Blurry Class-Incremental Learning for IMU-Based Human Activity Recognition: An Empirical Study

Takumi Yamamoto, Suguru Kanoga, Mitsunori Tada, Yuta Sugiura

***Abstract*— Inertial measurement unit (IMU)-based human activity recognition (HAR) has attracted considerable attention, leading to a growing demand for systems that support long-term deployment. In such scenarios, user requirements may evolve over time, necessitating the ability to recognize additional activity classes. Class-incremental learning (CIL) offers a promising approach by enabling models to incorporate new classes without retraining from scratch. Although previous studies have examined CIL in the context of HAR, they have largely overlooked cases where the same classes reappear across different tasks–a setting known as the Blurry class-incremental learning (B-CIL) scenario. In this work, we investigate the B-CIL scenario for IMU-based HAR and conduct extensive experiments on two widely used IMU datasets (UCI-HAR and USC-HAD). We evaluate nine continual learning methods under multiple configurations of overlapping classes. Our results demonstrate that replay-based methods consistently outperform regularization-based methods in the B-CIL scenario. Furthermore, we observe that increasing the number of overlapping classes can lead to improved performance. In the future, we aim to extend our study to additional datasets and explore more realistic blurry scenarios, including online continual learning.**

I. Introduction

Human activity recognition (HAR) aims to classify daily human activities based on sensor data and has attracted significant attention in recent years [1]–[3]. Among various sensing modalities, inertial measurement unit (IMU)-based HAR has been widely adopted for applications such as smart homes [4], healthcare [5], and input interfaces [6]. As these applications are increasingly deployed over extended periods [7], HAR systems must adapt to evolving user behaviors and newly introduced activity classes [8].

A naïve retraining approach using all available data is often impractical due to resource constraints and privacy concerns [9]. Moreover, plasticity-focused methods such as fine-tuning typically suffer from catastrophic forgetting [10]–[12], where performance on previously learned tasks deteriorates as the model overfits to new data. Consequently, there is growing interest in continual learning (CL) methods, which enable models to learn incrementally without forgetting prior knowledge [13], [14]. CL methods are generally categorized into three families: regularization-based, replay-based, and parameter isolation methods.

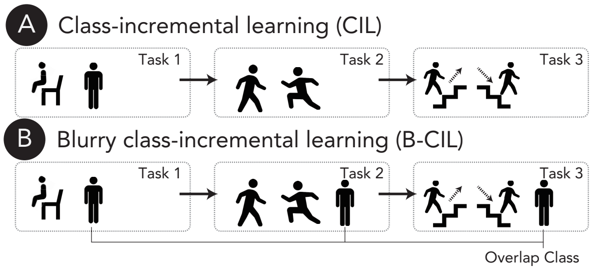
A
Class-incremental learning (CIL)
Task 1
Task 2
Task 3
B
Blurry class-incremental learning (B-CIL)
Task 1
Task 2
Task 3
Overlap Class

Fig. 1. Overview of (A) class-incremental learning and (B) blurry class-incremental learning scenarios.

Within CL, different problem scenarios are defined based on how tasks arrive sequentially. One widely studied scenario is class-incremental learning (CIL), where the model must learn an increasing number of classes over time without task labels and with no class repetition across tasks (Fig. 1(A)). However, real-world systems rarely operate under conditions where classes are completely disjoint across tasks. Blurry class-incremental learning (B-CIL) relaxes this constraint by allowing partial class overlap between tasks, making it more representative of practical deployment scenarios (Fig. 1(B)).

For IMU-based HAR, previous studies have explored CIL scenarios [15]–[22], but most assume that classes are fully disjoint. The only related work by Zhang et al. [23] considers a setting where one new class is added at each task and later reappears once. However, in real-world scenarios, old and new classes often coexist within the same task. To date, no systematic investigation has addressed B-CIL scenarios with varying degree of class overlap.

In this study, we evaluate nine CL methods under B-CIL scenarios with different levels of class overlap. Experiments are conducted on two publicly available datasets: the University of California Human Activity Recognition (UCI-HAR) and the University of Southern California Human Activity Dataset (USC-HAD). Our findings reveal that replay-based methods consistently outperform regularization-based ones and that increasing class overlap improves performance.

The research questions addressed in this work are as follows:

- **RQ1:** Which CL methods are most effective in B-CIL scenarios for IMU-based HAR?
- **RQ2:** How does the degree of class overlap influence

*This work was partially supported by JSPS KAKENHI Grant-in-Aid for Scientific Research (B) Grant Number JP23K28135, and Programs for Bridging the gap between R&D and the IDeal society (society 5.0) and Generating Economic and social value (BRIDGE)/Practical Global Research in the AI × Robotics Services, implemented by the Cabinet Office, Government of Japan.

T. Yamamoto and Y. Sugiura are with Keio University, Kanagawa, Japan (E-mail: imuka06x17@keio.jp).

T. Yamamoto, S. Kanoga, and M. Tada are with the National Institute of Advanced Industrial Science and Technology (AIST), Tokyo, Japan.

the performance of CL methods in B-CIL scenarios?

II. Continual Learning Framework and Scenario Definitions

In a CL scenario, tasks arrive sequentially, and the model $f(\boldsymbol{x};\theta)$ is incrementally updated upon the arrival of each new task, where $\boldsymbol{x}$ denotes the input data and θ represents the model parameters. Formally, the sequence of tasks is defined as $T = \{\tau^1, \tau^2, ..., \tau^N\}$, where each task τ^t ($t \in \{1, ..., N\}$) is associated with a dataset $D^t = \{(x_i^t, y_i^t) | y_i^t \in Y^t\}_{i=1}^{N_t}$ and a corresponding label space Y^t. Each input x_i^t is sampled from a task-specific input domain $\mathcal{X}^t$, and each label y_i^t belongs to the respective label space Y^t. We denote the model before training on task τ^t as f_{θ_t}, and the model after optimization as $f_{\theta_t^*}$. The CL scenarios considered in this study are defined as follows:

- CIL: Each task introduces a set of new, mutually exclusive classes. Formally, for any $i \neq j$, the label spaces are disjoint, i.e., $Y^i \cap Y^j = \varnothing$, indicating that no class is shared between tasks.
- B-CIL: This scenario relaxes the strict disjoint-class assumption of CIL by allowing partial overlap between the label spaces of tasks. Specifically, for some $i \neq j$, $Y^i \cap Y^j = \varnothing$, indicating that certain classes may appear in multiple tasks.

III. Methods

A. Backbone Model

As the backbone model, we employed one-dimensional convolutional neural network (1D-CNN) architecture commonly used in previous class-incremental learning studies [18]. Specifically, we adopted the architecture proposed in [18], which consists of four convolutional blocks. Each block consists of a convolutional layer (kernel size: 5, stride: 1, padding: 2) with output channels of 64, 128, 256, and 128, respectively. Following the convolutional layer, the block includes a ReLU activation function, a batch normalization layer, and a max-pooling layer (kernel size: 2, stride: 2).

B. Baselines

As baselines, we considered two approaches. The first is a naïve fine-tuning strategy (Naïve), in which the model is incrementally updated on each new task without applying any CL methods, serving as a reference point for catastrophic forgetting. The second is an offline setting (Offline), where the model is trained using data from all tasks simultaneously. This represents an upper-bound performance reference, as it assumes full access to the entire dataset across tasks.

C. Continual Learning Methods

As the CL methods for comparison, we selected nine representative methods: four regularization-based methods–Learning without Forgetting (LwF) [24], Elastic Weight Consolidation (EWC) [25], Memory Aware Synapses (MAS) [26], and Synaptic Intelligence (SI) [27]–and five replay-based methods–Experienced Replay (ER) [28], Dark Experience Replay (DER) [29], Fast Incremental Classifier and Representation Learning (FastICARL) [30], Adversarial Shapley value Experience Replay (ASER) [31], and Generative Replay (GR) [32].

IV. Materials

We used two publicly available datasets; UCI-HAR and USC-HAD.

UCI-HAR [33] comprises data collected from 30 subjects using a waist-mounted smartphone (Samsung Galaxy S II) equipped with inertial sensors. The sampling rate was set to 50 Hz. The dataset includes six activities: (1) walking, (2) walking upstairs, (3) walking downstairs, (4) sitting, (5) standing, and (6) lying. Each sample contains nine feature dimensions, consisting of 3-axis total acceleration, 3-axis estimated body acceleration, and 3-axis angular velocity. Sensor signals were segmented using sliding windows of 2.56 s and 50 % overlap, resulting in window shapes of 128×9.

USC-HAD [34] comprises data collected from 14 subjects using a single IMU sensor (MotionNode) positioned on the front right side of the body. The sampling rate was set to 100 Hz. The dataset includes 12 activities: (1) walking forwards, (2) walking left, (3) walking right, (4) walking upstairs, (5) walking downstairs, (6) running forwards, (7) jumping, (8) sitting, (9) standing, (10) sleeping, (11) elevator up, and (12) elevator down. Each sample has six feature dimensions, consisting of 3-axis acceleration and 3-axis angular velocity. Sensor signals were segmented using sliding windows of 1.28 s and 50% overlap, resulting in window shapes of 128×6.

V. Experiments

In this study, we evaluated two B-CIL scenarios (B-CIL1 and B-CIL2), and compared them against the standard CIL scenario. Fig. 2 illustrates the detailed configurations of these scenarios for the two datasets: (A) UCI-HAR and (B) USC-HAD.

A. Class Allocation Across Tasks

In this study, the number of tasks was set to three. In the CIL scenario, classes from each dataset were evenly distributed across tasks to ensure that no class overlapped between tasks. Consequently, for the UCI-HAR dataset, which contains six classes, each task was assigned two classes, while for the USC-HAD dataset, which contains twelve classes, each task was assigned four classes.

In the B-CIL1 scenario, compared to CIL, one class from the Task 1 was also assigned to Tasks 2 and 3. As a result, the number of classes per task was [2, 3, 3] for UCI-HAR and [4, 5, 5] for USC-HAD.

In the B-CIL2 scenario, two classes from Task 1 were shared with both Tasks 2 and 3. Therefore, the number of classes per task was [2, 4, 4] for UCI-HAR and [4, 6, 6] for USC-HAD.

The order of classes within each task was determined randomly. The split into training, validation, and test sets

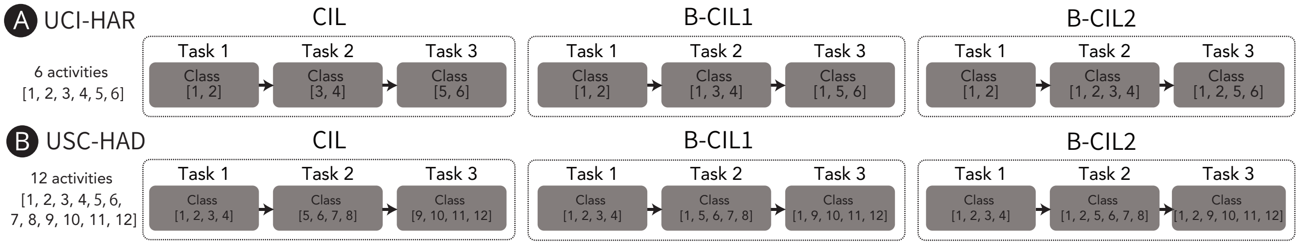

Fig. 2. Three configurations of CIL and B-CIL scenarios for two datasets: (A) UCI-HAR and (B) USC-HAD datasets.

was also performed randomly, with 60% of the data used for training, 20% for validation, and 20% for testing.

The assignment of classes to tasks and the data split into training, validation, and test sets may affect the results. Therefore, We performed five different random assignments of classes to tasks by varying the random seed. For each of these assignments, we also tested five different data splitting strategies for training, validation, and test sets. The final results are reported as the average of these trials.

B. Evaluation Metrics

We employed three evaluation metrics: (1) Final Average Accuracy, (2) Final Average Forgetting, and (3) Average Learning Accuracy. Let $a_{i,j}$ denote the average classification accuracy on Task j after the model has been trained on Task i, where $j \leqq i$. The total number of tasks is T.

1) Final Average Accuracy: This metric represents the average accuracy across all tasks after the model has completed training on all tasks:

$$\mathcal{A}_T = \frac{1}{T} \sum_{i=1}^{T} a_{T,i}. \tag{1}$$

2) Final Average Forgetting: This metric quantifies the average decrease in accuracy on previous tasks due to learning subsequent tasks:

$$\mathcal{F}_T = \frac{1}{T-1} \sum_{j=1}^{T} \max_{k \in \{1,\dots,T-1\}} (a_{k,j} - a_{i,j})(j < i). \tag{2}$$

3) Average Learning Accuracy: This metric measures the average accuracy on the current task immediately after training on that task:

$$\mathcal{A}_{\text{cur}} = \frac{1}{T} \sum_{i=1}^{T} a_{i,i}. \tag{3}$$

C. Learning Protocol

For the training configuration, Cross-Entropy Loss was used as the objective function, and the OneCycle learning rate scheduler was employed. The model was trained for 100 epochs. Early stopping was applied to the regularization-based methods and baseline method, with a patience parameter of 5.

For all methods, hyperparameter tuning was performed using grid search. The batch size, maximum learning rate, and learning rate adjustment strategy were optimized. The search space for batch size was 32, 64, 128, and for the initial learning rate, it was 0.01, 0.001, 0.0001. For the learning rate adjustment strategy, three approaches were evaluated: (1) reducing the learning rate by a factor of 0.1 every 10 epochs, (2) reducing it by a factor of 0.1 every 25 epochs, and (3) using only the scheduler without manual adjustments. Details of the search space for each method are summarized in Table I. Note that the optimal hyperparameter values depended on the class-to-task assignment and on the specific data splits used for training, validation, and testing.

TABLE I
HYPERPARAMETER SEARCH SPACE IN THIS STUDY FOR CL METHODS.

Method	Hyperparameter	Search range
LwF	λ	1, 0.1, 0.01, 0.001, 0.0001
EWC	λ	0.01, 0.001, 0.0001
MAS	λ	0.001, 0.0001, 0.00001
SI	λ	0.01, 0.001, 0.0001
ASER	Number of sample saved in buffer per class	2, 4
GR	Learning rate for generator Weight for reconstruction loss	0.001, 0.0001 0.01, 0.1, 1, 10

D. Implementation

All methods were implemented in Python 3.10.10 and PyTorch 1.13.1. Experiments were conducted on a machine running Ubuntu 22.04, equipped with four NVIDIA L40S GPUs.

VI. RESULTS

Table II presents the performance of the baseline and CL methods on the UCI-HAR and USC-HAD datasets. When comparing the three scenarios across both datasets, CIL consistently achieved the lowest Final Average Accuracy $\mathcal{A}_T$, followed by B-CIL1, while B-CIL2 yielded the highest accuracy–except for FastICARL and GR on USC-HAD. Regarding Final Average Forgetting $\mathcal{F}_T$, CIL exhibited the greatest degree of forgetting. As the number of overlapping classes increased, forgetting decreased, with B-CIL2 showing the lowest forgetting (except for FastICARL on USC-HAD). Average Learning Accuracy $\mathcal{A}_{\text{cur}}$ varied across methods without a clear trend. Overall, these results indicate that increasing the number of overlapping classes mitigates forgetting and improves final accuracy.

TABLE II
RESULTS OF BASELINE AND CL METHODS ON UCI-HAR AND USC-HAD DATASETS.

Dataset	Metrics	Scenario	Naïve	Offline	LwF	EWC	MAS	SI	ER	DER	ASER	FastICARL	GR
$\mathcal{A}_T$	UCI-HAR	CIL	33.04	98.00	34.35	35.36	40.47	34.35	70.97	66.62	**96.76**	66.44	36.34
		B-CIL1	61.16	N.A.	65.49	63.52	70.58	63.91	89.17	88.74	**98.86**	89.69	62.70
		B-CIL2	82.52	N.A.	83.15	83.44	84.60	80.56	96.00	92.83	**99.40**	95.64	83.49
	USC-HAD	CIL	32.82	91.43	38.72	37.15	38.69	33.64	76.94	64.42	**94.11**	73.13	29.23
		B-CIL1	49.07	N.A.	57.11	54.75	54.59	51.55	82.58	72.32	**95.67**	80.93	54.49
		B-CIL2	58.04	N.A.	62.34	65.18	60.79	57.17	86.78	77.30	**96.28**	78.67	50.85
$\mathcal{F}_T$	UCI-HAR	CIL	99.15	N.A.	91.35	95.44	85.20	88.95	19.80	17.06	**4.24**	29.51	91.81
		B-CIL1	56.78	N.A.	46.46	53.12	39.78	46.72	7.38	5.43	**1.27**	8.06	54.35
		B-CIL2	23.98	N.A.	19.42	22.66	18.24	20.89	2.81	3.53	**0.57**	2.97	22.92
	USC-HAD	CIL	98.62	N.A.	77.66	88.55	78.80	70.18	12.36	9.16	**3.94**	14.32	65.99
		B-CIL1	72.65	N.A.	55.73	62.03	55.33	50.61	8.38	6.32	**2.23**	11.47	52.65
		B-CIL2	58.60	N.A.	45.21	44.56	46.31	41.62	5.35	4.90	**1.48**	16.43	44.3
$\mathcal{A}_{\text{cur}}$	UCI-HAR	CIL	99.14	N.A.	96.75	98.98	97.27	93.65	81.37	76.74	**99.58**	85.54	97.55
		B-CIL1	99.01	N.A.	96.46	98.84	97.10	94.06	93.31	92.03	**99.69**	95.03	98.44
		B-CIL2	98.40	N.A.	96.08	98.46	96.74	94.48	97.78	95.06	**99.67**	97.52	98.58
	USC-HAD	CIL	**97.90**	N.A.	90.49	96.18	91.22	80.43	85.01	68.95	96.73	82.64	73.23
		B-CIL1	**97.50**	N.A.	94.27	96.10	91.48	85.21	88.07	75.81	97.15	88.53	89.59
		B-CIL2	97.11	N.A.	92.48	94.89	91.67	84.80	90.26	79.71	**97.15**	89.61	80.39

We compared the Final Average Accuracy $\mathcal{A}_T$ of CL methods against baselines (i.e., Naïve and Offline). In the CIL scenario, most CL methods achieved higher accuracy than Naïve, except for GR on USC-HAD. In the B-CIL1 scenario, all CL methods outperformed Naïve. In the B-CIL2 scenario, CL methods generally surpassed Naïve, with the exception of GR on USC-HAD and SI on both UCI-HAR and USC-HAD. Overall, across CIL and scenarios with task overlap, most CL methods tend to outperform the Naive baseline. Compared to the Offline baseline, most CL methods exhibited the lower accuracy across almost all scenarios, but surprisingly, ASER in B-CIL1 and B-CIL2 on UCI-HAR, and all scenarios on USC-HAD slightly outperformed Offline. Offline training incorporates all data, including noisy or outlier samples, which can negatively affect the high accuracy. In contrast, ASER selectively retain informative samples, potentially mitigating the impact of noise and enabling more effective learning from past data.

Among the evaluated CL methods, most replay-based methods outperformed regularization-based methods across both datasets and all scenarios, with exception of GR. For the four replay-based methods (ER, DER, ASER, and FastICARL), the average Final Average Accuracy $\mathcal{A}_T$ was 75.20 in CIL, 91.62 in B-CIL1, and 95.97 in B-CIL2 for UCI-HAR, and 77.15 in CIL, 82.88 in B-CIL1, and 84.76 in B-CIL2 for USC-HAD. In contrast, for the regularization-based methods (LwF, EWC, MAS, and SI), the average Final Average Accuracy was 36.13 in CIL, 70.04 in B-CIL1, and 82.94 in B-CIL2 for UCI-HAR, and 37.05 in CIL, 55.40 in B-CIL1, and 61.37 in B-CIL2 for USC-HAD. Among the replay-based methods, ASER achieved the highest Final Average Accuracy across all scenarios for both UCI-HAR and USC-HAD.

When comparing UCI-HAR and USC-HAD, the CIL scenario yields higher accuracy on USC-HAD for methods such as LwF, EWC, ER, and FastICARL. In contrast, B-CIL1 and B-CIL2 scenarios consistently favor UCI-HAR. This difference is attributable to the greater class-level overlap in UCI-HAR: B-CIL1 includes 1 of 6 classes (16.7%) compared to 1 of 12 (8.3%) in USC-HAD, while B-CIL2 includes 2 of 6 classes (33.3%) compared to 2 of 12 (16.7%). Increased overlap reinforces previously learned representations, thereby reducing catastrophic forgetting.

VII. DISCUSSION

A. Answers to Research Questions

1) RQ1: Which CL methods are most effective in B-CIL scenarios for IMU-based HAR?: Across both UCI-HAR and USC-HAD datasets, replay-based methods consistently outperformed regularization-based methods in B-CIL scenarios. This observation is consistent with prior findings in CIL scenarios [18], which also reported the superiority of replay-based methods. Among the replay-based methods, ASER achieved the highest Final Average Accuracy across all B-CIL scenarios and both datasets.

2) RQ2: How does the degree of class overlap influence the performance of CL methods in B-CIL scenarios?: Increasing the degree of class overlap in B-CIL scenarios had a positive impact on the performance of CL methods. As the number of overlapping classes increased from B-CIL1 to B-CIL2, most methods exhibited a notable reduction in forgetting and an improvement in Final Average Accuracy. Furthermore, the improvement was more pronounced when the proportion of overlapping classes relative to the total number of classes was higher, reinforcing the benefit of overlap in mitigating catastrophic forgetting.

B. Limitations and Future Work

1) Expansion to other datasets and methods: This study evaluated the performance of CL methods in B-CIL scenarios using two publicly available datasets: UCI-HAR and USC-HAD. A key limitation is that the experiments were restricted to these two datasets. In future work, we plan to extend the evaluation to additional human activity datasets to investigate how factors such as the number and order of overlapping classes influence model accuracy.

To the best of our knowledge, no previous studies have applied the B-CIL scenarios to time-series data. While CIL has been explored for IMU data and other time-series modalities, such as surface electromyogram (sEMG) signals [35], the blurry setting remains unexplored. Future research will aim to generalize the B-CIL framework to other types of time-series data.

In this study, we focused on four regularization-based methods (LwF, EWC, MAS, and SI) and five replay-based methods (ER, ASER, DER, FastICARL, and GR). Parameter isolated methods, such as Progressive Neural Networks (PNN) [36], were not considered. Incorporating such methods into future experiments will provide a more comprehensive comparison of CL strategies under B-CIL scenarios.

2) More various blurry scenarios: In this study, the model was trained and updated using batch learning. However, in real-world applications, data often arrives in a streaming format. As future work, it is essential to evaluate the performance of online CL [20], where the model is updated incrementally as data stream in, under B-CIL scenarios.

Furthermore, this study applied the Blurry problem setting exclusively to CIL. In contrast, Blurry settings can also be extended to Domain-Incremental Learning (Domain-IL) [37], [38], where the domain changes rather than the classes. Future research will focus on adapting the Blurry problem setting to Domain-IL and assessing its effectiveness.

VIII. Conclusion

This study introduced the B-CIL scenario for IMU-based HAR, addressing a previously overlooked challenge in continual learning. Through comprehensive experiments on two benchmark datasets, we demonstrated that replay-based methods are particularly effective in mitigating catastrophic forgetting under the B-CIL scenario. Our analysis further highlights the positive impact of class overlap on model stability and accuracy, offering new insights into designing robust incremental learning systems for real-world HAR applications. Future work will focus on extending these findings to diverse datasets and exploring online and streaming settings to better reflect practical deployment scenarios.

References

[1] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE Commun. Surv. Tutor.*, vol. 15, no. 3, pp. 1192–1209, 2012.

[2] F. Attal, S. Mohammed, M. Dedabrishvili, F. Chamroukhi, L. Oukhellou, and Y. Amirat, "Physical human activity recognition using wearable sensors," *Sensors*, vol. 15, no. 12, pp. 31 314–31 338, 2015.

[3] F. Kulsoom, S. Narejo, Z. Mehmood, H. N. Chaudhry, A. Butt, and A. K. Bashir, "A review of machine learning-based human activity recognition for diverse applications," *Neural Comput. Appl.*, vol. 34, no. 21, pp. 18 289–18 324, 2022.

[4] Y. Du, Y. Lim, and Y. Tan, "A novel human activity recognition and prediction in smart home based on interaction," *Sensors*, vol. 19, no. 20, 2019.

[5] L. Schrader, A. Vargas Toro, S. Konietzny, S. Rüping, B. Schäpers, M. Steinböck, C. Krewer, F. Müller, J. Güttler, and T. Bock, "Advanced sensing and human activity recognition in early intervention and rehabilitation of elderly people," *J. Popul. Ageing*, vol. 13, pp. 139–165, 2020.

[6] J.-H. Hsiao, Y.-H. Deng, T.-Y. Pao, H.-R. Chou, and J.-Y. Chang, "Design of a wireless 3D hand motion tracking and gesture recognition glove for virtual reality applications," in *Inf. Stor. Proc. Syst.*, vol. 58103, 2017.

[7] S. K. Hiremath and T. Plötz, "The lifespan of human activity recognition systems for smart homes," *Sensors*, vol. 23, no. 18, 2023.

[8] S. K. Hiremath, Y. Nishimura, S. Chernova, and T. Plötz, "Bootstrapping human activity recognition systems for smart homes from scratch," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 6, no. 3, pp. 1–27, 2022.

[9] J. Han, Z. Zhang, C. Mascolo, E. André, J. Tao, Z. Zhao, and B. W. Schuller, "Deep learning for mobile mental health: Challenges and recent advances," *IEEE Signal Process. Mag.*, vol. 38, no. 6, pp. 96–105, 2021.

[10] R. M. French, "Catastrophic forgetting in connectionist networks," *Trends Cogn. Sci.*, vol. 3, no. 4, pp. 128–135, 1999.

[11] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychol. Learn. Motiv.* Elsevier, 1989, vol. 24, pp. 109–165.

[12] S. Tian, L. Li, W. Li, H. Ran, X. Ning, and P. Tiwari, "A survey on few-shot class-incremental learning," *Neural Netw.*, vol. 169, pp. 307–324, 2024.

[13] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "A continual learning survey: Defying forgetting in classification tasks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 7, pp. 3366–3385, 2021.

[14] Z. Chen and B. Liu, *Lifelong machine learning*. Morgan & Claypool Publishers, 2018.

[15] C. F. S. Leite and Y. Xiao, "Resource-efficient continual learning for sensor-based human activity recognition," *ACM Trans. Embed. Comput. Syst.*, vol. 21, no. 6, pp. 1–25, 2022.

[16] B. Kann, S. Castellanos-Paez, and P. Lalanda, "Evaluation of regularization-based continual learning approaches: Application to HAR," in *2023 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*. IEEE, 2023, pp. 460–465.

[17] Y. D. Kwon, J. Chauhan, A. Kumar, P. H. HKUST, and C. Mascolo, "Exploring system performance of continual learning for mobile and embedded sensing applications," in *2021 IEEE/ACM Symposium on Edge Computing (SEC)*. IEEE, 2021, pp. 319–332.

[18] Z. Qiao, Q. Pham, Z. Cao, H. H. Le, P. N. Suganthan, X. Jiang, and S. Ramasamy, "Class-incremental learning for time series: Benchmark and evaluation," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 5613–5624.

[19] R. Adaimi and E. Thomaz, "Lifelong adaptive machine learning for sensor-based human activity recognition using prototypical networks," *Sensors*, vol. 22, no. 18, 2022.

[20] M. Schiemer, L. Fang, S. Dobson, and J. Ye, "Online continual learning for human activity recognition," *Pervasive Mob. Comput.*, vol. 93, 2023.

[21] K. Fan, J. Li, S. Lai, L. Lv, A. Liu, J. Tang, H. H. Song, Y. Yue, and H. Zhuang, "TS-ACL: A time series analytic continual learning framework for privacy-preserving and class-incremental pattern recognition," *arXiv preprint arXiv:2410.15954*, 2024.

[22] Y. Wu, M. Nie, T. Zhu, L. Chen, H. Ning, and Y. Wan, "PTMs-TSCIL pre-trained models based class-incremental learning," *arXiv preprint arXiv:2503.07153*, 2025.

[23] X. Zhang, H. Yu, Y. Yang, J. Gu, Y. Li, F. Zhuang, D. Yu, and Z. Ren, "HarMI: Human activity recognition via multi-modality incremental learning," *IEEE J. Biomed. Health Inform.*, vol. 26, no. 3, pp. 939–951, 2021.

[24] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 2935–2947, 2017.

[25] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 114, no. 13, pp. 3521–3526, 2017.

[26] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, and T. Tuytelaars, "Memory aware synapses: Learning what (not) to forget," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 139–154.

[27] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," in *Int. Conf. Mach. Learn.* PMLR, 2017, pp. 3987–3995.

[28] D. Rolnick, A. Ahuja, J. Schwarz, T. Lillicrap, and G. Wayne, "Experience replay for continual learning," *Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.

[29] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[30] Y. D. Kwon, J. Chauhan, and C. Mascolo, "FastICARL: Fast incremental classifier and representation learning with efficient budget allocation in audio sensing applications," *arXiv preprint arXiv:2106.07268*, 2021.

[31] D. Shim, Z. Mai, J. Jeong, S. Sanner, H. Kim, and J. Jang, "Online class-incremental continual learning with adversarial shapley value," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 11, 2021, pp. 9630–9638.

[32] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[33] D. Anguita, A. Ghio, L. Oneto, X. Parra Perez, and J. L. Reyes Ortiz, "A public domain dataset for human activity recognition using smartphones," in *Proceedings of the 21th International European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, 2013, pp. 437–442.

[34] M. Zhang and A. A. Sawchuk, "USC-HAD: A daily activity dataset for ubiquitous activity recognition using wearable sensors," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, 2012, pp. 1036–1043.

[35] S. Kanoga, R. Karakida, T. Hoshino, Y. Okawa, and M. Tada, "Deep generative replay-based class-incremental continual learning in sEMG-based pattern recognition," in *2024 46th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2024.

[36] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, "Progressive neural networks," *arXiv preprint arXiv:1606.04671*, 2016.

[37] F. Matteoni, A. Cossu, C. Gallicchio, V. Lomonaco, and D. Bacciu, "Continual learning for human state monitoring," *arXiv preprint arXiv:2207.00010*, 2022.

[38] B. Kann, S. Castellanos-Paez, and P. Lalanda, "Cross-dataset continual learning: assessing pre-trained models to enhance generalization in HAR," in *2024 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*. IEEE, 2024, pp. 1–6.