CLOPS: Continual Learning of Physiological Signals

Dani Kiyasseh
Department of Engineering Science
University of Oxford
Oxford, UK
dani.kiyasseh@eng.ox.ac.uk

Tingting Zhu
Department of Engineering Science
University of Oxford
Oxford, UK
tingting.zhu@eng.ox.ac.uk

David A. Clifton
Department of Engineering Science
University of Oxford
Oxford, UK
david.clifton@eng.ox.ac.uk

Abstract

Deep learning algorithms are known to experience destructive interference when instances violate the assumption of being independent and identically distributed (i.i.d). This violation, however, is ubiquitous in clinical settings where data are streamed temporally and from a multitude of physiological sensors. To overcome this obstacle, we propose CLOPS, a healthcare-specific replay-based continual learning strategy. In three continual learning scenarios based on three publically-available datasets, we show that CLOPS can outperform its multi-task learning counterpart. Moreover, we propose end-to-end trainable parameters, which we term task-instance parameters, that can be used to quantify task difficulty and similarity. This quantification yields insights into both network interpretability and clinical applications, where task difficulty is poorly quantified.

1 Introduction

Deep learning algorithms typically expect instances from a dataset to be independent and identically distributed (i.i.d.). Therefore, violating these assumptions can be detrimental to the training behaviour and performance of an algorithm. For instance, the independence assumption can be violated when data are streamed temporally from a particular sensor. On the other hand, the introduction of multiple sensors in a changing environmental context can introduce covariate shift, arguably the 'Achilles heel' of machine learning model deployment [Quionero-Candela et al., 2009].

A plethora of realistic scenarios violate the i.i.d. assumption. This is particularly true in healthcare where the multitude of physiological sensors generate time-series recordings that may vary temporally (due to seasonal diseases; e.g. flu), across patients (due to different hospitals or hospital settings), and in their modality. Tackling the challenges posed by such scenarios is the focus of continual learning (CL) whereby a learner, when exposed to tasks in a sequential manner, is expected to perform well on current tasks without compromising performance on previously seen tasks. The outcome is a single algorithm that can reliably solve a multitude of tasks. Given the potential impact of designing such an algorithm and the machine learning community's efforts towards achieving artificial general intelligence, research on continual learning has increased [Parisi et al., 2019]. However, most, if not all, research in this field has been limited to a small handful of imaging datasets [Lopez-Paz and Ranzato, 2017, Aljundi et al., 2019a,b]. Although understandable from a benchmarking perspective, this approach fails to explore the utility and design of healthcare-specific continual learning methodologies. This is despite the potential impact of CL on medical diagnostics

as mentioned in Farquhar and Gal [2018]. To the best of our knowledge, we are the first to explore and propose a CL approach in the context of physiological signals.

The dynamic and chaotic environment that characterizes healthcare necessitates the availability of algorithms that are dynamically reliable; those that can adapt to potential covariate shift without catastrophically forgetting how to perform tasks from the past. Such dynamic reliability has a twofold effect. Firstly, a CL algorithm no longer needs to be retrained on data or tasks to which it has been exposed in the past, thus improving its data-efficiency. Secondly, consistently strong performance by an algorithm across a multitude of tasks increases its trustworthiness, which is a desirable trait sought by medical professionals [Spiegelhalter, 2020].

Our Contributions. In this paper, we propose a healthcare-specific replay-based continual learning methodology that is based on the following:

1. **Importance-Guided Storage:** task-instance parameters, a scalar corresponding to each instance in each task, as informative signals for *loss-weighting* and *buffer-storage*.

2. **Uncertainty-Based Acquisition:** an active learning inspired methodology that determines the degree of informativeness of an instance and thus acts as a *buffer-acquisition* mechanism.

2 Related Work

Continual learning (CL) has resurfaced in recent years with van de Ven and Tolias [2019] suggesting a three-tier categorization of methods: those based on 1) dynamic architectures, 2) regularization, and 3) memory. Although we only review memory-based methods as they are most similar to our approach, an extensive summary of all methods can be found in [Parisi et al., 2019]. Memory-based methods replay instances from previously seen tasks while training on the current task. For instance, in Learning without Forgetting (LwF) Li and Hoiem [2017], parameters from the previous task are used to generate soft targets for current task inputs, forming an auxiliary loss. Methods that involve a replay buffer include iCaRL Rebuffi et al. [2017], CLEAR Rolnick et al. [2019], GEM Lopez-Paz and Ranzato [2017], and aGEM Chaudhry et al. [2018] where the latter two naively populate their replay buffer with the last *m* examples observed for a particular task. A more sophisticated buffer storage strategy is employed by Isele and Cosgun [2018] and Aljundi et al. [2019a] where the latter solves a quadratic programming problem in the absence of task boundaries. By way of contrast, we propose task-instance parameters to guide the storage of instances from each task. Aljundi et al. [2019b] store instances using reservoir sampling and propose sampling instances that incur the greatest change in loss if parameters were to be updated on the subsequent task. Such a process is computationally expensive as it requires two forward and backward passes per batch. In contrast to previous research that independently investigated buffer storage and acquisition strategies, we focus on a *dual* storage and acquisition strategy. Moreover, the only work that focuses on CL for medical purposes is that of Lenga et al. [2020] wherein existing methodologies are simply implemented on chest X-ray datasets.

Active learning (AL) and healthcare have been relatively under-explored within machine learning. Early work in AL acquires instances using a mixture of Gaussians to minimize the variance of a learner [Cohn et al., 1996] and a support vector machine (SVM) to reduce the size of the version space [Tong and Koller, 2001]. A more complete review of active learning methodologies can be found in Settles [2009]. In the healthcare domain, Gong et al. [2019] propose a Bayesian deep latent Gaussian model to acquire important features from electronic health record (EHR) samples in the MIMIC dataset [Johnson et al., 2016] to improve mortality prediction. Also dealing with EHR data, Chen et al. [2013] use the distance of unlabelled samples from the hyperplane in an SVM to acquire datapoints. Wang et al. [2019] implement an RNN with active learning to acquire ECG samples during training. Zhou et al. [2017] propose using transfer learning in conjunction with a convolutional neural network to acquire biomedical images in an online manner. Smailagic et al. [2018] actively acquire unannotated medical images by measuring their distance in a latent space to images in the training set. Such similarity metrics, however, are sensitive to the amount of available labelled training data. This work was extended to the online domain by Smailagic et al. [2019]. Gal et al. [2017] adopt BALD [Houlsby et al., 2011] in the context of Monte Carlo Dropout to acquire datapoints that maximize the Jensen-Shannon divergence (JSD) across MC samples. BatchBALD [Kirsch et al., 2019] is an extension to this work which considers correlations between samples within the acquired batch. To the best of our knowledge, we are the first to employ AL-inspired acquisition functions in the context of CL.

3 Background

3.1 Continual Learning

In this work, we consider a learner $f_\omega : X \to Y$, a neural network parameterized by ω that, for each task $\mathcal{T} \in [1 \dots N]$, maps inputs $\boldsymbol{X_{\mathcal{T}}} \in \mathbb{R}^m$ of dimension m to outputs $\boldsymbol{Y_{\mathcal{T}}} \in [1 \dots C]$, where C is the number of classes. This learner is exposed to tasks sequentially whereby new tasks are only tackled once previous tasks are mastered. In this paper, we formulate our tasks based on a modification of the three-tier categorization of continual learning proposed by van de Ven and Tolias [2019]. In all of the following cases, task identities are *absent* during both training and testing and neural architectures are single-headed.

1. **Class Incremental Learning (Class-IL) -** in this case, mutually-exclusive pairs of classes belonging to the same dataset are presented to the learner in a sequential manner.
2. **Time Incremental Learning (Task-IL) -** in this case, although the same dataset and prediction problem are used for each task, the time of year at which the data were collected differs from one task to another. Such seasonality is most common in healthcare applications.
3. **Domain Incremental Learning (Domain-IL) -** in this case, although the same dataset and prediction problem are used for each task, the modality of the inputs differ from one task to another.

4 Methods

The two key ideas behind our proposal are the storage *and* acquisition of instances from a buffer such that destructive interference is mitigated. We describe these in more detail below.

4.1 Task-Instance Parameters, β

Task-instance parameters, $\beta_{i\mathcal{T}}$, are learnable parameters assigned to each instance, i, in each task, $\mathcal{T}$. We use them to weight instance losses and populate a replay buffer. Such a buffer, of finite size, $\mathcal{M}$, consists of instances that augment the training set of subsequent tasks.

Loss-Weighting Mechanism. We incorporate β as a coefficient of the loss, $\mathcal{L}_{i\mathcal{T}}$, for each instance of the current task. The objective function of a mini-batch of size, B, where B_k is the number of samples from each task, k, is as follows.

$$\mathcal{L} = \frac{1}{B_{\mathcal{T}}} \sum_{i=1}^{B_{\mathcal{T}}} \beta_{i\mathcal{T}} \mathcal{L}_{i\mathcal{T}} \tag{1}$$

In this setup, whereby $\frac{\partial \mathcal{L}}{\partial \beta} = \mathcal{L}_{i\mathcal{T}} > 0$, lower values of β are indicative of instances that are relatively difficult to train on. After discovering that β decays to zero and thus hinders learning, we introduced a regularization term that allows for informative loss signals while also maintaining differences between instances, as shown in Sec. 6.1.

$$\mathcal{L}_{\text{current}} = \frac{1}{B_{\mathcal{T}}} \sum_{i=1}^{B_{\mathcal{T}}} \beta_{i\mathcal{T}} \mathcal{L}_{i\mathcal{T}} + \lambda(\beta_{i\mathcal{T}} - 1)^2 \tag{2}$$

When $\mathcal{T} > 1$, extra loss terms are required to account for the instances that are replayed from the buffer (see Sec. 4.2). These instances, in contrast to those from the current task, are not weighted.

$$\mathcal{L}_{\text{replay}} = \frac{1}{B - B_{\mathcal{T}}} \sum_{k=1}^{\mathcal{T}-1} \sum_{i}^{B_k} \mathcal{L}_{ik} \tag{3}$$

$$\mathcal{L} = \mathcal{L}_{\text{current}} + \mathcal{L}_{\text{replay}} \tag{4}$$

Buffer-Storage Mechanism. Prior work suggests that task-instance parameters may converge to similar values [Saxena et al., 2019]. This would limit their ability to discriminate between instances

and the utility of the buffer. To circumvent this potential issue, we propose tracking the task-instance parameter after each training epoch, t, until the final epoch, τ, for the task at hand. This results in the storage function, $s_{i\mathcal{T}}$, approximated using the trapezoidal rule.

$$s_{i\mathcal{T}} = \int_0^\tau \beta_{i\mathcal{T}}(t)dt \approx \sum_{t=0}^\tau \left(\frac{\beta_{i\mathcal{T}}(t + \Delta t) + \beta_{i\mathcal{T}}(t)}{2} \right) \Delta t \tag{5}$$

Once training is complete on the task at hand, instances are ranked in descending order based on the value of the storage function, $s_{i\mathcal{T}}$, before the top b fraction are acquired. We opted for this strategy over storing the bottom b fraction due to its superiority, as shown in Sec. 6.4. Each task is allotted a fixed portion of the buffer.

4.2 Acquisition Function, α

Uncertainty-based acquisition functions such as BALD [Houlsby et al., 2011, Gal and Ghahramani, 2016] acquire unlabelled instances for which a set of decision boundaries disagree the most.

Buffer-Acquisition Mechanism. At epoch number, τ_{MC}, which we refer to as Monte Carlo (MC) epochs, each instance, x, in the buffer, $\mathcal{D}_B$, undergoes T forward passes through the network. Each forward pass is associated with a stochastic binary dropout mask and generates a posterior probability distribution over the classes, C. These distributions are stored in a matrix $G \in \mathbb{R}^{MxTxC}$. An acquisition function, BALD_{MCD}, is thus a function $\mathcal{F} : \mathbb{R}^{MxTxC} \to \mathbb{R}^M$.

$$\text{BALD}_{\text{MCD}} = \text{JSD}(p_1, p_2, \ldots, p_T) = \text{H}\left(p(y|x)\right) - \mathbb{E}_{p(\omega|D_{train})}\left[\text{H}\left(p(y|x, \hat{\omega})\right)\right] \tag{6}$$

where $\text{H}(p(y|x))$ represents the entropy of the posterior predictive distribution averaged across the MC samples, and $\hat{\omega} \sim p(\omega|D_{train})$ as in Gal and Ghahramani [2016]. At sample epochs, τ_S, instances are ranked in descending order based on the acquisition function before the top a fraction from each task in the buffer are sampled. This replay procedure exposes the network to instances from previous tasks that it is most confused about and thus nudges it to avoid destructive interference in a data-efficient manner. Algorithms 1-4 illustrate the entire procedure.

Algorithm 1: CLOPS

Input: MC epochs τ_{MC}, sample epochs τ_S, MC samples T, storage fraction b, acquisition fraction a, task data $\mathcal{D}_T$, buffer $\mathcal{D}_B$, training epochs per task τ

for $(x, y) \sim \mathcal{D}_T$ **do**
    calculate $\mathcal{L}$ using eq. 4
    update α_{ij}
    if epoch = τ **then**
        $\mathcal{D}_B$ = StoreInBuffer(α_{ij}, b)
    end if
    if epoch in τ_{MC} **then**
        G = MonteCarloSamples($\mathcal{D}_B$)
    end if
    if epoch in τ_S **then**
        $\mathcal{D}_T$ = AcquireFromBuffer($\mathcal{D}_B$, G, a)
    end if
end for

Algorithm 2: StoreInBuffer

Input: task-instance parameters $\beta_{i\mathcal{T}}$, b
calculate $s_{i\mathcal{T}}$ using eq. 5
SortDescending($s_{i\mathcal{T}}$)
$(x_b, y_b) \subset \mathcal{D}_T$
$\mathcal{D}_B \in (\mathcal{D}_B \cup (x_b, y_b))$

Algorithm 3: MonteCarloSamples

Input: $\mathcal{D}_B$
for $(x, y) \sim \mathcal{D}_B$ **do**
    for MC sample in T **do**
        obtain $p(y|x, \hat{\omega})$ and store in
        $G \in \mathbb{R}^{MxTxC}$
    end for
end for

Algorithm 4: AcquireFromBuffer

Input: $\mathcal{D}_B$, MC posterior distributions G, a
calculate β using eq. 6
SortDescending(β)
$(x_a, y_a) \subset \mathcal{D}_B$
$\mathcal{D}_T \in (\mathcal{D}_T \cup (x_a, y_a))$

---

5 Experimental Design

5.1 Datasets and Network

Experiments were implemented in PyTorch [Paszke et al., 2019] and given our emphasis on healthcare settings, we evaluate our approach on three publically-available datasets that include physiological

time-series data such as the electrocardiogram (ECG) alongside cardiac arrhythmia labels. We use $\mathcal{D}_1$ = Cardiology ECG [Hannun et al., 2019] (12-way), $\mathcal{D}_2$ = Chapman ECG [Zheng et al., 2020] (4-way), and $\mathcal{D}_3$ = PhysioNet 2020 ECG (9-way, multi-label). The same network was used for all formulations and can be found in Appendix B.

5.2 Hyperparameters

In all experiments, we chose the number of training epochs for each task, $\tau = 40$, as we found that to achieve strong performance on the validation set. We chose the MC epochs $\tau_{MC} = 40 + n$ and the sample epochs $\tau_S = 41 + n$ where $n \in \mathbb{N}^+$ in order to sample data from the buffer at every epoch following the first task. The only constraint that these values must satisfy is $\tau_S \geq \tau_{MC} > \tau$. For computational reasons, we chose the storage fraction $b = 0.1$ of the size of the training dataset and the acquisition fraction $a = 0.25$ of the number of samples per task in the buffer. To calculate the acquisition function, we chose the number of Monte Carlo samples, $T = 20$. We also explore the effect of changing these values on the performance. We chose the regularization coefficient, $\lambda = 10$.

5.3 Continual Learning Scenarios

Here, we outline the three primary continual learning scenarios we use for our experiments. Sample sizes corresponding to the tasks within these scenarios can be found in Appendix A.

1. **Class-IL** - in this case, dataset $\mathcal{D}_1$ is split according to mutually- exclusive pairs of classes $[0, 1]$, $[2, 3]$, $[4, 5]$, $[6, 7]$, $[8, 9]$, and $[10, 11]$. This scenario allows us to evaluate the sensitivity of a network to new classes.

2. **Time-IL** - in this case, dataset $\mathcal{D}_2$ is split into three tasks; Term 1, Term 2, and Term 3 corresponding to mutually-exclusive times of the year during which patient data were collected. This scenario allows us to evaluate the effect of temporal non-stationarity on a network's performance.

3. **Domain-IL** - in this case, dataset $\mathcal{D}_3$ is split according to the 12 leads of an ECG; 12 different projections of the same electrical signal generated by the heart. This scenario allows us to evaluate how robust a network is to the input distribution.

5.4 Baseline Methods

Multi-Task Learning (MTL) [Caruana, 1993] is a strategy whereby all datasets are assumed to be available at the same time and thus can be simultaneously used for training. Although this assumption may not hold in clinical settings due to the nature of data collection, privacy or memory constraints, it is nonetheless a strong baseline.

Fine-tuning is a strategy that involves updating all parameters when training on subsequent tasks as they arrive without explicitly dealing with catastrophic forgetting.

5.5 Evaluation Metrics

To evaluate our methods, we exploit metrics suggested by Lopez-Paz and Ranzato [2017] such as the average AUC and Backward Weight Transfer (BWT) and propose two additional evaluation metrics.

t-Step Backward Weight Transfer. To determine how performance changes 't-steps into the future', we propose BWT_t which evaluates the performance of the network on a previously-seen task, after having trained on t-tasks after it.

$$\text{BWT}_t = \frac{1}{N-t} \sum_{j=1}^{N-t} \text{R}_j^{j+t} - \text{R}_j^j \tag{7}$$

Lambda Backward Weight Transfer. BWT_t can be extended to all possible time-steps, t, to generate BWT_λ. Using this, one can identify potential improvements in methodology at the task-level.

$$\text{BWT}_\lambda = \frac{1}{N-1} \sum_{j=1}^{N-1} \left[\frac{1}{N-j} \sum_{t=1}^{N-j} \text{R}_j^{j+t} - \text{R}_j^j \right] \tag{8}$$

6 Experiments

6.1 Class-IL

Short and long-term destructive interference is experienced by a learner when exposed to binary tasks involving novel classes. In Fig. 1a, after achieving an AUC≈0.92 on the $[0, 1]$ (green) task, the network quickly forgets how to perform this task when shown the $[2, 3]$ (red) task, as shown by the significant drop in the AUC. Moreover, the final performance of the network at epoch 120 on the first task (AUC≈0.78) is below the best achieved performance. CLOPS, shown in Fig. 1b, alleviates this interference where on the $[0, 1]$ task, the AUC curve is 'flatter' and ends at a higher value (AUC≈0.85). Such improvements in BWT are quantified in Table 1. To pinpoint better the cause of this benefit, we perform ablation experiments in Section 6.5. Furthermore, task order can significantly affect the degree of destructive interference. We illustrate this phenomenon and the robustness of our method to task order in Appendix E.

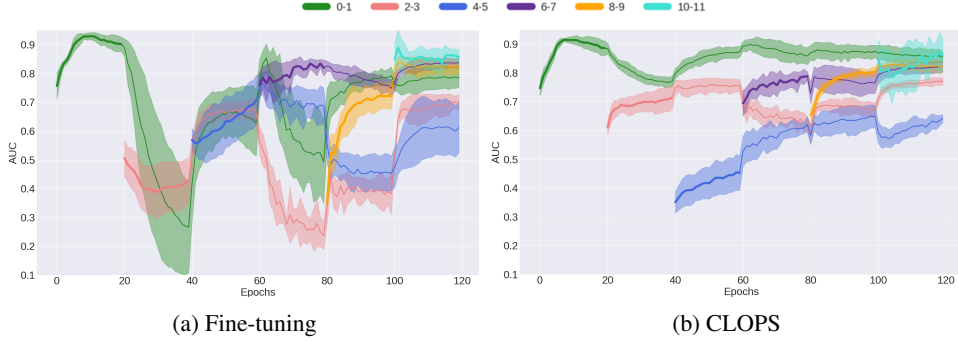(a) Fine-tuning　　　　　　　　　　　　　　(b) CLOPS

Figure 1: Mean validation AUC of a) fine-tuning and b) CLOPS strategy in the Class-IL scenario. Each task belongs to a mutually-exclusive pair of classes from $\mathcal{D}_1$. Storage and acquisition fractions are $b = 0.25$ and $a = 0.50$, respectively. Thicker curves indicate tasks on which the learner is currently being trained. The shaded area represents one standard deviation from the mean across 5 seeds.

Table 1: Performance of CL strategies in the Class-IL scenario. Storage and acquisition fractions are $b = 0.25$ and $a = 0.50$, respectively. Mean and standard deviation are shown across five seeds.

Method	Average AUC	BWT	BWT_t	BWT_λ
MTL	0.701 ± 0.014	-	-	-
Fine-tuning	0.770 ± 0.020	0.037 ± 0.037	$(0.076) \pm 0.064$	$(0.176) \pm 0.080$
CLOPS	$\mathbf{0.796 \pm 0.013}$	$\mathbf{0.053 \pm 0.023}$	$\mathbf{0.018 \pm 0.010}$	$\mathbf{0.008 \pm 0.016}$

Qualitative Assessment of Task-Instance Parameters. Tracked task-instance parameters, s, result in distinguishable distributions as shown in Fig. 2. Since lower s values are indicative of more difficult instances, the different distributions suggest that tasks differ in difficulty with $[6, 7]$ being the most difficult. To support our proposed interpretation of task-instance parameters, we plot two ECG tracings that correspond to the lowest and highest s values. Although both tracings belong to the class Normal Sinus Rhythm, the tracing with the lower s value contains more noise and exhibits an elongated P-wave, both potential sources of confusion for the network.

6.2 Time-IL

Time-incremental learning is arguably the most relevant to medical applications. Upon introducing the first two tasks, minimal destructive interference is observed. We show this in Fig. 3a where the AUC of Term 1 remains unchanged after having trained on Term 2. Such behaviour, in the context of Fig. 3b, is indicative that CLOPS does not disturb performance during these phases of the training procedure.
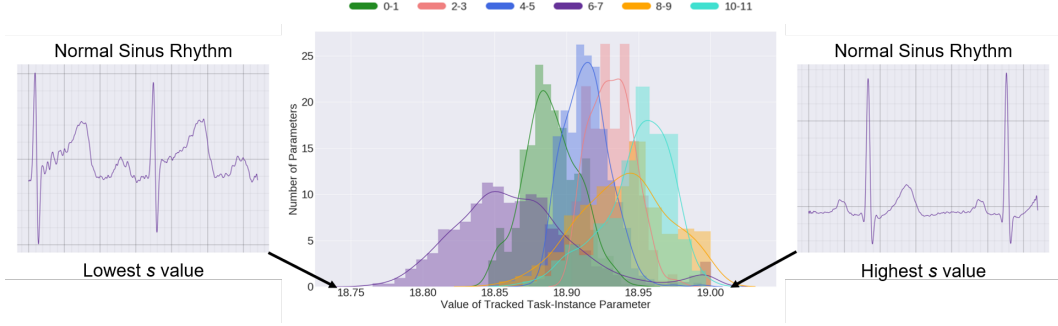
Figure 2: Distribution of the tracked task-instance parameter values, s, corresponding to CLOPS in the Class-IL scenario. Each colour corresponds to a different task. Storage and acquisition fractions are $b = 0.25$ and $a = 0.50$, respectively with results shown for one seed. The ECG tracings associated with the lowest and highest s values are shown.

Notably, this CL scenario sheds light on the forward weight transfer (FWT) capabilities of CLOPS. In the spirit of 'doing more with less', CLOPS generates an AUC≈0.62 on the Term 3 (blue) task after only one epoch of training. In contrast, the fine-tuning strategy requires a 20-fold increase in training time to reach that equivalent performance. We attribute this FWT to the loss-weighting role played by the task-instance parameters. By placing greater emphasis on more useful instances, the generalization performance of the algorithm is improved.
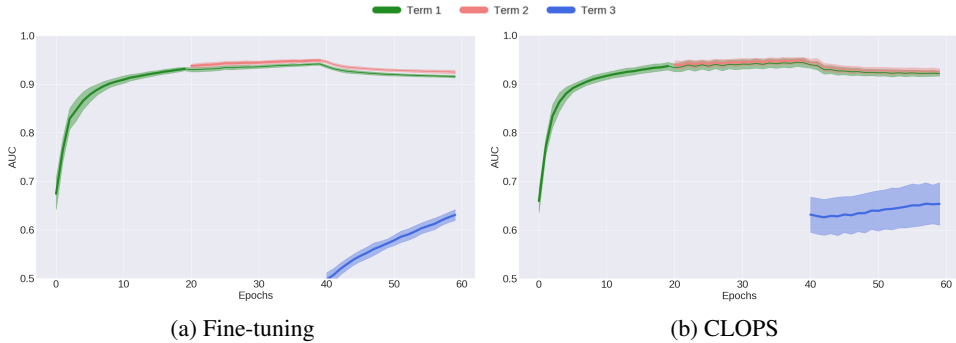

(a) Fine-tuning   (b) CLOPS

Figure 3: Mean validation AUC of (a) fine-tuning and (b) CLOPS strategy in the Time-IL scenario. Each task represents datasets collected at different times of the year. Thicker curves indicate tasks on which the learner is currently being trained. The shaded area represents one standard deviation from the mean across 5 seeds.

6.3 Domain-IL

When exposing a learner to ECG inputs belonging to different leads, a significant amount of destructive interference is experienced. This is supported by the large magnitude and negative BWT values shown for the fine-tuning strategy in Table 2. The reduction in the magnitude of the BWT values (less negative) once again illustrates the ability of CLOPS to mitigate destructive interference. This can be explained by the incorporation of instances from previous tasks about which the network is most confused. In doing so, the version space is adjusted to account for such instances, and thus allows for generalization to those tasks.

6.4 Effect of Storage Fraction, b, and Acquisition Fraction, a

Larger storage and acquisition fractions expose the learner to a more representative distribution of data from previous tasks. Arguably, this should further alleviate destructive interference and improve generalization performance. We quantify this graded response in Fig. 4a where the AUC increases from 0.758 to 0.832 as b and a increase from 0.1 to 1. We also claim that a performance bottleneck

Table 2: Performance of CL strategies in the Domain-IL scenario. Storage and acquisition fractions are $b = 0.25$ and $a = 0.50$, respectively. Mean and standard deviation are shown across five seeds.

Method	Average AUC	BWT	BWT_t	BWT_λ
MTL	0.730 ± 0.016	-	-	-
Fine-tuning	0.687 ± 0.007	$(0.041) \pm 0.008$	$(0.047) \pm 0.004$	$(0.070) \pm 0.007$
CLOPS	$\mathbf{0.731 \pm 0.001}$	$\mathbf{(0.011) \pm 0.002}$	$\mathbf{(0.020) \pm 0.004}$	$\mathbf{(0.019) \pm 0.009}$

lies at the storage phase. This can be seen by the larger improvement in performance as a result of an increased storage fraction compared to that observed for a similar increase in acquisition fraction. Despite this, a strategy with fractions as low as $b = 0.25$ and $a = 0.1$ is sufficient to outperform the fine-tuning strategy.

In addition to exploring the graded effect of fraction values, we wanted to explore the effect of storing the bottom, most difficult, b fraction of instances in a buffer. The intuition is that if a learner can perform well on these difficult replayed instances, then strong performance should be expected on the remaining relatively easier instances. We show in Fig. 4b that although the performance seems to be on par with that in Fig. 4a for $b = (0.5, 1)$, the most notable differences arise at fractions $b < 0.5, a < 0.5$ (red box). We believe this is due to the extreme 'hard-to-classify' nature of such replayed instances. These findings justify our storing of the top b fraction of instances.

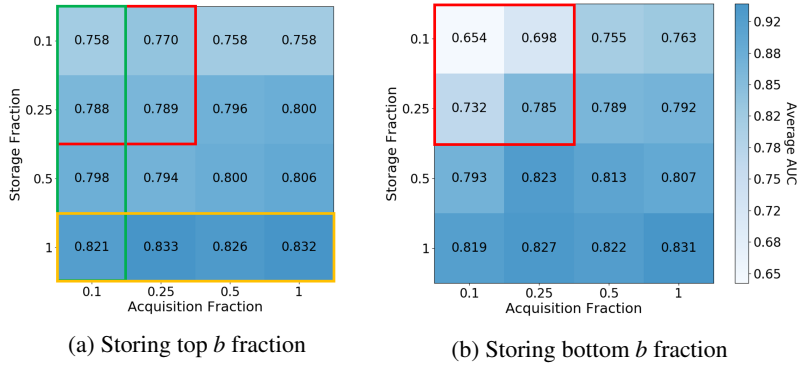(a) Storing top b fraction

(b) Storing bottom b fraction

Figure 4: Mean validation AUC of CLOPS in the Class-IL scenario at combinations of storage fractions, b, and acquisition fractions, a. Results are shown as an average across five seeds.

6.5 Effect of Task-Instance Parameters and Acquisition Function

To understand better the root cause of CLOPS' benefits, we conduct two ablation studies, 1) **Random Storage** dispenses with task-instance parameters and instead randomly stores instances into the buffer and 2) **Random Acquisition** dispenses with acquisition functions and instead randomly acquires instances from the buffer. We illustrate the effect of both of these strategies on generalization performance in Fig. 5. Their effect on BWT can be found in Appendix F.

Dispensing with task-instance parameters eliminates the graded performance response observed in Fig. 4a. This implies that, with CLOPS, more storage is better. To isolate the loss-weighting contribution of our parameters, we look to the $b = 1$ row (yellow rectangle) in Fig. 5a and Fig. 4a. At $b = 1$, $a = 0.5$, the absence of this loss-weighting mechanism worsens performance (AUC 0.826 vs. 0.770). We hypothesize that this mechanism is analogous to an attention mechanism placed on instance losses. Consequently, the network learns which instances to learn more from.

To isolate the effect of the acquisition function, we direct your attention to the $a = 0.1$ column (green rectangle) in Fig. 5b and Fig. 4a. At $b = 0.5$, $a = 1$, the absence of an acquisition function worsens performance (AUC 0.798 vs. 0.766). This makes sense as $BALD_{MCD}$ samples instances that lie in the region of uncertainty, thus reducing the size of the version space Settles [2009].
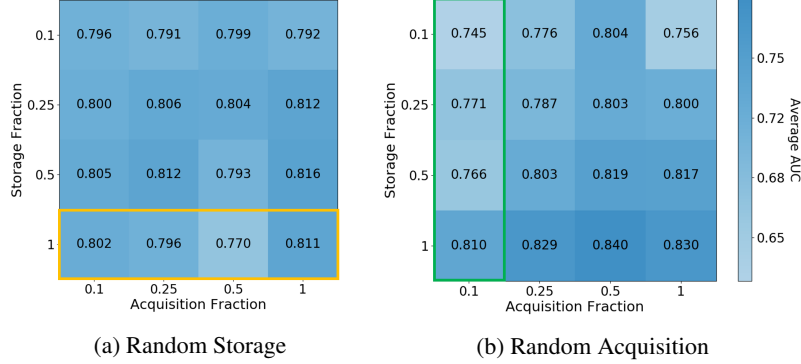
8

(a) Random Storage   (b) Random Acquisition

Figure 5: Mean validation AUC of random strategies in the Class-IL scenario at combinations of storage fractions b and acquisition fractions a. Results are shown as an average across five seeds.

6.6 Quantifying Task Similarity via Task-Instance Parameters, β

Inspired by the work of Silver and Mercer [1996], Nguyen et al. [2019], Rostami et al. [2020], we aim to use the tracked task-instance parameters, s, to compare tasks. To do so, we first fit a Gaussian to each of the distributions in Fig. 2. We then define cross-task similarity using the Hellinger distance, $\mathcal{D}_H \in [0, 1]$, between two Gaussian distributions parameterized by μ_0, σ_0^2 and μ_1, σ_1^2, respectively.

$$
\text{Cross Task Similarity} = 1 - \underbrace{\sqrt{1 - \sqrt{\frac{2\sigma_0\sigma_1}{\sigma_0^2\sigma_1^2}} e^{-\frac{1}{4}\frac{(\mu_0-\mu_1)^2}{\sigma_0^2\sigma_1^2}}}}_{\mathcal{D}_H} \tag{9}
$$

To validate our hypothesis, a learner is presented with an easy task followed by similar tasks. We choose the first task based on the highest mean s value and subsequent tasks based on cross-task similarity values (see Appendix K). These decisions are influence by curriculum learning [Bengio et al., 2009] and the intuition that training on similar tasks is less likely to lead to destructive interference. Indeed, by following this procedure, constructive interference is improved (BWT 0.053 vs. 0.087), as shown in Table 3. Such an outcome lends support to our proposed definition of task difficulty and similarity derived from task-instance parameters.

Table 3: Performance of CLOPS in the Class-IL scenario with different task orderings. Storage and acquisition fractions are $b = 0.25$ and $a = 0.50$, respectively. Results are shown across five seeds.

Task Order	Average AUC	BWT	BWT_t	BWT_λ
Random	**0.796 ± 0.013**	0.053 ± 0.023	0.018 ± 0.010	0.008 ± 0.016
Easy to Hard	0.744 ± 0.009	**0.087 ± 0.011**	**0.038 ± 0.021**	**0.076 ± 0.037**
Hard to Easy	0.783 ± 0.022	0.058 ± 0.016	(0.013) ± 0.013	(0.003) ± 0.014

7 Discussion and Future Work

In this paper, we introduced CLOPS, a healthcare-specific replay-based method to mitigate destructive interference during continual learning. CLOPS consists of an importance-guided buffer-storage and active-learning inspired buffer-acquisition mechanism. We showed that our approach can outperform multi-task training and a naive fine-tuning strategy when evaluated based on forward and backward weight transfer. Furthermore, our proposed task-instance parameters can assist with network interpretability, be used to quantify task difficulty and cross-task similarity, and guide training procedures positively. We now elucidate promising future paths.

Extensions to Task Similarity. The notion of task similarity was initially explored by Thrun and O'Sullivan [1996], Silver and Mercer [1996]. In this work we briefly proposed a novel definition of task similarity and used it to guide the order of presentation of tasks. Exploring more robust

definitions, validating them through domain knowledge, and exploiting them to further improve generalization is an exciting extension.

Dynamic Storage and Acquisition Fraction. In this work, the fraction of data stored into and sampled from the buffer was fixed throughout the lifetime of the learner. An interesting line of research could focus on identifying an optimal strategy that dynamically changes those fractions *during* training. Such decisions can be based on task similarity and the relative amount of data in each task.

Predicting Destructive Interference. This work, and most research in CL, deals reactively with the notion of destructive interference. We observe this phenomenon and attempt to design approaches to alleviate it. An impactful research question can focus on *predicting* the degree of forgetting that may occur if one where to sequentially train on a particular dataset. With such knowledge, one can deploy tools to alleviate the problem proactively.

8 Broader Impact

The exploration and design of continual learning algorithms in the context of healthcare is more critical than ever. Nowadays, common clinical scenarios involve heterogenous data streaming from a multitude of physiological sensors over time. Exploiting this data reliably without forgetting what was previously learned in a situation where privacy is of utmost importance will lead to more trustworthy algorithms. This, in turn, should help increase the adoption rate of clinical decision support systems by medical practitioners.

Although our work exploits several publically-available datasets and a diverse set of continual learning formulations, it still suffers from various limitations that may have dire consequences. Firstly, it does not explore destructive interference nor its alleviation for tasks beyond cardiac arrhythmia classification. Even in our existing formulations, destructive interference is not completely eradicated. As a result, the algorithm may generate incorrect predictions for patients seen in the past, thus negatively affecting clinical decision making and potentially patient outcomes.

As the US Food and Drug Administration begins to explore best practices for modifying algorithms Feng et al. [2019], we hope that our work will encourage researchers and policymakers to more seriously consider the role of continual learning in healthcare.

References

Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. The MIT Press, 2009.

German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 2019.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, pages 6467–6476, 2017.

Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. In *Advances in Neural Information Processing Systems*, pages 11816–11825, 2019a.

Rahaf Aljundi, Eugene Belilovsky, Tinne Tuytelaars, Laurent Charlin, Massimo Caccia, Min Lin, and Lucas Page-Caccia. Online continual learning with maximal interfered retrieval. In *Advances in Neural Information Processing Systems*, pages 11849–11860, 2019b.

Sebastian Farquhar and Yarin Gal. Towards robust evaluations of continual learning. *arXiv preprint arXiv:1805.09733*, 2018.

David Spiegelhalter. Should we trust algorithms? *Harvard Data Science Review*, 2(1), 1 2020. doi: 10.1162/99608f92.cb91a35a. URL https://hdsr.mitpress.mit.edu/pub/56lnenzj. https://hdsr.mitpress.mit.edu/pub/56lnenzj.

Gido M van de Ven and Andreas S Tolias. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*, 2019.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. In *Advances in Neural Information Processing Systems*, pages 348–358, 2019.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*, 2018.

David Isele and Akansel Cosgun. Selective experience replay for lifelong learning. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

Matthias Lenga, Heinrich Schulz, and Axel Saalbach. Continual learning for domain adaptation in chest x-ray classification. *arXiv preprint arXiv:2001.05922*, 2020.

David A Cohn, Zoubin Ghahramani, and Michael I Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145, 1996.

Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 2(Nov):45–66, 2001.

Burr Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison, Department of Computer Sciences, 2009.

Wenbo Gong, Sebastian Tschiatschek, Richard Turner, Sebastian Nowozin, and José Miguel Hernández-Lobato. Icebreaker: element-wise active information acquisition with bayesian deep latent gaussian model. *arXiv preprint arXiv:1908.04537*, 2019.

Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-III, a freely accessible critical care database. *Scientific Data*, 3:160035, 2016.

Yukun Chen, Robert J Carroll, Eugenia R McPeek Hinz, Anushi Shah, Anne E Eyler, Joshua C Denny, and Hua Xu. Applying active learning to high-throughput phenotyping algorithms for electronic health records data. *Journal of the American Medical Informatics Association*, 20(e2): e253–e259, 2013.

Guijin Wang, Chenshuang Zhang, Yongpan Liu, Huazhong Yang, Dapeng Fu, Haiqing Wang, and Ping Zhang. A global and updatable ecg beat classification system based on recurrent neural networks and active learning. *Information Sciences*, 501:523–542, 2019.

Zongwei Zhou, Jae Shin, Lei Zhang, Suryakanth Gurudu, Michael Gotway, and Jianming Liang. Fine-tuning convolutional neural networks for biomedical image analysis: actively and incrementally. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7340–7351, 2017.

Asim Smailagic, Pedro Costa, Hae Young Noh, Devesh Walawalkar, Kartik Khandelwal, Adrian Galdran, Mostafa Mirshekari, Jonathon Fagert, Susu Xu, Pei Zhang, et al. Medal: Accurate and robust deep active learning for medical image analysis. In *IEEE International Conference on Machine Learning and Applications*, pages 481–488, 2018.

Asim Smailagic, Pedro Costa, Alex Gaudio, Kartik Khandelwal, Mostafa Mirshekari, Jonathon Fagert, Devesh Walawalkar, Susu Xu, Adrian Galdran, Pei Zhang, et al. O-medal: Online active deep learning for medical image analysis. *arXiv preprint arXiv:1908.10508*, 2019.

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep Bayesian active learning with image data. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1183–1192. JMLR. org, 2017.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

Andreas Kirsch, Joost van Amersfoort, and Yarin Gal. Batchbald: efficient and diverse batch acquisition for deep bayesian active learning. *arXiv preprint arXiv:1906.08158*, 2019.

Shreyas Saxena, Oncel Tuzel, and Dennis DeCoste. Data parameters: A new family of parameters for learning a differentiable curriculum. In *Advances in Neural Information Processing Systems*, pages 11093–11103, 2019.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019.

Awni Y Hannun, Pranav Rajpurkar, Masoumeh Haghpanahi, Geoffrey H Tison, Codie Bourn, Mintu P Turakhia, and Andrew Y Ng. Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. *Nature Medicine*, 25(1):65, 2019.

Jianwei Zheng, Jianming Zhang, Sidy Danioko, Hai Yao, Hangyuan Guo, and Cyril Rakovski. A 12-lead electrocardiogram database for arrhythmia research covering more than 10,000 patients. *Scientific Data*, 7(1):1–8, 2020.

Richard A Caruana. Multitask connectionist learning. In *In Proceedings of the 1993 Connectionist Models Summer School*. Citeseer, 1993.

Daniel L Silver and Robert E Mercer. The parallel transfer of task knowledge using dynamic learning rates based on a measure of relatedness. In *Learning to learn*, pages 213–233. Springer, 1996.

Cuong V Nguyen, Alessandro Achille, Michael Lam, Tal Hassner, Vijay Mahadevan, and Stefano Soatto. Toward understanding catastrophic forgetting in continual learning. *arXiv preprint arXiv:1908.01091*, 2019.

Mohammad Rostami, David Isele, and Eric Eaton. Using task descriptions in lifelong machine learning for improved performance and zero-shot transfer. *Journal of Artificial Intelligence Research*, 67:673–704, 2020.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009.

Sebastian Thrun and Joseph O'Sullivan. Discovering structure in multiple learning tasks: The tc algorithm. In *ICML*, volume 96, pages 489–497, 1996.

Jean Feng, Scott Emerson, and Noah Simon. Approval policies for modifications to machine learning-based software as a medical device: A study of bio-creep. *arXiv preprint arXiv:1912.12413*, 2019.