Mixed-Up Experience Replay for Adaptive Online Condition Monitoring

Matthew Russell, *Student Member, IEEE*, Peng Wang, *Senior Member, IEEE*, Shaopeng Liu, and I. S. Jawahir

***Abstract*—Data-driven predictive maintenance reduces manufacturing downtime, and complex process-sensing relationships encourage the use of Deep Learning to automatically extract features. However, labeled training data is often lacking, and novel fault conditions may occur. Practical deployments must learn from unlabeled data, adapt to emerging conditions, and do so without prior knowledge of when the condition changes. Combining state-of-the-art Self-Supervised Learning (SSL) with Continual Learning (CL) facilitates adaptation as new conditions are observed. This study proposes a framework for adaptive online condition monitoring based on Barlow Twins SSL and novel Mixed-Up Experience Replay (MixER) for unsupervised CL. Tailored for 1D sensing data, Barlow Twins effectively clusters unlabeled data. When combined with MixER, the system outperforms state-of-the-art unsupervised CL on a motor health condition data set, reaching 92.4% classification accuracy. Future work will demonstrate human-in-the-loop integration for real manufacturing environments.**

***Index Terms*—Condition Monitoring, Continual Learning, Deep Learning, Predictive Maintenance**

I. Introduction

DATA-driven predictive models show promise for assessing machine health in real-time and enabling cost-saving Predictive Maintenance. Advances in Deep Learning (DL) can automatically extract valuable features from Condition Monitoring (CM) data when complex process-observation relationships are not well understood [1]. However, significant roadblocks hinder practical deployments since these models assume that test-time conditions match training conditions [2]. Training data largely comes from normal conditions, so deployed models often encounter unknown faults without ground truth labels [3], [4]. Considering these constraints, practical predictive models for CM must satisfy certain requirements:

R1) learn from unlabeled sensing observations,
R2) learn novel conditions not represented in the initial training data, and
R3) adapt to these unpredictable shifts in data on the fly.

Self-Supervised Learning (SSL) could alleviate R1 through specialized tasks with automatically generated labels [5]. These tasks are designed to extract information related to downstream applications. Simple approaches have included using autoencoders and statistical labels when missing ground truth labels [6], [7]. Using the more advanced Deep InfoMax [8] approach, Li et al. [9] learned features from 2D grayscale images derived from vibration signals, later fine-tuning the network with limited labeled data to classify bearing and gearbox faults. Alternatively, Wei et al. [10] used A Simple Framework for Contrastive Learning of Visual Representations (SimCLR) [11] to learn features from 1D vibration signals and diagnose bearing and cutting tool faults. However, this study relied on stacking 1D vibration data into 2D images instead of developing physically motivated approaches for 1D signals. Recognizing this, Peng et al. [12] applied SSL directly on vibration signals using Bootstrap Your Own Latent [13]. The positive results encourage further work to leverage state-of-the-art SSL like Barlow Twins [14] to realize R1 for CM.

Practical models must also distinguish novel conditions (i.e., emerging faults) not represented in the original training data set (R2). Zhang et al. [3] fit a Gaussian model to the latent features of an autoencoder and flagged out-of-distribution features at test time as a novel condition, but the model was inconsistent and could only identify a single new fault. Other methods have focused on detecting novel conditions within offline historical data. Li et al. [15] leveraged Domain Adversarial Transfer Learning to detect the an emerging fault in unlabeled target data. A follow-up work applied iterative K-means to count the number of emerging conditions [16]. Similarly, Chao et al. [4] trained a variational autoencoder on offline data and traded K-means for OPTICS, a density-based clustering algorithm, to identify novel faults. However, both studies required offline data sets and neither produced an updated model that could classify both old and new faults.

Incrementally learning tasks as described by R2 is called Continual Learning (CL). The goal of CL is to maximize performance across an ordered sequence of tasks processed one-by-one. This differs from multitask learning because the tasks must be learned sequentially instead of simultaneously and from transfer learning because performance must be maximized across all tasks instead of just the target task [17]. Existing work has demonstrated using Elastic Weight Consolidation [18]

Manuscript received Month xx, 2xxx; revised Month xx, xxxx; accepted Month x, xxxx. This work is supported by NSF Grant 2015889.

Matthew Russell is with the Department of Electrical and Computer Engineering, University of Kentucky, Lexington, KY 40506, USA (phone: 8595620181; e-mail: matthew.russell@uky.edu).

Peng Wang is with the Department of Electrical and Computer Engineering and the Department of Mechanical and Aerospace Engineering, University of Kentucky, Lexington, KY 40506, USA (phone: 8595622415; e-mail: edward.wang@uky.edu).

Shaopeng Liu is with GE Research, 1 Research Circle, Niskayuna, New York 12309, USA (e-mail: sliu@ge.com).

I. S. Jawahir is with the Institute for Sustainable Manufacturing and the Department of Mechanical and Aerospace Engineering, University of Kentucky, Lexington, KY 40506, USA (e-mail: is.jawahir@uky.edu)

or Memory Aware Synapses [19] to predict the health condition of turbofan engines across varying conditions [20], [21] and assess the quality of simulated products [22]. Xing et al. [23] took a related approach to transfer fault knowledge when process parameters changed, but all these methods violated R3 by requiring task change information (e.g., when an emerging fault occurs) and often violated R1 by requiring labeled data. Dark Experience Replay (DER) circumvented the former by maintaining a small buffer of past data (i.e., experience) and including (i.e., replaying) it in the training objectives alongside new data to avoid forgetting [24]. Lifelong Unsupervised Mixup (LUMP) combined SSL and DER for unsupervised CL capable of satisfying R1, R2, and R3 but has not yet been tested for CM [25].

Building on advancements in SSL and CL, this study proposes a novel framework for adaptive online CM that applies Barlow Twins to 1D time series data from rotating machinery and invents Mixed-Up Experience Replay (MixER) that extends LUMP. This paper's contributions are threefold:

1) the novel application of Barlow Twins SSL to CM data with new time series augmentations for unlabeled 1D signals to achieve R1,
2) a novel adaptive online CM architecture and pipeline that proposes Mixed-Up Experience Replay (MixER) for CL to achieve R2 and R3, and
3) experiments validating MixER by comparing clustering performance of state-of-the-art unsupervised CL approaches on unlabeled data from sequentially observed motor health conditions.

In the remainder of this paper, Section II presents related work in SSL and CL, Section III presents the proposed methodology, Section IV describes validation experiments, Section V discusses the results, and Section VI offers concluding thoughts.

II. Related Work

The proposed adaptive online CM framework draws from previous SSL and CL literature.

A. Self-Supervised Learning

Self-Supervised Learning (SSL) learns representations from unlabeled data through pretext tasks [26], [27] or domain-specific data augmentation. For the latter, SSL treats random augmentations of the same data example as a single pseudoclass and learns to map them to similar features [28]. Effective augmentations randomize unimportant attributes without destroying semantic information. Many approaches rely on contrastive loss that consolidates features from the same pseudoclass while distancing features from different pseudoclasses [5], [8]. Notable attempts to address difficulties with contrastive learning include Momentum Contrast [29], its derivatives [11], [13], and Simple Siamese Representation Learning [30]. To circumvent the drawbacks altogether, Barlow Twins [14] replaced contrastive loss with cross-correlation loss that encouraged each dimension of a feature projection to be independent from the rest but correlated with itself across augmentations of the same data example. This achieved state-of-the-art performance without specialized network layers or training needed by contrastive learning.

B. Continual Learning Strategies

While Barlow Twins could satisfy R1 of adaptive online CM, R2 and R3 motivate exploration of Continual Learning (CL). Popular approaches like Learning without Forgetting [31], Elastic Weight Consolidation [18], Memory Aware Synapses [19], and Incremental Classification and Representation Learning [32] performed incremental learning at discrete task boundaries and did not support continuously changing data. In contrast, Dark Experience Replay (DER) opted to train continuously while using a fixed-length set of randomly saved examples to capture past experience [24]. Reservoir sampling filled the buffer uniformly throughout the learning history [33]. This experience was then replayed in the loss function to mitgate forgetting. The DER loss function was

$$\mathcal{L}_{\text{DER}} = \mathcal{L}_{\text{sup}}(x, y) + \alpha \| g_\phi(\tilde{x}) - \tilde{z} \|_2^2 \tag{1}$$

where $\mathcal{L}_{\text{sup}}(x, y)$ was the supervised loss on new examples, $g_\phi(\tilde{x})$ was the classifier's output logits for buffer example $\tilde{x}$, and $\tilde{z}$ was the original logits for $\tilde{x}$. The hyperparameter α controlled the strength of regularization preventing the classifier from changing its predictions for past experience. The authors also proposed extending DER to DER++ with another loss term to promote memory of past experience:

$$\mathcal{L}_{\text{DER++}} = \mathcal{L}_{\text{sup}}(x, y) + \alpha \| g_\phi(\tilde{x}) - \tilde{z} \|_2^2 + \beta \mathcal{L}_{\text{sup}}(\tilde{x}, \tilde{y}) \tag{2}$$

If the data distribution changes significantly, forcing the model to match its old predictions may be counterproductive [24]. There could be another way to arrive at the same class prediction that better supports the distribution shift. Thus, the DER++ loss function included teaching the model past experience with ground truth labels $\tilde{y}$. Training with the ground truth labels $\tilde{y}$ could cause overfitting to past experience [34], but combining this with the α term can improve performance.

However, DER and DER++ did not support unlabeled data. In response, Madaan et al. [25] adapted DER to unlabeled data by replacing the classification task with SSL. To further improve performance, the authors proposed Lifelong Unsupervised Mixup (LUMP) that linearly blended examples of new data with experience from the replay buffer. LUMP was implemented by modifying the DER loss function (1):

$$\mathcal{L}_{\text{LUMP}} = \mathcal{L}_{\text{SSL}}(x') + \alpha \| g_\phi(\tilde{x}) - \tilde{h} \|_2^2 \tag{3}$$

$\mathcal{L}_{\text{SSL}}$ is the SSL loss (e.g., SimSiam or Barlow Twins) on augmentations of a "mixed-up" seed input $x' = \lambda x + (1 - \lambda)\tilde{x}$ where x is the batch from the current task, and λ is the mixing hyperparameter. Although λ could be sampled from a Beta distribution, the experiments used a fixed value. The second term encourages the model to preserve the same projections $\tilde{h}$ for experience $\tilde{x}$ in the replay buffer. With state-of-the-art results for image classification, LUMP introduced a promising solution for adaptive online CM.

III. Proposed Method for Adaptive Online Condition Monitoring

This study presents a novel pipeline for learning emerging faults from unlabeled sensing data through the proposed Mixed-Up Experience Replay (MixER) with Barlow Twins.

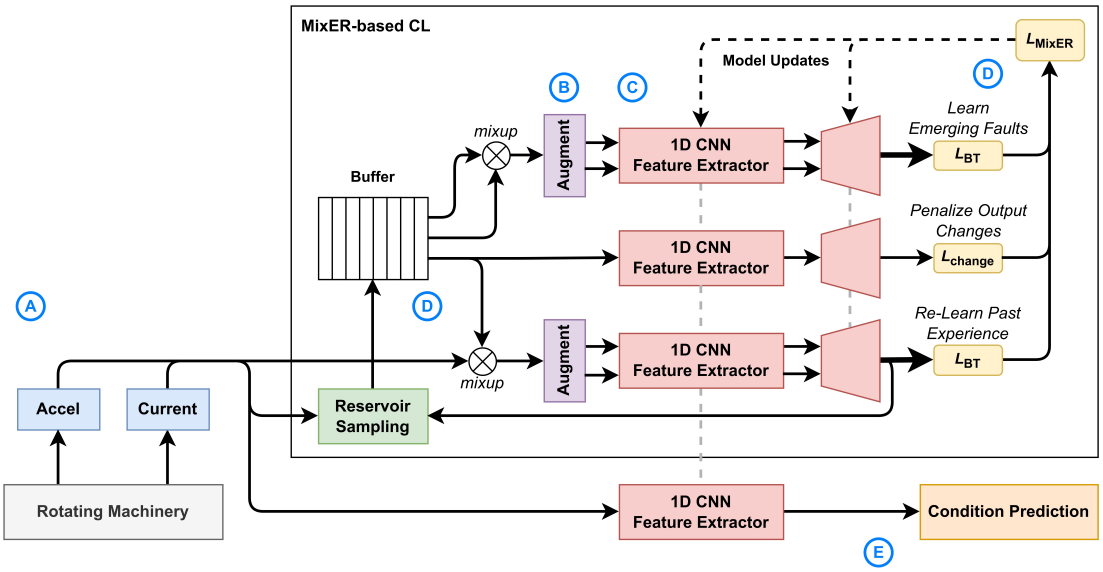

Fig. 1. The proposed system architecture for CL to monitor machine health. Feature extractors connected by dashed lines use the same weights.

TABLE I
AUGMENTATION TRANSFORMATIONS

Random Flip	`x = -x if rand() < 0.5 else x`
Random Scaling	`k = randuniform(0.1, 1.0)` `x = k * x`
Random Jitter	`n = randint(0, x.shape[-1])` `x = concat(x[:,n:], x[:,:n])`
Random Masking	`n = randint(0, x.shape[-1] - 64)` `x[:,n:n + 64] = 0`

A. Proposed System for Continual Learning

Fig. 1 shows a framework for adaptive online CM using Mixed-Up Experience Replay (MixER) with Barlow Twins SSL for CL on unsupervised time series signals. Acceleration and current sensors collect high-frequency time series signals from rotating machinery for the predictive model (A in Fig. 1).

The framework applies simple transformations (B in Fig. 1) to diversify the signals without destroying important semantic (i.e., condition) information. The augmentations consist of random flipping, scaling with a factor from 0.1 to 1.0, jitter up to the length of the window, and masking (zeroing) a random 64-point section of the signal (see Table I). From a physical standpoint, these transformations randomize amplitude and phase to obscure time series artifacts while preserving frequency content important for rotating machinery signals.

Augmented examples are passed to Barlow Twins SSL (C in Fig. 1) with three 1D CNN residual blocks (see Fig. 2) that extract features while improving backward gradient flow versus regular CNN layers. A projector embeds the features into a metric space where cross-correlation loss is applied. Representing the CNN encoder as f_θ with parameters θ and projection head g_ϕ with parameters ϕ, the projections from batch $\mathbf{x} \in \mathbb{R}^{N\times L}$ of N examples are $\mathbf{h} = g_\phi(f_\theta(\mathbf{x}))$. The normalized values are $\hat{\mathbf{h}} = (\mathbf{h} - \bar{\mathbf{h}})/\sigma_{\mathbf{h}}$ where $\bar{\mathbf{h}}$ and $\sigma_{\mathbf{h}}$ are the mean and standard deviation of each feature across the batch. By augmenting the input twice, the model produces two projections $\hat{\mathbf{h}}_a, \hat{\mathbf{h}}_b \in \mathbb{R}^{N\times M}$, and calculates $M \times M$ cross-correlation matrix:

$$R = \hat{\mathbf{h}}_a^\top \hat{\mathbf{h}}_b / N \tag{4}$$

The Barlow Twins loss function is

$$\mathcal{L}_{\text{BT}} = \sum_i (1 - R_{ii})^2 + \gamma \sum_i \sum_{j\neq i} R_{ij}^2 \tag{5}$$

where the first term is an "invariance term" that encourages similar features from the same seed example, and the second term is a "redundancy reduction term" (with scaling factor γ) that encourages independence among features [14].

To adapt the representation to emerging faults, this study proposes the novel MixER algorithm (D in Fig. 1). MixER combines LUMP with DER++, training the Barlow Twins model on mixed-up examples of both emerging faults *and* past experience. Mixing up past experience improves data diversity to prevent overfitting. The machine condition can then be classified for predictive maintenance (E in Fig. 1).

B. Mixed-Up Experience Replay (MixER)

LUMP combined Barlow Twins with DER, mixing new examples with those from the experience replay buffer, but left out the DER++ β objective. The proposed Mixed-Up Experience Replay (MixER) demonstrates using this β term on unlabeled experience with linear mixup:

$$\mathcal{L}_{\text{MixER}} = \underbrace{\mathcal{L}_{\text{BT}}(x')}_{\substack{\text{learn emerging}\\ \text{faults}}} + \underbrace{\alpha \|g_\phi(f_\theta(\tilde{x})) - \tilde{h}\|_2^2}_{\substack{\text{penalize changes in}\\ \text{model behavior}}} + \underbrace{\beta \mathcal{L}_{\text{BT}}(\tilde{x}')}_{\substack{\text{re-learn past}\\ \text{experience}}} \tag{6}$$

$\mathcal{L}_{\text{BT}}(x')$ is Barlow Twins cross-correlation loss on $x' = \lambda x + (1-\lambda)\tilde{x}$ using past experience $\tilde{x}$ (like LUMP). Unlike Lump, MixER adds the β term in which $\mathcal{L}_{\text{BT}}(\tilde{x}')$ is Barlow Twins loss on $\tilde{x}' = \lambda_2 \tilde{x}_a + (1-\lambda_2)\tilde{x}_b$ (i.e., linear mixup of examples from the experience replay buffer with ratio λ_2). While LUMP uses a constant λ, MixER randomly samples the mixing ratio

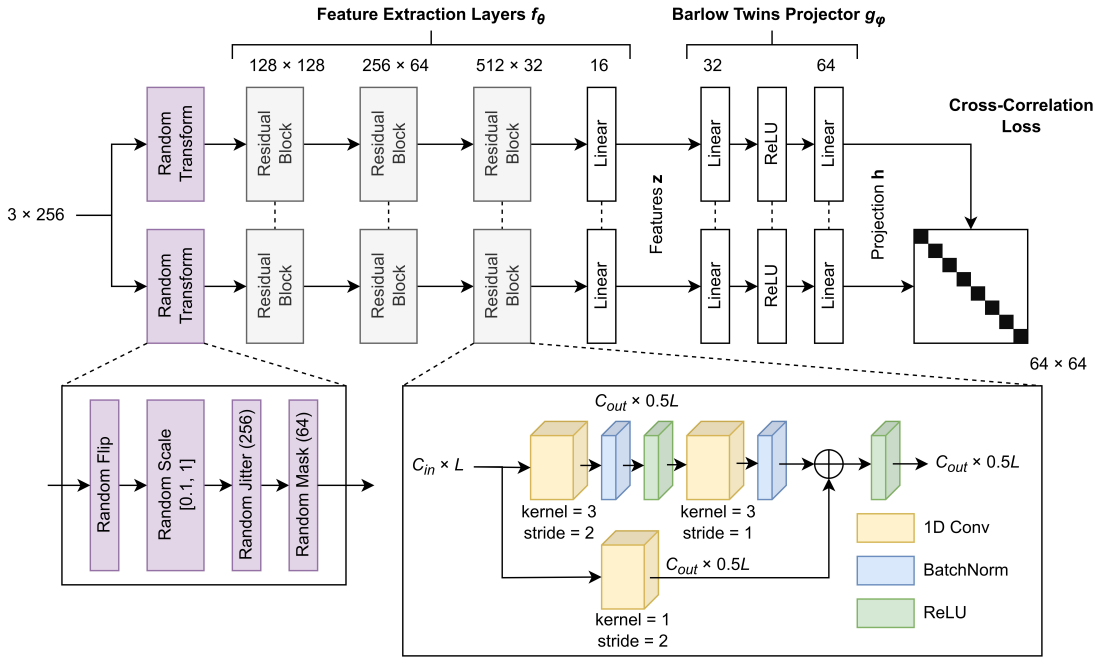

Fig. 2. (Top) data augmentation, feature extraction layers, and projector in the Barlow Twins model; (Bottom Left) data augmentation sequence; (Bottom Right) 1D convolutional residual block of the feature extractor

λ_2 from a Beta distribution: $\lambda_2 \sim \text{Beta}(\nu, \nu)$. The first term allows the model learn emerging faults, the second penalizes changes in the model's behavior to prevent forgetting, and the third enables the model to re-learn past experience. The β term explicitly reminds the model of past experience, and the α term prevents overfitting to specific historical examples.

C. Adaptive Online Condition Monitoring with MixER

Fig. 3 surveys a data processing pipeline for adaptive online CM using the proposed framework. While the model could be pretrained offline, it can also be deployed without prior training. Once features are extracted, MixER loss provides the gradients to refine the representation via Barlow Twins on unlabeled data (R1) without forgetting previous conditions (R2). Reservoir sampling curates the experience replay buffer of 100 256-point windows without needing condition change points (R3). The system flags new feature clusters that could indicate emerging faults and enables experts to decide if they are a fault or a benign change in process parameters. Within this framework, this study seeks to validate MixER-based CL.

IV. Experiments

Since MixER depends on Barlow Twins, experiments are designed to verify Barlow Twins can learn from unlabeled time series data containing all eight operating conditions. Subsequently, CL experiments with emerging faults compare the feature representations produced by DER, DER++, LUMP, and MixER from unlabeled data.

A. Motor Condition Data Set

Vibration and current data were collected from eight motor conditions using the SpectraQuest Machinery Fault Simulator (MFS) in Fig. 4. The conditions were normal operation (N), faulted bearings (FB), bowed rotor (BoR), broken rotor (BrR), misaligned rotor (MR), unbalanced rotor (UR), phase loss (PL), and unbalanced voltage (UV). The MFS was run at 2000 RPM and 3000 RPM with 0.06 Nm and 0.7 Nm loads. Data from vertically and horizontally mounted accelerometers and a current clamp were sampled at 12 kHz during steady state operation for 60 seconds. Each signal was normalized to [-1, 1] and split into 256-point windows.

B. Experimental Objectives

Before simulating emerging faults, experiments must verify that Barlow Twins with the augmentations from Table I is suitable for CM time series data. Training Barlow Twins with all the motor condition data at once establishes a baseline upper limit of performance. The experiments assess representation quality via the accuracy of a linear classifier (e.g., a single fully-connected neural network layer with no activation function), following standard practice in the literature [14], [25], [30]. This linear classifier is not part of the proposed framework but serves as a consistent way to evaluate features. Five SSL variations are tested: one with all augmentations and four that each drop out a different one. If performance decreases when one is removed, this could reveal what physically meaningful information the Barlow Twins model learns.

Next, emerging fault experiments evaluate the ability of DER, DER++, LUMP, and MixER to replicate the baseline performance in the more difficult CL situation. To demonstrate catastrophic forgetting, a Fine-Tune model is trained without using experience replay or mixup. All examples from the same condition are grouped together to create eight equally sized, homogeneous data sets of 11248 examples subdivided

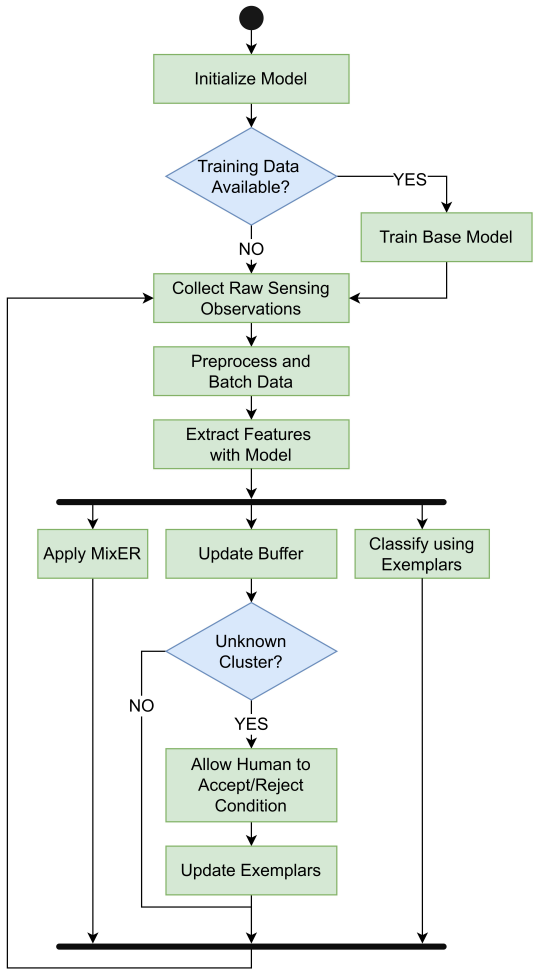

Fig. 3. Flowchart of data processing pipeline for adaptive online CM

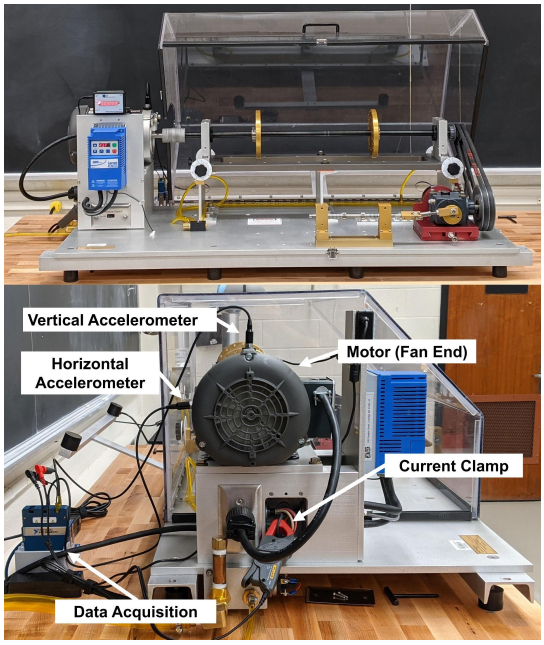

Fig. 4. SpectraQuest Machinery Fault Simulator (MFS)

TABLE II
HYPERPARAMETER SEARCH RANGES

Hyperparameter	Values
η (learning rate)	[0.0005, 0.001, 0.002, 0.01]
α (penalize output changes)	[0.1, 0.5, 1.0]
β (re-learn past experience)	[0.5, 1.0, 2.0, 5.0]
λ (LUMP mixup)	[0.1, 0.4, 0.8]
ν (MixER Beta distribution)	[0.5, 1.0, 2.0]

TABLE III
SELECTED HYPERPARAMETER VALUES

Method	η	α	β	λ	ν
Fine-Tune	0.0005	–	–	–	–
DER	0.0005	0.1	–	–	–
DER++	0.001	0.1	0.5	–	–
LUMP	0.001	0.5	–	0.4	–
MixER	0.001	0.1	1.0	0.4	1.0

into 80%/10%/10% splits for training, validation, and testing, respectively. These eight tasks are then randomly ordered since some orders may be easier to learn than others. Each experiment starts by training the model on the first task's data (containing a single motor condition) without labels, simulating R1. Training continues for 50 epochs to allow the feature representation to converge [24], [25]. The first task's data set is then replaced with that of the second task (also containing a single condition), and training resumes without notifying the model of the change. This simulates the occurence of a new, unknown health condition as described by R2 and R3. After 50 epochs the data set is again replaced, and this continues until all eight tasks have been shown to the model. After each task, the model is evaluated on all conditions observed to that point in training. For all experiments, the batch size is 256, and the experience replay buffer holds 100 examples that are curated via reservoir sampling.

C. Experimental Implementation

All models and experiments are developed in PyTorch. When testing Barlow Twins without emerging faults, each of the five combinations of augmentations is repeated five times with different random seeds. The Barlow Twins model is trained with the Adam optimizer and learning rate of 0.0002 for 100 epochs on data from all eight motor conditions. The subsequent CL experiments contain several hyperparameters to control regularization for DER, DER++, LUMP, and MixER (i.e., α, β, λ, and ν). Hyperparameter values and learning rate are selected through a grid search with 10% of the training data (see Table II). Performance is scored via validation accuracy of a linear classifier, and the resulting values are shown in Table III. With these hyperparameters, each CL model is trained 25 times: five random seeds for each of five random orderings (see Table IV) of the eight motor conditions. Experiments utilize an NVIDIA P100 GPU to accelerate training.

V. RESULTS AND DISCUSSION

After training, the feature extractors are frozen before assessing representation quality with a linear classifier.

TABLE IV
CONDITION ORDERINGS

	Task 1	2	3	4	5	6	7	8
1	PL	FB	N	UR	BoR	BrR	MR	UV
2	UV	N	MR	FB	BoR	UR	PL	BrR
3	BrR	N	UR	BoR	FB	MR	PL	UV
4	FB	BrR	UR	BoR	UV	N	PL	MR
5	N	BoR	MR	UV	FB	UR	BrR	PL

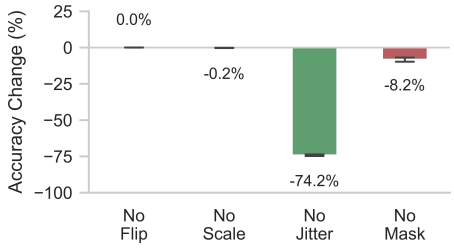

Fig. 5. Effect on accuracy of removing augmentation transformations

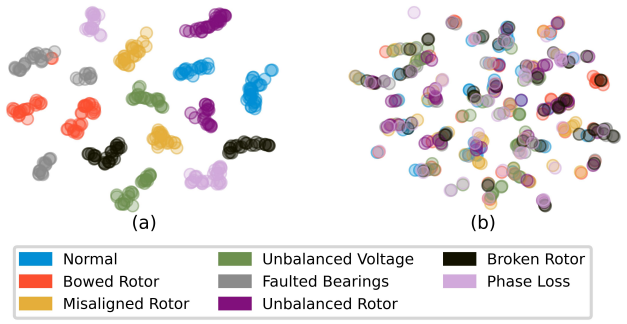

Fig. 6. Barlow Twins features (t-SNE) with (a) and without jitter (b)

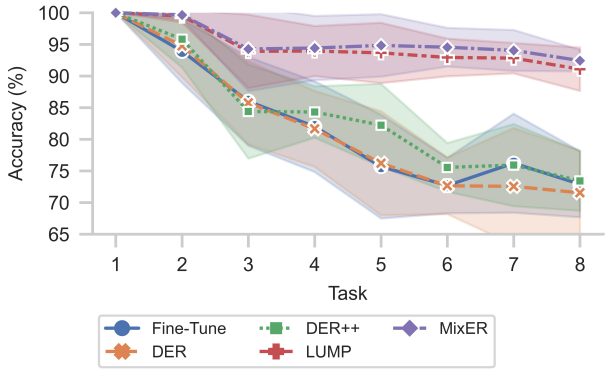

Fig. 7. Test set accuracy on incrementally added conditions

A. Barlow Twins with Time Series Transforms

The linear classifier reaches 99.6±0.0% accuracy on the features produced by the Barlow Twins model trained on all eight motor conditions (i.e., no simulation of emerging faults). This indicates that Barlow Twins can learn meaningful representations of vibration data when using the augmentations in Table I. Fig. 5 shows the change in accuracy when individual transformations are removed. Accuracy remains high without the random flip or scaling and falls less than 10 points without masking. However, it drops more than 74 points when jitter is removed due to poor clustering of features from the same motor condition (see Fig. 6). From a physical understanding, jitter randomizes the time offset of each window while preserving the frequency content. These offsets are artifacts of data windowing and not related to the health condition. Nevertheless, to data-driven models these coincidental time-domain patterns can dominate clustering and obfuscate underlying condition information. Randomizing the offsets through jitter forces the model to learn more semantically meaningful features. Although the t-SNE visualization does not show one cluster per condition, the underlying representation has 16 features and sufficiently avoids overlapping conditions for the linear classifier to achieve high accuracy.

B. Continual Learning of Emerging Faults with MixER

Table V shows the performance of CL methods in the emerging fault simulations with variability caused by random model initialization and task orderings. Fig. 7 shows that Fine-Tune degrades in performance on each successive task without a strategy to avoid catastrophic forgetting. DER and DER++ worked well on supervised problems in the literature but perform poorly in this unsupervised case (73.4±4.7% for DER++). LUMP sharply increases accuracy to 91.1±3.4%. Since LUMP simply adds mixup to DER, this indicates that mixup is likely the dominant reason for LUMP's gains over DER. Under this assumption, the proposed method of adding more advanced mixup to DER++ to create MixER should outperform LUMP. MixER increases accuracy to 92.4±1.7% after Task 8 and outperforms LUMP on all intermediate tasks. Fig. 8 shows the evolution of MixER's latent representation. The latent space remains well-structured compared to DER++. A statistical significance test value of 0.042 indicates a low probability of observing these results if MixER was not actually different from LUMP (considered moderately significant). Thus, the proposed MixER method outperforms the state-of-the-art by adding mixup to DER++ and sampling the mixing parameter from a Beta distribution. Mixup plays an important role by diversifying past experience and fusing it with current observations. These initial results demonstrate that MixER exceeds state-of-the-art CL performance.

C. Evaluation of the Adaptive Online CM Framework

The experiments validate the core CL aspect of the adaptive online CM framework from Section III. Despite learning conditions sequentially, MixER falls only 7.2 points below the best case Barlow Twins model trained on all conditions at once. Furthermore, MixER only requires 6.7 min to train for 50 epochs (8.1 sec/epoch) on 48 sec of training data per condition, and additional optimization could further reduce this. This study focuses on learning quality features (assessed via a supervised linear classifier), and future work can develop human-in-the-loop classification approaches on the clusters in Fig. 8 to complete end-to-end validation of the adaptive

TABLE V
LINEAR CLASSIFIER ACCURACY AFTER EACH EMERGING FAULT

	Linear Classifier Accuracy (%)							
	Task 1†	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8
Fine-Tune	100±0.0	93.9±5.0	86.0±7.0	82.0±7.2	75.7±8.2	72.7±4.4	76.2±7.8	72.9±5.2
DER	100±0.0	94.7±4.9	85.8±6.6	81.6±6.0	76.2±8.2	72.6±4.5	72.5±9.2	71.5±6.7
DER++	100±0.0	95.8±4.5	84.4±7.4	84.3±4.1	82.2±6.6	75.5±3.9	75.9±6.5	73.4±4.7
LUMP	100±0.0	99.5±1.0	93.9±5.8	93.9±4.0	93.6±4.8	92.9±3.0	92.8±2.4	91.1±3.4
MixER (proposed)	100±0.0	**99.6±0.6**	**94.3±6.7**	**94.4±5.1**	**94.8±4.9**	**94.5±3.1**	**94.0±3.2**	***92.4±1.7**

† All methods score 100% on Task 1 since there is only a single condition to classify
*Welch's t-test value vs. LUMP = 0.042

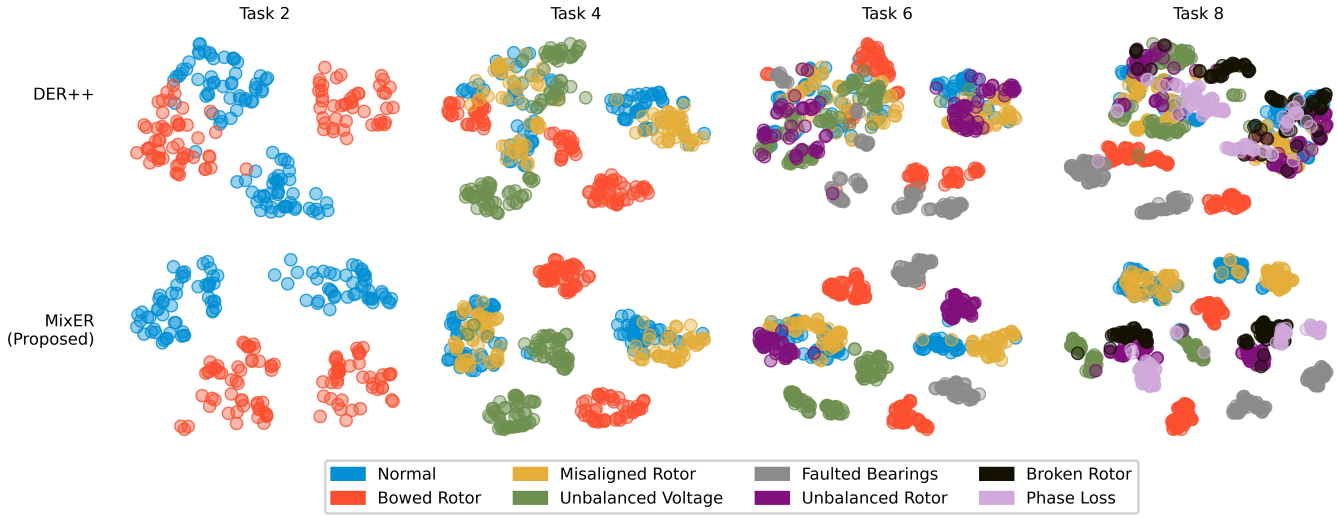

Fig. 8. t-SNE plots of DER++ (top) and MixER (bottom) features as classes are incrementally observed (left to right)

online CM framework. Importantly, the effectiveness of MixER without pretraining indicates that the proposed framework can be deployed with no *a priori* data collection.

VI. CONCLUSION

Adaptive online CM depends on learning from unlabeled sensing data (R1) and adapting to emerging faults (R2) that occur unpredictably (R3). This study contributes a novel application of Barlow Twins to address R1 for unlabeled 1D sensing data and achieves 99.6% linear classification accuracy. Furthermore, the MixER proposal shows 92.4% accuracy with a linear classifier after sequentially observing the motor conditions (addressing R2 and R3). Future work can implement human-in-the-loop, exemplar-based classification to validate model decisions according to the proposed framework.

ACKNOWLEDGMENT

We would thank the University of Kentucky Center for Computation Sciences and Information Technology Services Research Computing for their support and use of the Lipscomb Compute Cluster and associated research computing resources.

REFERENCES

[1] J. Wang, Y. Ma, , L. Zhang, R. X. Gao, and D. Wu, "Deep learning for smart manufacturing: Methods and applications," *J. Manuf. Syst.*, vol. 48, pp. 144–156, 2018.

[2] Y.-H. Lin and L. Chang, "An online transfer learning framework for time-varying distribution data prediction," *IEEE Trans. Ind. Electron.*, vol. 69, no. 6, pp. 6278–6287, 2021.

[3] S. Zhang, M. Wang, W. Li, J. Luo, and Z. Lin, "Deep learning with emerging new labels for fault diagnosis," *IEEE Access*, vol. 7, pp. 6279–6287, 2018.

[4] M. A. Chao, B. Adey, and O. Fink, "Implicit supervision for fault detection and segmentation of emerging fault types with deep variational autoencoders," *Neurocomputing*, vol. 454, pp. 324–338, 2021.

[5] P. Bachman, R. D. Hjelm, and W. Buchwalter, "Learning representations by maximizing mutual information across views," in *Adv. Neural Inf. Process. Syst.*, vol. 32, pp. 15 535–15 545, 2019.

[6] W. Zhang, D. Chen, and Y. Kong, "Self-supervised joint learning fault diagnosis method based on three-channel vibration images," *Sensors*, vol. 21, no. 14, p. 4474, 2021.

[7] T. Zhang, J. Chen, S. He, and Z. Zhou, "Prior knowledge-augmented self-supervised feature learning for few-shot intelligent fault diagnosis of machines," *IEEE Trans. Ind. Electron.*, vol. 69, no. 10, pp. 10 573–10 584, 2022.

[8] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio, "Learning deep representations by mutual information estimation and maximization," in *7th ICLR*, 2019.

[9] G. Li, J. Wu, C. Deng, M. Wei, and X. Xu, "Self-supervised learning for intelligent fault diagnosis of rotating machinery with limited labeled data," *Applied Acoustics*, vol. 191, p. 108663, 2022.

[10] M. Wei, Y. Liu, T. Zhang, Z. Wang, and J. Zhu, "Fault diagnosis of rotating machinery based on improved self-supervised learning method and very few labeled samples," *Sensors*, vol. 22, no. 1, p. 192, 2022.

[11] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of the 37th ICML*, pp. 1597–1607, 2020.
[12] T. Peng, C. Shen, S. Sun, and D. Wang, "Fault feature extractor based on bootstrap your own latent and data augmentation algorithm for unlabeled vibration signals," *IEEE Trans. Ind. Electron.*, vol. 69, no. 9, pp. 9547–9555, 2022.
[13] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, B. Piot, k. kavukcuoglu, R. Munos, and M. Valko, "Bootstrap your own latent: A new approach to self-supervised learning," in *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 21 271–21 284, 2020.
[14] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, "Barlow twins: Self-supervised learning via redundancy reduction," in *Proceedings of the 38th ICML*, pp. 12 310–12 320, 2021.
[15] J. Li, R. Huang, G. He, S. Wang, G. Li, and W. Li, "A deep adversarial transfer learning network for machinery emerging fault detection," *IEEE Sens. J.*, vol. 20, no. 15, pp. 8413–8422, 2020.
[16] J. Li, R. Huang, G. He, Y. Liao, Z. Wang, and W. Li, "A two-stage transfer adversarial network for intelligent fault diagnosis of rotating machinery with multiple new faults," *IEEE/ASME Trans. Mechatron.*, vol. 26, no. 3, pp. 1591–1601, 2021.
[17] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proc. IEEE*, vol. 109, no. 1, pp. 43–76, 2020.
[18] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barinska, D. Hassabis, C. Clopath, D. Kurmaran, and R. Hadsell, "Overcoming catastrophic forgetting in neural networks," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 114, no. 13, pp. 2531–3526, 2017.
[19] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, and T. Tuytelaars, "Memory aware synapses: Learning what (not) to forget," in *ECCV 2018*, pp. 144–161, 2018.
[20] B. Maschler, H. Vietz, N. Jazdi, and M. Weyrich, "Continual learning of fault prediction for turbofan engines using deep learning with elastic weight consolidation," in *25th IEEE ETFA*, pp. 959–966, 2020.
[21] L. Yang, L. Wang, Z. Zheng, and Z. Zhang, "A continual learning-based framework for developing a single wind turbine cybertwin adaptively serving multiple modeling tasks," *IEEE Trans. Ind. Informat.*, vol. 18, no. 7, pp. 4912–4921, 2022.
[22] H. Tercan, P. Deibert, and T. Meisen, "Continual learning of neural networks for quality prediction in production using memory aware synapses and weight transfer," *J. Intell. Manuf.*, vol. 33, pp. 283–292, 2022.
[23] S. Xing, Y. Lei, B. Yang, and N. Lu, "Adaptive knowledge transfer by continual weighted updating of filter kernels for few-shot fault diagnosis of machines," *IEEE Trans. Ind. Electron.*, vol. 69, no. 2, pp. 1968–1976, 2022.
[24] P. Buzzega, M. Boschini, A. Porrello, D. Abati, and S. Calderara, "Dark experience for general continual learning: a strong, simple baseline," in *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 15 920–15 930, 2020.
[25] D. Madaan, J. Yoon, Y. Li, Y. Liu, and S. J. Hwang, "Representational continuity for unsupervised continual learning," in *10th ICLR*, 2022.
[26] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," in *6th ICLR*, 2018.
[27] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv:1807.03748 [cs.LG]*, 2018.
[28] A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, and T. Brox, "Discriminative unsupervised feature learning with convolutional neural networks," in *Adv. Neural Inf. Process. Syst.*, vol. 27, pp. 766–774, 2014.
[29] K. He, H. Fan, Y. Wu, S. Xie, and R. B. Girshick, "Momentum contrast for unsupervised visual representation learning," in *IEEE/CVF CVPR*, pp. 9726–9735, 2020.
[30] X. Chen and K. He, "Exploring simple siamese representation learning," in *IEEE/CVF CVPR*, pp. 15 750–15 758, 2021.
[31] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 2935–2947, 2017.
[32] S. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "iCaRL: Incremental classifier and representation learning," in *IEEE/CVF CVPR*, pp. 5533–5542, 2017.
[33] J. S. Vitter, "Random sampling with a reservoir," *ACM Trans. Math. Softw.*, vol. 11, no. 1, pp. 37–57, 1985.
[34] D. Lopez-Paz and M. A. Ranzato, "Gradient episodic memory for continual learning," in *Adv. Neural Inf. Process. Syst.*, vol. 30, pp. 6470–6479, 2017.

Matthew Russell graduated in 2018 with a B.S. in computer engineering from Liberty University, Lynchburg, VA, USA and is a Ph.D. candidate at the University of Kentucky, Lexington, KY, USA.

He has worked six tours as a Graduate Pathways Intern at NASA Johnson Space Center (Houston, TX, USA) developing software for crewed spaceflight including Orion, Artemis, and the Gateway Lunar Orbiting Platform. He is currently a Research Assistant in the Augmented Intelligence for Smart Manufacturing (AISM) lab at the University of Kentucky, Lexington, KY, USA.

Peng Wang is an assistant professor at the Department of Electrical and Computer Engineering and Department of Mechanical and Aerospace Engineering at the University of Kentucky. He received his B.S. and M.S. degrees in Information Science from Beijing University of Chemical Engineering in 2010 and 2013, and his Ph.D. degree in Mechanical and Aerospace Engineering from Case Western Reserve University in 2017, respectively. His research interests are in the areas of stochastic modeling, machine learning, and data fusion for machine performance prediction and uncertainty analysis, manufacturing process modeling, and human-robot collaboration.

Shaopeng Liu received the B.S. and M.E. degrees in mechanical engineering from Tsinghua University, Beijing, China in 2004 and 2007 respectively, and received his Ph.D. degree in mechanical engineering from the University of Connecticut in 2012. He joined GE Research in Niskayuna, NY in 2012, and is currently a Senior Scientist Industrial AI in the Digital Research group. His research interests include sensor applications, complex system design, edge computing, software and advanced analytics, and artificial intelligence (AI), with a focus on developing AI-enabled systems and technologies for diverse industrial applications in the fields of renewable energy, aviation, advanced manufacturing, healthcare, life sciences, and information systems.

I. S. Jawahir is the James F. Hardymon Endowed Chair in Manufacturing Systems and the Founding Director of the Institute for Sustainable Manufacturing (ISM) at the University of Kentucky. Prof. Jawahir's current research activities include sustainable product design and manufacturing processes. Prof. Jawahir is a Fellow of CIRP, ASME, and SME. He is the Founding Editor-in-Chief of the International Journal of Sustainable Manufacturing and the Technical Editor of the Journal of Machining Science and Technology.