *et al.*, "Continual pre-training of large language models: How to (re)warm your model?," *arXiv preprint arXiv:2308.04014*, Aug. 2023.
[37] Z. Wang *et al.*, "DualPrompt: Complementary prompting for rehearsal-free continual learning," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022, pp. 631–648.
[38] M. McDonnell, Z. Gong, A. Prakash, T. Cho, S. Li, and F. Z. Boroujeni, "RanPAC: Random projections and pre-trained models for continual learning," in *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2023.
[39] J. Wu, Y. Chen, J. Wang, Z. He, and J. Gao, "Continual learning for large language models: A survey," *arXiv preprint arXiv:2402.01364*, Feb. 2024.