Sleep Stage Classification with Learning from Evolving Datasets

Huayu Li, Xiwen Chen, Gregory Ditzler, William D.S. Killgore, Stuart F. Quan, Janet Roveda, and Ao Li

***Abstract*— The precise measurement of sleep stages is crucial for evaluating sleep quality. Deep learning has recently been utilized for automatic sleep stage classification, demonstrating exceptional performance. Previous studies on deep learning-based sleep stage classification assumed stationary data generation environments, where samples were drawn from fixed – albeit unknown – unknown distributions and annotated based on predefined criteria. However, this assumption of a stationary distribution is no longer valid in real-world applications due to shifts in data distribution between new and old datasets and changes in classification tasks. Moreover, the unavailability of historical data often poses challenges in training deep learning models. Sleep stage classification faces challenges associated with evolving data acquisition, changing patient demographics, and shifting annotation criteria over time. This paper addresses the classification of sleep stages with varying data distributions, missing historical datasets, and changing label granularity for the first time. We proposed learning strategies for addressing the challenges described above, as well as constructed benchmarks for evaluating the proposed learning strategies. The results demonstrate the effectiveness and performance of the proposed learning strategies. These findings provide compelling evidence for the significance and impact of this work. Furthermore, a comprehensive discussion is presented, highlighting the limitations of our approach, and proposing several avenues for future research.**

***Index Terms*— Continual Learning, Deep Learning, Generative Model, Sleep Stage Classification**

I. Introduction

Sleep is crucial to maintaining physical and mental health [1]. Monitoring and classifying sleep stages are critical for diagnosing sleep disorders and creating personalized treatment plans. Traditionally, trained clinicians classified sleep stages using electroencephalography (EEG) signals. Unfortunately, this manual process of labeling EEG signals is time-consuming and prone to inter-rater variability. In recent years, deep learning models have emerged as a promising approach for automatic sleep stage classification [2]–[4]. These models have successfully automatically classified sleep stages using raw EEG signals. They can accurately classify sleep stages in real-time and provide valuable insights into sleep patterns and disorders. Moreover, deep learning models can be trained on large volumes of data which typically corresponds to better performance. Despite the significant progress in applying deep learning models for automatic sleep stage classification, new challenges have emerged in the dynamic nature of datasets where the availability of historical data, the updating of data acquisition process, and the evolution of sleep stage annotations pose significant obstacles.

Consider the following scenario: A company collaborates with a hospital and acquires access to a sleep EEG dataset annotated with sleep/wake stages. The company develops a mobile application using a deep learning model to classify asleep and awake stages. The model's high accuracy provides valuable insights into patients' sleep patterns and enables personalized treatment plans. As time passes, the company receives new data annotated with five sleep stages from a different collaborator. The new data are expected to further the performance. The company recognizes the need to train a new – more complex – network that can accurately classify patients' sleep stages into these five categories. Nevertheless, the company faces a significant challenge in retaining the old dataset due to several factors, such as the end of the collaboration, data privacy laws, and data scarcity. The shift between the old and new dataset distributions may also lead the network trained on the new dataset to perform poorly on the patients' EEG collected from the old devices. Meanwhile, the old dataset annotated with only sleep/wake stages would limit the effective training of the new network even if the old dataset is obtained.

We formalize the challenges discussed above as sleep stage classification with learning from evolving datasets. The aim of this study is to investigate effective learning strategies and solutions to address the challenges associated with automatic sleep stage classification under varying data distribution, evolving label granularity, and inaccessibility of historical data. To begin, we constructed benchmarks using two sleep

This work was supported by grants from the National Heart, Lung, and Blood Institute (#R21HL159661), and the National Science Foundation (IUCRC #2052528 and CAREER #1943552).

H. Li, J. Roveda, and A. Li are with the Department of Electrical & Computer Engineering at the University of Arizona, Tucson, AZ 85719 USA. (E-mail: hl459@arizona.edu, meilingw@arizona.edu, aoli1@arizona.edu)

X. Chen is with the School of Computing at Clemson University, Clemson, SC 29634 USA. E-mail: xiwenc@g.clemson.edu

G. Ditzler is with the Department of Electrical & Computer Engineering at Rowan University, Glassboro, NJ 08028 USA. E-mail: ditzler@rowan.edu

W. Killgore is with the Department of Psychiatry at the University of Arizona, Tucson, AZ 85719 USA. E-mail: killgore@psychiatry.arizona.edu

S. Quan is with the Departments of Medicine and Neurology, Brigham and Women's Hospital, Harvard Medical School, Boston, MA, USA and Asthma and Airway Disease Research Center, College of Medicine, The University of Arizona, Tucson, AZ, USA. E-mail: stuart_quan@hms.harvard.edu

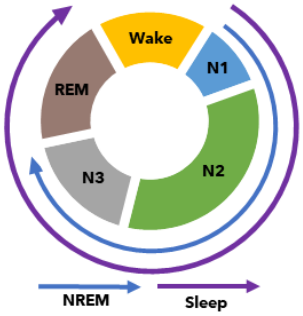

Fig. 1. Illustration of the hierarchy between the sleep stages, showcasing the relationships among different sleep stages. The hierarchy diagram visually represents the interconnectedness and hierarchical structure of the sleep stages.

TABLE I
DETAILS OF THE SLEEPEDF AND SHHS DATASET. EACH SAMPLE IS A 30 SECOND EEG SEGMENT.

Dataset		SleepEDF	SHHS
Subjects		153	329
EEG Channel		Fpz-Cz	C4-A1
Sampling Rate		100	125
Num of Each Stage	Wake	65,951	46,319
	Sleep	129,528	278,535
	NREM	103,693	212,582
	N1	21,522	10,304
	N2	69,132	142,125
	N3	13,039	60,153
	REM	25,835	65,953
Total 30s Epochs		195,479	324,854

study datasets to facilitate the examination of this application. Furthermore, we approach automatic sleep stage classification from two perspectives based on the accessibility of historical data, with a particular focus on the critical distribution shift between old and new datasets. To tackle this distribution shift challenge, we propose the use of unsupervised domain adaptation, which effectively aligns the distributions of the old and new datasets and addresses the issue of insufficient annotations in the old dataset. Additionally, we incorporate hierarchy-aware feature learning [5], which helps to extract informative features using old annotations. Another significant challenge is the unavailability of the old dataset. To address this issue, we propose using generative models [6], [7] to synthesize additional samples to interleave and leverage samples from the old data into the new data without actual access to the old dataset [8].

In summary, we present several significant contributions as follows: **(1)** Benchmarks are constructed based on the widely used sleep study datasets for facilitating further research. The benchmarks highlight the challenges faced by sleep stage classification with learning from evolving datasets. **(2)** Learning strategies with the combination of various techniques are explored to address the challenges associated with the evolving nature of automatic sleep stage classification. The experimental results show that we achieved remarkable results demonstrating the potential for efficient and accurate sleep stage classification under changing data distributions and evolving annotations. **(3)** We delve into unresolved questions and identify potential directions for future research in automatic sleep stage classification with learning from evolving datasets. By highlighting the limitations and gaps in current approaches, the study paves the way for further investigations and encourages researchers to explore novel methodologies and solutions.

II. MATERIALS AND BENCHMARKS

Understanding sleep and the different stages of sleep is crucial for diagnosing and treating sleep disorders. Therefore, accurate sleep stage classification is critical to provide personalized treatments for those suffering from sleep disorders. Sleep stages can be broadly classified into two main types: Rapid Eye Movement (REM) and Non-REM (NREM) sleep [9]. NREM sleep can be further segmented into three stages based on brain activity and muscle tone characteristics. The initial stage of NREM, namely N1, represents the lightest sleep stage and is characterized by the transition from wakefulness to sleep. The subsequent stage, N2, is slightly deeper and typically constitutes around half of the sleep time in healthy adults. The third stage, N3, corresponds to the deepest phase of NREM sleep, often called slow-wave sleep, which is crucial for physical restoration and memory consolidation [10]. Conversely, REM sleep is characterized by vivid dreaming, rapid eye movements, and temporary muscle paralysis. Transitions between sleep stages occur continuously and cyclically, with each cycle lasting approximately 90 to 110 minutes. Figure 1 shows the hierarchy between different sleep stages. For constructing the coarse annotation of the old dataset, we aggregate N1, N2, and N3 stages into the NREM stage, while combining the NREM and REM stages under the category of sleep.

The sleep community has collected and released two publicly available datasets that enables research in this domain. Namely, the Sleep Heart Health Study (SHHS) [11], [12] and Sleep-EDF Database Expanded (Sleep-EDF) [13], [14] are used in this work. The details of these datasets are provided in Table I. Sleep-EDF [14] comprises whole-night polysomnography (PSG) recordings, including EEG, EOG, chin EMG, and event markers. Each PSG file has two EEG channels, Fpz-Cz and Pz-Oz, sampled at 100 Hz. For our experiments, we used the Fpz-Cz channel as the input, resampling at a rate of 125 Hz to align with the sampling rate of the SHHS dataset. SHHS [11] is a multi-center cohort study conducted by the National Heart, Lung, and Blood Institute to investigate the cardiovascular and other impacts of sleep-disordered breathing. We chose the C4-A1 EEG channel from 329 PSG records out of 6,441 records as the input to our model, following the settings used in previous studies [4], [15]. The EEG signals in the SHHS dataset are sampled at 125 Hz.

In this work, we use the SHHS and Sleep-EDF datasets as benchmarks to evaluate the performance of the proposed strategies, as shown in Figure 2. Specifically, we designate the Sleep-EDF and SHHS as the old and new datasets, respectively. To gradually increase the difficulty level, we assign two label sets to the Sleep-EDF dataset: Wake/Sleep and Wake/NREM/REM. Meanwhile, for the SHHS dataset,

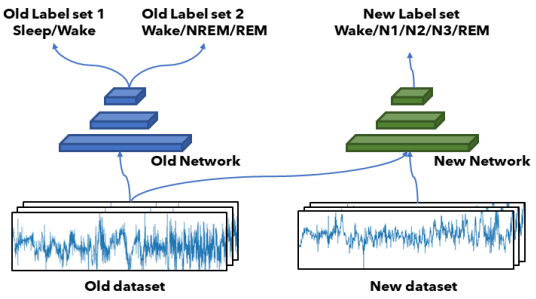

Fig. 2. Illustration of the benchmarks. Our objective is to evaluate the capability of a new network to classify the old dataset into five stages using only coarse annotation. Two types of coarse annotation are employed: Sleep/Wake and Wake/NREM/REM. The distinction between **B1** and **B2** lies in the accessibility of the old dataset.

we adopt a label set that encompasses all five stages, namely Wake/N1/N2/N3/REM. Further, we establish two benchmarks that account for the absence of the old dataset (i.e., Sleep-EDF). The first benchmark (**B1**) allows the new network access to the old dataset; however, the label sets remain unchanged and do not undergo relabeling to the five stages. The second benchmark (**B2**), which poses a greater challenge, does not allow the new network to access the old dataset. In this latter scenario, the new network is expected to train solely on the old network's knowledge (e.g., knowledge distillation), or the generative model trained using the old dataset. To evaluate the effectiveness of our proposed learning strategies, we use the new network trained on the SHHS dataset, which incorporates the more detailed five-stage label set, for performing five-stage classification on the Sleep-EDF dataset.

III. Related Works

A. Continual Learning and Deep Generative Replay

Continual learning [16]–[19]is a subfield of machine learning that focuses on learning over time while avoiding catastrophic forgetting [20]. Traditional machine learning approaches operate under the assumption that the data distribution remains fixed over time, enabling training on a static dataset. However, real-world environments often exhibit evolving or changing data distributions, which requires models to learn new concepts or adapt to nonstationary environments when facing new tasks [21]. Meanwhile, continual learning operates under the assumption that access to previous data is restricted solely to the current task.

Deep generative replay (DGR) [8] is a continual learning approach that uses generative models to produce synthetic data samples from previous tasks. DGR is in contrast to replay methods which retain a small sample data from prior tasks [22], [23]. The (synthetic) replay samples train the model when new task data arrive. The core idea is to use replay to prevent the network from forgetting data from the prior tasks without access to the old data. DGR has shown promise to mitigate catastrophic forgetting. This study examines the use of DGR to tackle the learning scenario within the context of sleep stage classification with learning from evolving datasets for several reasons. Firstly, DGR does not require access to the old dataset once the model is trained. Secondly, DGR is independent of the neural network architecture, making it compatible with new and old networks. Additionally, DGR can generate synthetic samples before training the new networks, reducing the training overhead.

B. Hierarchy Aware Feature Learning

Label hierarchies represent valuable resources that can be used across diverse domains, including biological taxonomy [24] and language datasets [25]. These hierarchies organize labels in a structured manner, effectively capturing the inherent relationships and semantic dependencies. In recent years, researchers increasingly recognize label hierarchy's potential to enhance classifier performance and substantially reduce the occurrence of severe errors [5], [26], [27]. One promising research direction in this context involves the exploration of hierarchy-aware feature learning [5]. This approach integrates hierarchical information encoded in label hierarchies into the learning process, enabling classifiers to make semantically meaningful mistakes while minimizing the overall error. By considering the relationships between labels at varying levels of granularity, hierarchy-aware feature learning provides a more nuanced comprehension of the underlying data structure and facilitates more informed decision-making.

Leveraging label hierarchies in learning tasks offers multiple advantages. Firstly, it facilitates the identification of shared characteristics and commonalities among related labels, enabling the classifier to generalize knowledge across similar categories. This is particularly beneficial in scenarios with limited or expensive labeled data, as the hierarchical relationships facilitate knowledge transfer from higher-level to lower-level labels. However, in this work, we use the concept of hierarchy-aware feature learning to leverage the annotations from the old datasets. It is important to note that our objective is not to reduce the severity of mistakes. Rather, we seek to achieve joint training of neural networks on the new and old datasets using coarse annotations. Our approach allows the networks to learn meaningful features from the old EEG signals.

C. Unsupervised Domain Adaptation

Unsupervised Domain Adaptation (UDA) [28]–[30] transfers knowledge from a source to a target domain without labeled information. These domain adaptation scenarios typically have limited labeled data in the target domain. The source and target domains frequently exhibit distinct data distributions in numerous real-world applications, including computer vision, natural language processing, and speech recognition. The primary objective of UDA is to mitigate the domain discrepancy and facilitate model generalization on the target domain by harnessing unlabeled data from the target domain with labeled data from the source domain. In contrast to supervised learning, which benefits from abundant labeled data, UDA leverages unsupervised learning to align the distributions of the source and target domains, thereby improving the model's performance on the target domain.

UDA makes an underlying assumption that despite the disparate distributions of the source and target domains, they possess similarities and structural patterns. UDA leverage shared

characteristics to learn domain-invariant representations that capture task-specific information while suppressing domain-specific variations. In this study, our specific focus lies in applying UDA to sleep stage classification with EEG data. We assume that the dissimilarities in data characteristics between the source and target datasets arise from distinct devices collecting the data and variations between patient groups. Additionally, we face the challenge of missing annotations in the old dataset for the five-stage classification, rendering supervised learning infeasible. Thus, UDA becomes a viable approach to adapt a model trained on the source domain to the target domain, circumventing the need for labeled data in the target domain.

IV. Methods

We address the problem of sleep stage classification with learning from evolving datasets from two perspectives by considering the access to historical data (**B1** and **B2**). Section IV-A presents a detailed problem formulation, elucidating our study's key challenges and goals. Subsequently, in Section IV-B, we introduce the concepts and techniques for unsupervised domain adaptation to align the old and new feature data distributions. Section IV-C covers the hierarchy aware features learning, which is used to exploit the coarse annotations of the old dataset. Finally, in Sections IV-D and IV-E, we present two generative models to synthesize data in scenarios where access to the old dataset is limited (**B2**).

A. Problem Formulation

As described in Section II, we are provided with the old dataset $\mathcal{D}^{old} := \{x_i^{old}, y_i^{old}\}_{i=1}^{N}$ comprising N input EEG signals x_i^{old}, and their corresponding sleep stages y_i^{old}. We use supervised learning to train the old neural network f^{old} on $\mathcal{D}^{old}$ by minimizing the cross-entropy loss, ℓ:

$$f^{old} = \arg\min_{f \in \mathcal{F}} \mathbb{E}_{\mathcal{D}^{old}} \ell(f(x_i^{old}), y_i^{old}), \tag{1}$$

At a future time, we receive a new dataset $\mathcal{D}^{new} = \{x_i^{new}, y_i^{new}\}_{i=1}^{M}$, where the input EEG signals x_i^{new} are sampled from a distribution different from that of x_i^{old}. This change can be due to changes in collection procedures, such as a new EEG collection device or patients from different sub-populations. Moreover, the new sleep stages y_i^{new} are annotated using different criteria than y_i^{old}. It is worth noting that some sleep stage annotations remain the same in both the old and new datasets (e.g., wake). Others are subcategories of the old annotations (i.e., N1/N2/N3 can be grouped as NREM, and N1/N2/N3/NREM can be grouped as sleep).

A new model f^{new} can easily be trained with the new dataset $\mathcal{D}^{new}$; however, such an approach fails to capture knowledge learned in f^{old}. Our learning setting requires using the new network to classify the old dataset based on the new labeling. The challenges associated with the new network with different labeling than the old dataset are twofold. First, the distribution shift between the old and new EEG signals will result in poor generalization performance of f^{new} evaluated on $\mathcal{D}^{old}$. Second, the absence of new annotations for $\mathcal{D}^{old}$ makes joint training infeasible even with access to the old dataset. Additionally, we assume that $\mathcal{D}^{old}$ will not be available once $\mathcal{D}^{new}$ is received, which makes training f^{new} even more challenging.

B. Aligning Feature Distributions via Unsupervised Domain Adaptation

UDA is a natural approach to address the distribution shift between datasets when labeled target domain data are scarcely available, which is the case for the old dataset. UDA methods adapt the model trained on the source domain (the new dataset) to the target domain (the old dataset) by aligning the feature distributions across the two domains. Here, we define the new and old EEG signal distributions as $\mathcal{X}^{new}$ and $\mathcal{X}^{old}$, respectively. Based on these definitions, Ben-David et al. [29] established an upper bound on the error (i.e., $\epsilon_{old}(f)$) of a classifier f on the target domain by f's error on the source domain (i.e., $\epsilon_{new}(f)$) in addition to several other terms. The bound is expressed as follows:

$$\epsilon_{old}(f) \leq \epsilon_{new}(f) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{X}^{new}, \mathcal{X}^{old}) + C \tag{2}$$

where C is a constant term independent of the specific classifier f. The $d_{\mathcal{H}\Delta\mathcal{H}}$ term represents the $\mathcal{H}\Delta\mathcal{H}$-distance, which characterizes the discrepancy between the two classifier's decisions over the two domains. Formally, the $\mathcal{H}\Delta\mathcal{H}$-distance is defined as:

$$\begin{aligned} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{X}^{new}, \mathcal{X}^{old}) = & 2 \sup_{f_1, f_2 \in \mathcal{F}} |P_{x \sim \mathcal{X}^{new}}[f_1(x) \neq f_2(x)] \\ & - P_{x \sim \mathcal{X}^{old}}[f_1(x) \neq f_2(x)]|. \end{aligned} \tag{3}$$

Minimizing the $\mathcal{H}\Delta\mathcal{H}$-distance in practice can be challenging and infeasible; however, recent work has developed methods to approximate this divergence. For example, Ganin and Lempitsky (2015) approximated this divergence with the Domain Adversarial Neural Network (DANN) framework [31] using JS-divergence, and it can be formulated as the following objective:

$$\begin{aligned} \min_{c,f} \max_{g} & \mathbb{E}_{\mathcal{D}^{new}} \ell(c \circ f^{'}(x_i^{new}), y_i^{new}) + \\ & \lambda \mathbb{E}_{(\mathcal{X}^{new}, \mathcal{X}^{old})} \text{JSD}(g \circ f^{'}(x_i^{new}), g \circ f^{'}(x_j^{old})), \end{aligned} \tag{4}$$

where we slightly change the notations by splitting the network f into the classifier c and feature extractor $f^{'}$. The term g is a domain classifier trained to maximize the domain classification error. Thus encouraging $f^{'}$ to learn domain-invariant features.

In [32], f-Domain-Adversarial Learning (namely, fDAL) enhances the training stability of UDA by minimizing an f-divergence. The f-divergence between two distribution functions P_s and P_t is defined as $D_\phi(P_s||P_t) = \int p_t(x) \phi(\frac{p_s(x)}{p_t(x)}) dx$, where p_s and p_t represent the densities of P_s and P_t, respectively. The f-divergence can also be reformulated using variational forms [33] as:

$$D_\phi(P_s||P_t) \geq \sup_{T \in \mathcal{T}} \mathbb{E}_{x \in P_s}[T(x)] - \mathbb{E}_{x \in P_t}[\phi^*(T(x))], \tag{5}$$

where $T : \mathcal{X} \to \text{dom}(\phi)$ is an arbitrary measurable function of the set $\mathcal{T}$, and ϕ^* is the conjugate function of ϕ. More

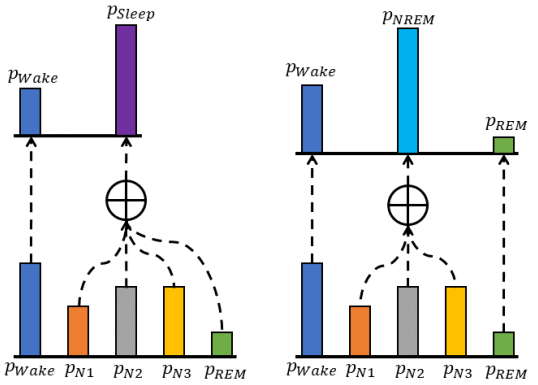

Fig. 3. Illustration of joint training with hierarchy learning. In this approach, we aggregate the output probabilities of the new sleep stages based on the hierarchy. This allows us to perform joint training on both the new and old datasets, leveraging the labels from the old dataset.

specifically, we can incorporate fDAL into our benchmark using the following min-max objective:

$$\min_{c,f'} \max_{g} \mathbb{E}_{\mathcal{D}^{new}} \ell(c \circ f'(x_i^{new}), y_i^{new}) + \mathbb{E}_{\mathcal{X}^{new}}[\hat{\ell}(g \circ f'(x_i^{new}), c \circ f'(x_i^{new}))] - \mathbb{E}_{\mathcal{X}^{old}}[(\phi^* \circ \hat{\ell})(g \circ f'(x_i^{old}), g \circ f'(x_i^{old}))]. \quad (6)$$

We let $\hat{\ell}(c, b) = a(b_{\arg\max_c})$, where $a(.)$ is a monotonically increasing function. By incorporating fDAL, the new network can generate domain-invariant features for both EEGs from the old and new datasets.

C. Exploit Old Labels with Hierarchy Aware Features Learning

In continual learning, joint training is commonly used to establish an upper bound for neural network performance. Joint training can be considered a near-optimal solution when the old dataset is fully annotated according to our desired specifications. Unfortunately, in our case, the old dataset's EEG signals have a coarser labeling level than the new dataset, making vanilla joint training impractical. We can still leverage the old dataset's annotations to enhance the new network's performance. Given that a significant challenge in our scenario is the domain shift between the two datasets, the network must acquire valuable features from the old dataset. Hence, we can address this requirement by applying Hierarchy Aware Feature Learning [5].Joint training in our setting can be achieved as follows. The new network provides output probabilities of the new annotations to the replay data with coarse sleep stage labels. This process is illustrated in Figure 3. Specifically, the cross-entropy loss computed on the old annotations can be defined as $l(Hp, y) = -\sum_{i=1}^{C}(Hp)_i \log(y_i)$, where p represents the softmax probability vectors, and H is a matrix utilized to calculate the summation of fine-to-coarse probabilities. The construction of matrix H involves setting $H_{i,j} = 1$ if the ith stage in the new annotations falls under the jth stage of the old annotations; otherwise, $H_{i,j} = 0$.

We use knowledge distillation loss to incorporate the old network's information using the old network's soft labels. Knowledge distillation facilitates knowledge transfer from the old (i.e., teacher) to the new (i.e., student) network. Specifically, we use the matrix H to aggregate the new network's output probabilities and then compute the Kullback-Leibler (KL) divergence $\mathrm{KL}^{\tau} = \sum_i q_i \log \frac{p_i^{\tau}}{q_i^{\tau}}$ using the old network's soft labels with a temperature factor, τ. Then we apply a softmax activation function to the logits z to get probabilities to obtain $p_i^{\tau} = \frac{\exp(z_i/\tau)}{\sum_j \exp(z_j/\tau)}$. The final objective for the joint training with hierarchy is as follows:

$$\min_{f} \mathbb{E}_{(\mathcal{D}^{new}, \mathcal{D}^{old})}[l(Hf(x_i), y_i^{old})] + \mathbb{E}_{\mathcal{X}^{old}}[\mathrm{KL}^{\tau}(Hf(x_i)), f_{old}(x_i)]. \quad (7)$$

In this objective, we optimize the expectation over the new and old datasets by minimizing the cross entropy loss $l(Hf(x_i), y_i^{old})$ for the old annotations, and the KL divergence $D_{KL}^{\tau}(Hf(x_i)), f_{old}(x_i)$ for the old network's soft labels. By jointly considering the new and old datasets, we aim to improve the new network's performance by leveraging the knowledge acquired by the old network, and enhancing the transfer of information across different sleep stage annotations.

By incorporating UDA and joint training with label hierarchy, our approach seeks to leverage the benefits of both techniques. Through UDA, we address the domain shift between the old and new datasets, allowing the new network to learn domain-invariant features and adapt to the characteristics of the target domain. Simultaneously, the joint training with hierarchy enables the new network to use the information from the old network, leveraging the fine-to-coarse annotations to enhance the performance on the coarse sleep stages. This combination of UDA and joint training with hierarchy provides a comprehensive framework that addresses the challenges posed by our sleep-stage classification scenarios. Further, our approach enables the new network to learn discriminative features from the new dataset, while benefiting from the knowledge distilled from the old network. Our approach enhances the generalization capability of the new network by jointly optimizing the UDA and hierarchy, thus, facilitating improved classification accuracy for both the old and new sleep stages.

D. Generative Samples with Wasserstein GAN

We have discussed the strategies used to address the distribution shift between datasets and the absence of annotations. However, we assume the old dataset is unavailable in scenario **B2**. One potential solution is to use Generative Adversarial Networks (GANs) to generate realistic synthetic ECG signals [34]. GANs have demonstrated effectiveness in various biomarker classification tasks, such as augmenting imbalanced datasets for electrocardiogram (ECG) classification [35]. A GAN consists of two neural networks: a generator G and a discriminator C. These networks collaborate within a game-theoretic framework to learn the underlying distribution of the training data and generate new samples that closely resemble the data distribution. Generally, GANs are trained using the

following objective:

$$\min_G \max_C \mathbb{E}_{x\sim P_{\text{data}}}[\log C(x)] + \mathbb{E}_{z\sim P_{\text{noise}}}[\log(1 - C(G(z))]. \tag{8}$$

In this formulation, x (z) represents real (noise) data samples, P_{data} denotes the distribution of the real data, and P_{noise} denotes the distribution of the noise.

Wasserstein GANs (WGAN) [36] are a variant of the original GAN that addresses challenges associated with training. WGAN introduces a different loss function from (8). Specifically, the Wasserstein distance, also known as the Earth Mover's Distance, to measure the dissimilarity between the generated and real distributions. This modification improves the stability and reliability of GAN training (e.g., reduction of mode collapse). The objective of WGAN can be expressed as follows:

$$\min_G \max_C \mathbb{E}_{x\sim P_{\text{data}}}[C(x)] - \mathbb{E}_{z\sim P_{\text{noise}}}[C(G(z)]. \tag{9}$$

Wasserstein GAN with Gradient Penalty (WGAN-GP) [6] is an enhanced version of WGAN that incorporates gradient penalty regularization (GP) to enforce the Lipschitz continuity constraint on the discriminator. This modification improves training stability and prevents mode collapse by controlling the discriminator's power. The following equation represents the objective function of WGAN-GP:

$$\begin{aligned}\min_G \max_C \mathbb{E}_{x\sim P_{\text{data}}}[C(x)] - \mathbb{E}_{z\sim P_{\text{noise}}}[C(G(z)]+ \\ \lambda \mathbb{E}_{\hat{x}\sim P\text{interp}}[(||\nabla_{\hat{x}} C(\hat{x})||_2 - 1)^2]\end{aligned} \tag{10}$$

where λ is a hyperparameter to control the strength of the gradient penalty. Here, $\hat{x}$ represents a point along the straight line connecting a real sample and a sample generated from P_{interp}. While the original WGAN-GP is an unconditional model with non-conditional probability distributions in the loss function, we aim to generate synthetic samples conditioned on labels. To achieve this, we introduce random synthetic labels y' and denote the true labels of real samples as y. Accordingly, $\hat{y}$ represents a point along the straight line connecting the real and synthetic labels, sampled from P_{interp}. The conditional version of WGAN-GP is trained using the following objective:

$$\begin{aligned}\min_G \max_C \mathbb{E}_{x\sim P_{\text{data}}}[C(x|y)] - \mathbb{E}_{z\sim P_{\text{noise}}}[C(G(z)|y^{'})]+ \\ \lambda \mathbb{E}_{\hat{x}\sim P\text{interp}}[(||\nabla_{\hat{x}} C(\hat{x}|\hat{y})||_2 - 1)^2],\end{aligned} \tag{11}$$

We can generate synthetic samples conditioned on specific classes by introducing conditional labels y and y'. Thus, improving the applicability of the WGAN-GP framework to our task. This modification allows us to generate ECG signals with label-specific characteristics, facilitating more targeted analysis and classification tasks.

E. Generative Samples with Denoising Diffusion Probabilistic Models

In the field of deep generative models, the Denoising Diffusion Probabilistic Model (DDPM) [7] belongs to a category of models that focus on converting noise into realistic data samples by progressively eliminating noise through a denoising procedure. In this approach, the training data are iteratively corrupted by introducing Gaussian noise, and the model is trained to reverse this process and restore the original data. As a result, a well-trained DDPM can generate novel data by applying a denoising process to randomly generated noise.

Specifically, DDPM encompasses two main processes: the forward process, also known as the diffusion process, where data is progressively diffused to a well-behaved distribution by adding noise, and the reverse process, which transforms noise back into a sample from the target distribution. In the forward process, a distribution denoted as q gradually introduces noise to a given data point $x_0 \sim q(x_0)$. DDPM implements the diffusion process using a fixed Markov Chain with conditional Gaussian translation at each step, defined as follows:

$$q\left(x_{1:T}|x_0\right) := \prod_{t=1}^{T} q\left(x_t|x_{t-1}\right), \tag{12}$$

$$q\left(x_t|x_{t-1}\right) := \mathcal{N}\left(x_t; \sqrt{1-\beta_t}x_{t-1},\ \beta_t\mathbf{I}\right), \tag{13}$$

where $\beta_1, \ldots, \beta_T$ represents a variance schedule, and $\mathcal{N}$ denotes the Gaussian distribution with parameters μ and Σ. In contrast, the reverse process aims to recover the initial data point x_0 from a given state x_t by reversing the diffusion process. Starting with pure Gaussian noise sampled from $p(x_T) := \mathcal{N}(x_T, \mathbf{0}, \mathbf{I})$, the reverse process is defined by the following Markov chain:

$$p_\theta\left(x_{0:T}\right) := p\left(x_T\right) \prod_{t=1}^{T} p_\theta\left(x_{t-1}|x_t\right), x_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{14}$$

$$p_\theta\left(x_{t-1}|x_t\right) := \mathcal{N}\left(x_{t-1}; \boldsymbol{\mu}_\theta\left(x_t, t\right), \sigma_\theta\left(x_t, t\right)\mathbf{I}\right). \tag{15}$$

In this process, the time-dependent parameters of the Gaussian transitions are learned. In the context of DDPM, a specific parameterization for $p_\theta\left(x_{t-1}|x_t\right)$ is proposed:

$$\boldsymbol{\mu}_\theta\left(x_t, t\right) = \frac{1}{\alpha_t}\left(x_t - \frac{\beta_t}{\sqrt{1-\alpha_t}}\boldsymbol{\epsilon}_\theta\left(x_t, t\right)\right), \tag{16}$$

$$\sigma_\theta\left(x_t, t\right) = \sqrt{\tilde{\beta}_t}, \text{ where } \tilde{\beta}_t = \begin{cases} \frac{1-\alpha_{t-1}}{1-\alpha_t}\beta_t & t > 1 \\ \beta_1 & t = 1 \end{cases} \tag{17}$$

where $\boldsymbol{\epsilon}_\theta(\cdot, \cdot)$ is a learnable denoising function that estimates the noise vector $\boldsymbol{\epsilon}$ added to a noisy input x_t. The parameterization leads to an alternative loss function:

$$L(\theta) := \mathbb{E}_{t, x_0, \epsilon}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\left(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t\right)\right\|^2\right], \tag{18}$$

where $\bar{\alpha}t$ represents the schedule of values for αt.

Given the synthetic labels y', the conditional version of DDPM can be obtained by estimating the true conditional data distribution $q(x_0|y')$ through modeling the conditional distribution $p_\theta(x_0|y')$. Consequently, the *reverse process* is extended as follows:

$$p_\theta\left(x_{0:T}|y^{'}\right) := p\left(x_T\right) \prod_{t=1}^{T} p_\theta\left(x_{t-1}|x_t, y^{'}\right), x_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I}),$$

$$p_\theta\left(x_{t-1}|x_t, y^{'}\right) := \mathcal{N}\left(x_{t-1}; \boldsymbol{\mu}_\theta\left(x_t, t|y^{'}\right), \sigma_\theta\left(x_t, t|y^{'}\right)\mathbf{I}\right).$$

To accommodate the conditional aspect, a conditional denoising function $\boldsymbol{\epsilon}_\theta$ is introduced, which is conditioned on y'. This

TABLE II
ARCHITECTURE OF WGAN-GP. THE CONFIGURATION OF CONV1D IS SPECIFIED BY THE FOLLOWING PARAMETERS: OUTPUT CHANNEL, KERNEL SIZE, STRIDE, AND PADDING.

Generator		Discriminator	
Embedding 5→28	Randn 1×100	Embedding 5→100 Linear 3750	EEG signal 1×3750
Input size: 1×128		Input size: 2×3750	
Linear 128 BatchNorm,ReLU		Conv1D 32,4,2,0 InstanceNorm,ReLU	
Linear 256 BatchNorm,ReLU		Conv1D 64,4,2,0 InstanceNorm,ReLU	
Linear 512 BatchNorm,ReLU		Conv1D 128,4,2,0 InstanceNorm,ReLU	
Linear 3750 BatchNorm,ReLU		Conv1D 256,4,2,0 InstanceNorm,ReLU	
Conv1D 32,4,1,2 BatchNorm,ReLU		Conv1D 512,4,2,0 InstanceNorm,ReLU	
Conv1D 64,4,1,2 BatchNorm,ReLU		Conv1D 1,1,1,0 InstanceNorm,ReLU	
Conv1D 128,4,1,2 BatchNorm,ReLU			
Conv1D 1,4,1,0 BatchNorm,ReLU			

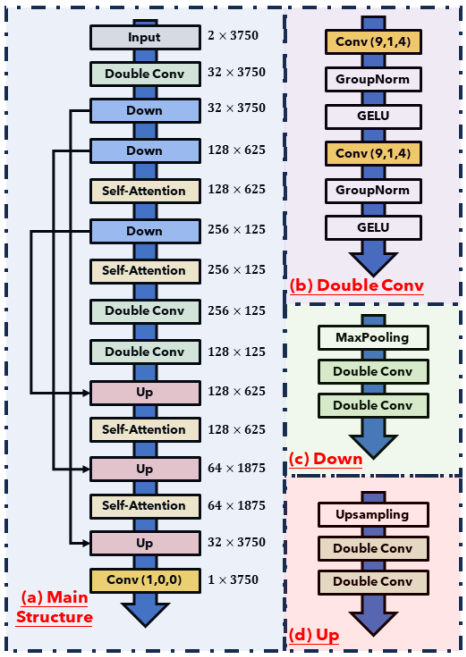

Fig. 4. UNet backbone architecture for DDPM. The main structure (a) is composed of downsampling modules (c) and upsampling modules (d), which include the double convolution module (b).

allows for the definition of the conditional loss function as follows:

$$\mathbb{E}_{x_0,\bar{\alpha},\epsilon}\left[\left\|\boldsymbol{\epsilon}-\boldsymbol{\epsilon}_\theta\left(\sqrt{\bar{\alpha}}x_0+\sqrt{1-\bar{\alpha}}\boldsymbol{\epsilon},y^{'},\bar{\alpha}\right)\right\|^2\right]. \quad (19)$$

In our case, DDPM offers several advantages. DDPM can generate new conditional data samples by leveraging the learned denoising process and the synthetic labels y'. DDPM effectively captures the conditional dependencies between the generated data and the corresponding labels by modeling the conditional distribution $p_\theta(x_0|y')$. Consequently, it generates realistic and diverse samples that align with specific label conditions. Further, the conditional loss function ensures that the generated samples exhibit similarity to the noise vector while conforming to the conditioning information. Thus, DDPM provides a powerful framework for conditional data generation, which makes it highly suitable for our particular application.

V. EXPERIMENTS

A. Experimental Configurations

This study used two neural networks: ResNet [37] and AttnSleep [4] as the old and new networks, respectively. AttnSleep was chosen for its architecture, which incorporates transformer layers [38] to effectively capture long-range dependencies. Both neural networks were trained with a batch size of 128 using the Adam optimizer [39]. The initial learning rate was set to 1×10^{-3} and then reduced to 1×10^{-4} after ten epochs. To mitigate overfitting, a weight decay of 1×10^{-3} was applied within Adam. The class-aware cross-entropy loss was used to train both networks on their respective datasets [4]. This loss function, denoted as $\ell(p,q) = \sum_i w_k q_i \log p_i$, includes precalculated weights w_k for each class k. These weights are crucial to address class imbalance and ensure balanced learning during training. We introduced an additional batch of size 128, sampled from either the old or synthetic datasets, to train the new network. This approach allowed us to evaluate the new network's performance under different conditions. The central learning objective is a linear combination of Equations 6 and 7. We adopted the same configuration of fDAL as described in the original paper [32], by using the Pearson χ^2 function and its conjugate. The generator and discriminator architectures in the WGAN are reported in Table II. The training procedure proposed by [40] is used to train the WGAN. Additionally, the architecture of the UNet Backbone [41] used in the DDPM (see Figure 4). The training procedures used in our previous work [42] for ECG reconstruction were replicated. All experiments were run on an NVIDIA RTX 3090 GPU. To ensure fair comparisons, fixed random seeds were used throughout the experiments. Further, we performed five-fold cross-validation and report the average performance metric.

B. Evaluation Metrics

We use multiple figures of merit to assess performance. These metrics include per-class F1-score (F1), accuracy, the area under the Receiver Operating Characteristic curve (AUROC), and the area under the Precision-Recall curve (AUPRC). Each metric provides valuable insights into the strengths and limitations of the model [43]. We denote true positive predictions as TP, true negative predictions as TN, false positive predictions as FP, and false negative predictions as FN. Precision (P) is calculated as $\frac{\text{TP}}{\text{TP+FP}}$, while Recall (R) is calculated as $\frac{\text{TP}}{\text{TP+FN}}$.

The *F1 Score*, a widely recognized metric for binary classification tasks, balances precision and recall, and is calculated as $\text{F1} = 2 \times \frac{\text{P} \times \text{R}}{\text{P+R}}$. This score is the harmonic mean of precision and recall. We compute the F1 score for each sleep stage and then average F1 score across all stages (MF1). The *Accuracy*

is another commonly used metric, representing the proportion of correct predictions out of the total number of predictions. The accuracy is calculated as $\text{Acc} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}}$, providing a high-level assessment of the model's correctness. Further, we also use *AUROC*, which assesses performance at distinguishing between positive and negative instances across various classification thresholds. AUROC is computed by integrating the Receiver Operating Characteristic (ROC) curve. In scenarios with class imbalance, such as sleep stage classification, *AUPRC* holds significance. It considers the precision and recall at different thresholds, comprehensively evaluating the classifier's performance. AUPRC is calculated by integrating the precision-recall curve. By using this diverse set of evaluation metrics, we gain an understanding of our model's performance and suitability for automatic sleep stage classification.

C. Main results

We evaluate the five-stage classification performance of AttnSleep on Sleep-EDF following the **B1** and **B2** settings defined in Section II. Table III shows the main results on the two benchmarks and two baselines. We first compare the performance of joint training and training on only SHHS. Joint training can be considered a loose upper bound and training on SHHS only can be considered a lower bound. The lower bound baseline is designed to establish a performance threshold that represents the minimal achievable results. This baseline typically involves naïve approaches that do not incorporate advanced techniques or use all available resources. When comparing our method against the lower bound, we can assess how much our approach surpasses or outperforms the minimal expectations. In contrast, the upper bound represents the ideal or optimal performance. By comparing our method against the upper bound, we can identify the gaps or limitations of our approach and determine areas for further improvement.

Our proposed strategy in **B1** surpasses the lower bound in per-class F1 without the need to manually relabel the old data set, achieving gains ranging from 0.02 to 0.1 when the old dataset has a sleep/wake annotation and gains of 0.05 to 0.2 when it has wake/REM/NREM annotation (see Table III). This result demonstrates the advantages of our approach. Unsurprisingly, the results on **B1** yield better performance than those obtained on **B2**. For example, if the old dataset is labeled with sleep/wake, using **B1** can obtain a gain of 0.12 and 0.04 MF1 over WGAN-GP and DDPM using **B2**, respectively. Further, if the old dataset is labeled with wake/REM/NREM then using **B1** can achieve 0.05 and 0.03 AUPRC gains over WGAN-GP and DDPM on **B2**, respectively. These results are expected since **B1** allows the new network to be trained on the old dataset.

However, accessing the old data is often not feasible or costly due to several concerns; therefore, **B2** is more challenging but valuable for such applications. Fortunately, our proposed strategy can still achieve reasonable gains over SHHS by using the generative model to synthesize the old data. We observe in this benchmark that DDPM is a more accurate network than WGAN-GP (see Table III), and DDPM achieves a 0.02 and 0.08 MF1 gain over SHHS for the old dataset labeled with sleep/wake and wake/REM/NREM, respectively. Moreover, if we look closer into the class-wise evaluation, DDPM obtains similar performance and has 0.01 to 0.02 per-class F1 drops over **B1** on class Wake and REM if the old dataset has wake/REM/NREM annotation. These results demonstrate that even with no or limited access to old data, our approach still provides benefits for hierarchy-aware feature learning on the new task while preserving information from the old dataset.

D. Ablation study

An ablation study was conducted on **B1** using the wake/REM/NREM old label set to investigate the learning strategies' impact. We select five strategies for comparison: **S1** represents the proposed integrated strategies of UDA (Section IV-B) and hierarchy joint training (Section IV-C), **S2** corresponds to hierarchy joint training, **S3** refers to only using UDA, **S4** represents hierarchy joint training without a KD loss, and **S5** represents the proposed integrated strategies (**S1**) without KD loss. The results for each strategy are presented in Table IV. These results show that replacing or omitting different aspects of our proposed strategy in **S2**-**S5** leads to a drop in class-wise and overall performance. For example, the experiments with **S2**-**S5** result in an MF1 degradation ranging from 0.02 to 0.2. The impact becomes more pronounced in the class-wise evaluation, especially for class N1. In the N1 class, S1 achieves per-class F1 scores that are multiple factors larger than those obtained using S2, S5, and S4 (i.e., 2x-7x improvement). These results reaffirm the need for our design to address the classification challenges presented by changing data distributions.

VI. Discussion and Limitations

This paper proposes an integrated learning strategy to address the challenge of sleep stage classification with learning from evolving datasets. This strategy offers a preliminary solution to enable machine learning models to adapt and learn effectively from datasets with evolving label sets. Experimental results demonstrate that integrating these learning strategies achieves classification performance comparable to joint training with relabeled old datasets. Additionally, the ablation study offers insights into the individual contributions of different learning strategies. By comparing the performance of various strategies, the study underscores that integrating these strategies, as proposed in this work, yields superior performance compared to individual or alternative approaches. Further, incorporating deep generative models such as WGAN-GP and DDPM addresses the practical constraints related to access to old datasets. Leveraging these generative models, the classifier can learn from historical data while adapting to current circumstances, providing a valuable solution for scenarios where direct access to the old data is unfeasible.

This study highlights the need for sleep stage classification algorithms with evolving label sets, changing data distributions, and inaccessible historical data. For example, in sleep stage scoring, the criteria for identifying each sleep stage evolve over time [44]. Moreover, classification tasks undergo

TABLE III
MAIN RESULTS ON THE TWO BENCHMARKS.

			Per-class F1					Overall Metrics			
			Wake	N1	N2	N3	REM	Accuracy	AUROC	AUPRC	MF1
SHHS (Baseline)			0.8473	0.1738	0.6468	0.5716	0.4923	66.20	0.7727	0.6371	0.6386
Joint training (Upper Bound)			0.8971	0.4282	0.8100	0.7672	0.7060	76.92	0.8221	0.7577	0.7610
Benchmarks	Generator	Old Labels									
B1	-	sleep/wake	0.8948	0.1980	0.7215	0.6781	0.5804	71.28	0.7937	0.6944	0.7060
		wake/REM/NREM	0.9070	0.2381	0.7686	0.7119	0.6977	75.88	0.8069	0.7315	0.7580
B2	WGAN-GP	sleep/wake	0.8273	0.1588	0.5768	0.4778	0.4194	58.83	0.7399	0.6148	0.5855
		wake/REM/NREM	0.8729	0.1648	0.6901	0.5929	0.6223	69.03	0.7812	0.6800	0.6916
	DDPM	sleep/wake	0.8617	0.2141	0.6921	0.6582	0.4659	67.12	0.7719	0.6568	0.6661
		wake/REM/NREM	0.8990	0.2158	0.7231	0.6142	0.6772	72.25	0.7957	0.7033	0.7227

TABLE IV
ABLATION STUDY ON THE INTEGRATION OF PROPOSED LEARNING STRATEGIES.

		S1	S2	S3	S4	S5
Per-class F1	Wake	0.9070	0.9042	0.5777	0.8994	0.8936
	N1	0.2381	0.0793	0.1781	0.0299	0.1075
	N2	0.7686	0.5995	0.6328	0.6152	0.7579
	N3	0.7119	0.5133	0.6713	0.5166	0.6955
	REM	0.6977	0.6916	0.4231	0.6774	0.6668
Overall Metrics	Accuracy	75.88	66.73	55.47	66.92	73.89
	AUROC	0.8069	0.8029	0.7270	0.7962	0.8019
	AUPRC	0.7315	0.7159	0.5852	0.6982	0.7123
	MF1	0.7580	0.6663	0.5481	0.6690	0.7352

adaptations to meet the specific demands of diverse applications [45]. Consequently, there is an urgent need to develop algorithms capable of effectively handling cross-datasets with evolving label sets. Therefore, accommodating the dynamics of real-world scenarios. In addition, domain changes between old and new datasets frequently occur due to advancements in ambulatory diagnosis equipment and variations in patient demographics. In particular, the inaccessibility of old datasets is frequently due to privacy concerns, policy restrictions, and the discontinuation of collaborations [46]. Consequently, we introduce an integrated strategy to tackle the challenge of sleep stage classification under changing dataset shifts, and offer a solution to address this issue.

Further, the significance of investigating learning from evolving datasets goes beyond the specific application domain. It serves as a representative example of the broader challenges faced in various fields where data distributions and classification requirements undergo temporal changes. In the rapidly evolving landscape of today's world, datasets collected from various sources exhibit inherent dynamics due to technological advancements, evolving user behaviors, and shifting environmental conditions. These dynamics introduce significant complexities that traditional static classification approaches are ill-equipped to address. Consequently, there is an increasing demand for methodologies and algorithms that can adapt to the changing nature of data and classification contexts. Hence, a reliable system can effectively handle nonstationary data distributions, adapt to evolving classification tasks, and provide accurate and up-to-date insights in dynamic real-world scenarios. The implications of such advancements are far-reaching, impacting fields such as healthcare, finance, social sciences, and many others, where the ability to classify and interpret dynamic data is critical for informed decision-making and understanding complex phenomena.

However, several open questions remain that need to be addressed in future research. Firstly, the experimental findings show low performance for the N1 stage. This outcome begs the need to investigate methods for improving the accuracy on specific indistinguishable categories. Further exploration is needed to understand the factors contributing to lower performances (especially the performances on the N1 stage). Additionally, the results indicate that generative methods perform inferiorly compared to using the "real" old dataset. This result raises the question of improving the synthesis of data from generative models to better reflect the distributions of real old data better. Exploring advanced techniques, such as adversarial training or incorporating additional constraints, may improve the performance of generative methods in the context of sleep stage classification. Moreover, this study focuses on the challenges of changing data distributions in sleep stage classification. Other related classification tasks in the healthcare domain face similar challenges. Investigating the proposed learning strategies for different healthcare domains, such as disease diagnosis or patient monitoring, presents an interesting future direction. These studies would involve adapting the strategies to the specific characteristics of different datasets and classification tasks, and evaluating their effectiveness in improving classification performance in those domains.

VII. CONCLUSION

This paper investigates the challenges of sleep stage classification with learning from evolving datasets caused by the dynamic nature of the datasets. We construct benchmarks using two widely employed sleep datasets to simulate learning scenarios and assess the performance of the algorithms. We tackle these challenges by integrating multiple techniques. By leveraging a combination of UDA, Hierarchy-Aware Feature Learning, and deep generative models, we achieve exceptional results, highlighting the potential for efficient and precise sleep stage classification amidst shifting data distributions, varying label granularity, and missing historical datasets. In summary, this study significantly advances the field of sleep stage classification by creating benchmarks and proposing the learning strategies. Furthermore, the concept of learning from evolving datasets, akin to that employed in sleep stage classification, has the potential for broader application across various healthcare domains. This research paves the way for future explorations in sleep stage classification and other classification tasks confronted with comparable challenges.

REFERENCES

[1] F. S. Luyster, P. J. Strollo Jr, P. C. Zee, and J. K. Walsh, "Sleep: a health imperative," *Sleep*, vol. 35, no. 6, pp. 727–734, 2012.

[2] A. Supratak, H. Dong, C. Wu, and Y. Guo, "Deepsleepnet: A model for automatic sleep stage scoring based on raw single-channel eeg," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 11, pp. 1998–2008, 2017.

[3] S. Mousavi, F. Afghah, and U. R. Acharya, "Sleepeegnet: Automated sleep stage scoring with sequence to sequence deep learning approach," *PloS one*, vol. 14, no. 5, p. e0216456, 2019.

[4] E. Eldele, Z. Chen, C. Liu, M. Wu, C.-K. Kwoh, X. Li, and C. Guan, "An attention-based deep learning approach for sleep stage classification with single-channel eeg," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 29, pp. 809–818, 2021.

[5] A. Garg, D. Sani, and S. Anand, "Learning hierarchy aware features for reducing mistake severity," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIV*. Springer, 2022, pp. 252–267.

[6] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," *Advances in neural information processing systems*, vol. 30, 2017.

[7] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.

[8] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," *Advances in neural information processing systems*, vol. 30, 2017.

[9] A. K. Patel, V. Reddy, and J. F. Araujo, "Physiology, sleep stages," in *StatPearls [Internet]*. StatPearls Publishing, 2022.

[10] U. Wagner and J. Born, "Memory consolidation during sleep: interactive effects of sleep stages and hpa regulation," *Stress*, vol. 11, no. 1, pp. 28–41, 2008.

[11] S. F. Quan, B. V. Howard, C. Iber, J. P. Kiley, F. J. Nieto, G. T. O'Connor, D. M. Rapoport, S. Redline, J. Robbins, J. M. Samet *et al.*, "The sleep heart health study: design, rationale, and methods," *Sleep*, vol. 20, no. 12, pp. 1077–1085, 1997.

[12] G.-Q. Zhang, L. Cui, R. Mueller, S. Tao, M. Kim, M. Rueschman, S. Mariani, D. Mobley, and S. Redline, "The national sleep research resource: towards a sleep data commons," *Journal of the American Medical Informatics Association*, vol. 25, no. 10, pp. 1351–1358, 2018.

[13] B. Kemp, A. H. Zwinderman, B. Tuk, H. A. Kamphuisen, and J. J. Oberye, "Analysis of a sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the eeg," *IEEE Transactions on Biomedical Engineering*, vol. 47, no. 9, pp. 1185–1194, 2000.

[14] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals," *circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

[15] P. Fonseca, N. Den Teuling, X. Long, and R. M. Aarts, "Cardiorespiratory sleep stage detection using conditional random fields," *IEEE journal of biomedical and health informatics*, vol. 21, no. 4, pp. 956–966, 2016.

[16] I. J. Goodfellow, M. Mirza, D. Xiao, A. Courville, and Y. Bengio, "An empirical investigation of catastrophic forgetting in gradient-based neural networks," *arXiv preprint arXiv:1312.6211*, 2013.

[17] 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," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[18] S.-W. Lee, J.-H. Kim, J. Jun, J.-W. Ha, and B.-T. Zhang, "Overcoming catastrophic forgetting by incremental moment matching," *Advances in neural information processing systems*, vol. 30, 2017.

[19] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.

[20] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychology of learning and motivation*. Elsevier, 1989, vol. 24, pp. 109–165.

[21] G. Ditzler, M. Roveri, C. Alippi, and R. Polikar, "Adaptive strategies for learning in nonstationary environments: a survey," *Computational Intelligence Magazine*, vol. 10, no. 4, pp. 12–25, 2015.

[22] S.-A. Rebuffi, A. Kolesnikov, and C. H. Lampert, "iCaRL: Incremental classifier and representation learning," in *International Conference on Computer Vision*, 2017.

[23] A. Antoniou, M. Patacchiola, M. Ochal, and A. Storkey, "Defining Benchmarks for Continual Few-Shot Learning," 2020. [Online]. Available: http://arxiv.org/abs/2004.11967

[24] G. Van Horn, O. Mac Aodha, Y. Song, Y. Cui, C. Sun, A. Shepard, H. Adam, P. Perona, and S. Belongie, "The inaturalist species classification and detection dataset," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8769–8778.

[25] C. Fellbaum, *WordNet: An electronic lexical database*. MIT press, 1998.

[26] H. Wu, M. Merler, R. Uceda-Sosa, and J. R. Smith, "Learning to make better mistakes: Semantics-aware visual food recognition," in *Proceedings of the 24th ACM international conference on Multimedia*, 2016, pp. 172–176.

[27] S. Karthik, A. Prabhu, P. K. Dokania, and V. Gandhi, "No cost likelihood manipulation at test time for making better mistakes in deep networks," *arXiv preprint arXiv:2104.00795*, 2021.

[28] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," *Advances in neural information processing systems*, vol. 19, 2006.

[29] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Machine learning*, vol. 79, pp. 151–175, 2010.

[30] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The journal of machine learning research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[31] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *International conference on machine learning*. PMLR, 2015, pp. 1180–1189.

[32] D. Acuna, G. Zhang, M. T. Law, and S. Fidler, "f-domain adversarial learning: Theory and algorithms," in *International Conference on Machine Learning*. PMLR, 2021, pp. 66–75.

[33] X. Nguyen, M. J. Wainwright, and M. I. Jordan, "Estimating divergence functionals and the likelihood ratio by convex risk minimization," *IEEE Transactions on Information Theory*, vol. 56, no. 11, pp. 5847–5861, 2010.

[34] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[35] E. Adib, F. Afghah, and J. J. Prevost, "Arrhythmia classification using cgan-augmented ecg signals," in *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2022, pp. 1865–1872.

[36] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International conference on machine learning*. PMLR, 2017, pp. 214–223.

[37] A. I. Humayun, A. S. Sushmit, T. Hasan, and M. I. H. Bhuiyan, "End-to-end sleep staging with raw single channel eeg using deep residual convnets," in *2019 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*. IEEE, 2019, pp. 1–5.

[38] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[40] E. Adib, A. Fernandez, F. Afghah, and J. J. Prevost, "Synthetic ecg signal generation using probabilistic diffusion models," *arXiv preprint arXiv:2303.02475*, 2023.

[41] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 2015, pp. 234–241.

[42] H. Li, G. Ditzler, J. Roveda, and A. Li, "Descod-ecg: Deep score-based diffusion model for ecg baseline wander and noise removal," *IEEE Journal of Biomedical and Health Informatics*, 2023.

[43] T. Fawcett, "An introduction to {ROC} analysis," *Pattern Recognition Letters*, vol. 27, pp. 861–874, 2006.

[44] R. B. Berry, R. Brooks, C. Gamaldo, S. M. Harding, R. M. Lloyd, S. F. Quan, M. T. Troester, and B. V. Vaughn, "Aasm scoring manual updates for 2017 (version 2.4)," pp. 665–666, 2017.

[45] G. Cay, V. Ravichandran, S. Sadhu, A. H. Zisk, A. Salisbury, D. Solanki, and K. Mankodiya, "Recent advancement in sleep technologies: A literature review on clinical standards, sensors, apps, and ai methods," *IEEE Access*, 2022.

[46] E. Vayena, A. Blasimme, and I. G. Cohen, "Machine learning in medicine: addressing ethical challenges," *PLoS medicine*, vol. 15, no. 11, p. e1002689, 2018.