FEDERATED CONTINUAL LEARNING GOES ONLINE: UNCERTAINTY-AWARE MEMORY MANAGEMENT FOR VISION TASKS AND BEYOND

Giuseppe Serra[1,2] **& Florian Buettner**[1,2,3]
serra@med.uni-frankfurt.de & florian.buettner@dkfz-heidelberg.de

ABSTRACT

Given the ability to model more realistic and dynamic problems, Federated Continual Learning (FCL) has been increasingly investigated recently. A well-known problem encountered in this setting is the so-called *catastrophic forgetting*, for which the learning model is inclined to focus on more recent tasks while forgetting the previously learned knowledge. The majority of the current approaches in FCL propose generative-based solutions to solve said problem. However, this setting requires multiple training epochs over the data, implying an offline setting where datasets are stored locally and remain unchanged over time. Furthermore, the proposed solutions are tailored for vision tasks solely. To overcome these limitations, we propose a new approach to deal with different modalities in the *online* scenario where new data arrive in streams of mini-batches that can only be processed once. To solve catastrophic forgetting, we propose an uncertainty-aware memory-based approach. Specifically, we suggest using an estimator based on the Bregman Information (BI) to compute the model's variance at the sample level. Through measures of predictive uncertainty, we retrieve samples with specific characteristics, and – by retraining the model on such samples – we demonstrate the potential of this approach to reduce the forgetting effect in realistic settings while maintaining data confidentiality and competitive communication efficiency compared to state-of-the-art approaches.

1 INTRODUCTION

In recent years, federated learning (FL) (McMahan et al., 2017) has received increasing attention for its ability to allow local clients work towards the same objective without the need for sharing limited and sensitive information. However, despite the benefits provided by the privacy-preserving collaborative training, the assumptions of the standard scenario are far from realistic (Babakniya et al., 2024). For example, the standard static-single-task framework has very limited practical applicability in real-world cases where local clients often need to continuously learn new tasks. Let us consider the problem of classifying new COVID-19 variants as illustrative use-case. Given the evolving nature of the virus, new variants (i.e., new classes) arise over time. In this context, healthcare facilities which collaborate using FL would fail since the standard setting does not consider the dynamic increment of new classes. For this reason, a new paradigm was recently introduced to model more complex dynamics: federated continual learning (FCL) (Yoon et al., 2021a; Dong et al., 2022). This new paradigm takes the global-local communication and privacy-preserving abilities enabled by FL and combines them with the Continual Learning (CL) ability of learning different tasks sequentially over time. However, FCL not only shares the characteristics at the intersection of FL and CL, but also inherits their respective challenges. In particular one of the most prominent problems in CL, the so-called *catastrophic forgetting* (CF) for which the model is prone to suffer from significant performance degradation on older tasks. In CL, this problem is addressed in different ways ranging from memory-based approaches (Yoon et al., 2021b; Kumari et al., 2022; Hurtado et al., 2023) to generative methods (Shin et al., 2017; Lesort et al., 2019) or to regularization techniques (Lee et al., 2017; Aljundi et al., 2018; Chaudhry et al., 2018b; He & Jaeger, 2018). In the context of FCL, instead, the majority of the proposed approaches exploit generative models at

[1] Goethe University Frankfurt,[2] German Cancer Consortium (DKTK),[3] German Cancer Research Center (DKFZ)

both local and global level to trace and encode the past information and generate synthetic instances which faithfully mimic previous history whilst preserving data privacy. A substantial drawback of such generative approaches is that they are tailored to image data and it is not clear how they could translate to other data modalities. Importantly, such generator-based approaches also imply an *offline* setting where task data are collected before training and remain unchanged over time: local generators require large training datasets of complete tasks and similarly, more recently proposed generative models based on data distillation can only be trained at the end of each task (Babakniya et al., 2024).

In realistic conditions however, all data from the current task may not be available for collection at the same time but may rather arrive in small chunks sequentially. Furthermore, given the ubiquity of smart and edge devices with limited capabilities (e.g., wearable devices), there is a need for updating the learning model with the new incoming data to minimize memory overload and communication bandwidth (Ma et al., 2022). This problem of learning from an *online* stream of data is well investigated in CL while it remains largely unexplored in FCL.

Inspired by the problem of training models over *online* streams of data, we first formalise a new scenario to tackle the online problem in FCL (online-FCL). In this new scenario, we assume each client to learn from a stream of data where new data arrive in mini-batches which can only be processed once. As such, in line with the definition of online-CL, the model is updated with high frequency (Soutif-Cormerais et al., 2023). Then, taking inspiration from the most popular solutions in online-CL, we introduce memory buffers to alleviate CF at local level. More in detail, we propose an uncertainty-based memory management where data points are stored in local buffers according to their predictive uncertainty. Intuitively, predictive uncertainty provides a glimpse of the samples' location in the decision space; samples with low uncertainty are the most representative ones for the respective class, while samples with high uncertainty represent data points that are close to the decision boundary and/or outliers. Thus, via estimates of predictive uncertainty, we can store in the memory samples with desired properties. Here, we propose to quantify uncertainty by directly estimating a generalized variance term from the loss function, which can be interpreted as a measure of epistemic uncertainty. To this end, we leverage a recently proposed bias-variance decomposition of the cross-entropy loss (Gruber & Buettner, 2023) and estimate the Bregman Information (BI) as the variance term in logit space.

The uncertainty-based memory management makes our approach flexible in terms of data modality. In fact, independent of the data modality (e.g., images, texts), our solution allows us to compute uncertainty estimates and thus populate the memory accordingly. This is in contrast to most of the current solutions in FCL which are limited to vision tasks and the offline setting.

In the last part of the paper, we demonstrate the ability of the proposed approach to reduce the forgetting effect in different scenarios. In the first part of the experiments, we evaluate our approach on CIFAR-10 and CIFAR-100 (Krizhevsky et al., 2009), two standard datasets used in this context. The goal is to understand how the proposed (epistemic) uncertainty estimate based on the Bregman Information performs in comparison with other standard estimates of overall model uncertainty under a memory-based regime. Then, departing from the common evaluation pipeline that would involve datasets like EMNIST (as in Qi et al. (2023); Wuerkaixi et al. (2024)) or larger datasets from the same naturalistic domain (e.g., ImageNet – as in Qi et al. (2023); Babakniya et al. (2024)), we validate our results on more probing real-world datasets from the medical domain. Finally, to showcase the ability of our approach to work with different data modalities, we test our findings on text classification tasks.

The contributions of this work can be summarized as follows:

- We propose and formalise a novel framework to tackle the FCL problem in the online setting which is largely overlooked in the literature.
- We highlight the limitations of current state-of-the-art generative-based solutions to work in the online setting and show empirically their inefficiency in the experimental section.
- We propose a memory-based solution that employs an alternative estimate for predictive uncertainty – which stems directly from a bias-variance decomposition of the cross-entropy loss for classification tasks – to populate the memory.
- We demonstrate the efficacy of our method in more realistic scenarios including datasets from different domains, with imbalance, and with different modalities.

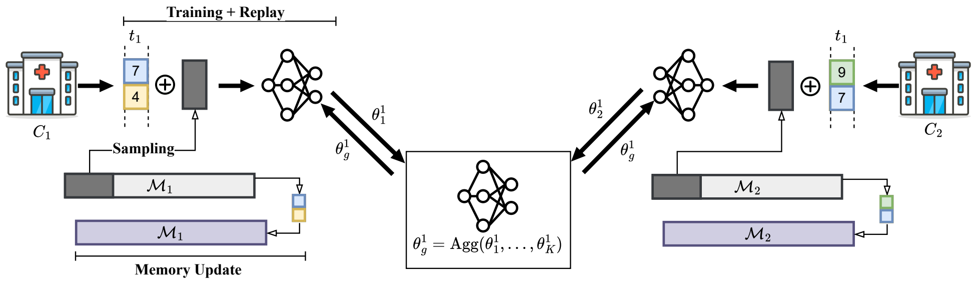

Figure 1: Schematic overview of the proposed online-FCL scenario. Each client $\mathcal{C}_k$ receives their data as a stream of batches. Each batch b_k^t is used for training and for updating the local memory $\mathcal{M}_k$. Once the current batch is processed, the client receives a new batch and continues their training without the possibility of retraining on previously seen batches.

2 Related Work

Continual Learning. There are three types of incremental learning (IL) (van de Ven et al., 2022); task-, domain-, and class-IL. In task-IL, training and testing data include explicitly task IDs. In this unrealistic scenario, the learning algorithm knows which task needs to be completed. In the domain-IL setup, instead, the classification problem remains the same while the input distribution (or domain) shifts over time. For instance, an example of domain drift appears when a model for detecting even/odd digits in images learns first $1s$ and $2s$ and then $4s$ and $5s$. Lastly, class-IL scenarios represent the most realistic setting where the model continuously learns from an increasing number of classes. CL problems can also be classified based on whether the data can be stored and viewed multiple times (*offline*) or whether they can be processed once (*online*) (Mai et al., 2022).

Class-Incremental Learning. Methods for class-IL can be divided into several categories (Mai et al., 2022) as follows; a) *Regularization techniques* alter the model parameter updates by adding penalty terms to the loss function (Lee et al., 2017; Aljundi et al., 2018), adjusting parameter gradients during optimization (Chaudhry et al., 2018b; He & Jaeger, 2018), or employing knowledge distillation (Rannen et al., 2017; Wu et al., 2019); b) *Memory-based techniques* exploit a fixed-size buffer containing samples from past tasks for replay (Aljundi et al., 2019; Chaudhry et al., 2019) or regularization (Nguyen et al., 2018; Tao et al., 2020); c) *Generative-based techniques* involve training generative models that can produce pseudo-samples mimicking data from the past (Shin et al., 2017; Lesort et al., 2019); d) *Parameter-isolation-based techniques* assign different model parameters to each task. This can be done either by activating only the relevant parameters for each task (Fixed Architecture) (Mallya & Lazebnik, 2018; Serra et al., 2018) or by adding new parameters and keeping the existing ones unchanged (Dynamic Architecture) (Aljundi et al., 2017; Yoon et al., 2018). In online-CL, where data arrive in single mini-batches and the model is updated frequently, rehearsal-based methods are favoured over more complex solutions, like generative methods, because they offer greater flexibility and require less training data and computational time (Mai et al., 2022).

In the context of memory-based techniques, the main question is how to optimally manage the memory. In order to reduce catastrophic forgetting, samples in the memory should be representative of their own class, discriminative towards the other classes, and informative enough for the model to recall the information about the old classes. In the literature, many conflicting strategies can be found. Some of them suggest to use the most representative samples (Yoon et al., 2021b; Hurtado et al., 2023), while others consider the samples near the decision boundary as the most useful to reduce CF (Kumari et al., 2022).

Federated Continual Learning. The intersection of federated learning and continual learning has only been investigated very recently. One of the first papers in this direction is Yoon et al. (2021a). The approach focuses on the less challenging task-IL scenario, where the task ID information is required during inference and testing. In Dong et al. (2022), the authors introduce the federated class-IL (FCIL) problem which tends to model more realistic scenarios. The clients have access to a large memory buffer and can share perturbed prototype samples with the global server, which differs with the standard FL setting where only the parameters of the local models are shared with the server (Babakniya et al., 2024). Ma et al. (2022) uses knowledge distillation on both local and global levels via unlabeled surrogate datasets. Other works (Hendryx et al., 2021; Qi et al., 2023; Babakniya et al., 2024; Wuerkaixi et al., 2024) pose their attention on the FCIL problem without the use of memory replay data. Hendryx et al. (2021) focus on few-shot learning and allows overlapping classes between tasks. Qi et al. (2023) instead, introduces the constraint of non-overlapping classes for intra-client tasks. The work is based on generative replay where clients train a discriminator and a generator locally. At the same time, at each communication round, the server performs a consolidation step and generates synthetic data using the locally trained generators. Following a similar principle, Babakniya et al. (2024) presents a generative model which is trained by the server in a data-free manner. Another FCIL scenario is considered in Shenaj et al. (2023), where tasks can arrive asynchronously at each client. The problem is tackled via a combination of prototype-based learning, representation loss, and stabilization of server aggregation.

Here we briefly summarise the identified limitations of such generator-based approaches. As anticipated before, they assume to work in the offline setting where static datasets are collected in advance and stored locally. This allows them to be trained over the same task dataset many times which is needed to reach convergence and to learn meaningful patterns in order to generate informative synthetic images. Finally, the storage of the whole task datasets and the generator models may be unfeasible in resource-limited devices.

3 Our Approach: O-FCL

3.1 Online-FCL: Problem Formulation

Following the notation proposed in Qi et al. (2023), we assume to have a set $\boldsymbol{\mathcal{C}} = \{\mathcal{C}_1, \ldots, \mathcal{C}_n\}$ of n different clients. Each client $\mathcal{C}_k$ keeps its private data $\boldsymbol{\mathcal{D}}_k = \{D_k^1, \ldots, D_k^t\}$ with its corresponding sequence of tasks $\boldsymbol{\mathcal{T}}_k = \{t_k^1, \ldots, t_k^t\}$. For each client k and task t, we have an associated dataset $D_k^t = \{(x_i, y_i)\}_{i=1}^{n_k^t}$ with x_i an input sample, y_i the corresponding class label, and n_k^t the number of training samples. In the proposed *online* setting, we assume that samples for each task t come gradually in a stream of mini-batches $b_k^t = \{(x_i, y_i) \in D_k^t\}_{i=1}^{bs}$ which can only be seen once. Similar to Qi et al. (2023), we assume non-overlapping classes for intra-client tasks. In other words, the clients can see a specific class only once during the training. At each communication round r, the client trains the local model on its own data and shares the local parameters $\boldsymbol{\theta}_k^r$ to update the global model parameters $\boldsymbol{\theta}_g^r$. Since data points can be processed only once, in order to exploit the information from the incoming data points at its best, communications with the server are performed when a single mini-batch or multiple consecutive mini-batches are processed. This is considerably different from the standard FCL scenario, where multiple iterations and communication rounds are performed for the *whole task dataset*. Following a popular trend in FCL, we also focus on the FCIL scenario which represents the most realistic and challenging case.

3.2 Methodology

As briefly anticipated above, the online nature of the newly introduced problem poses new challenges compared to the standard case. For this, we propose a memory-based approach on the client-side to alleviate catastrophic forgetting. The motivation is threefold: 1) the most effective solutions in online-CL are approaches with memory buffers; 2) given the online nature of the problem, generative-based approaches would be limited as they require many iterations over the same dataset; 3) compared with generative-based solutions, we only store a small amount of data reducing the overall overhead on the local device.

In the following subsections, we describe in more detail the characteristics of our approach at the client and server levels. For the client side, we outline our uncertainty-aware memory management detailing the properties of the employed uncertainty estimate and its advantages compared to standard scores. For the server side, we detail the adjustments done to improve the communication and parameter averaging effectiveness in the online scenario. A summary of the complete training procedure is reported in Algorithm 1 (see Appendix A.11).

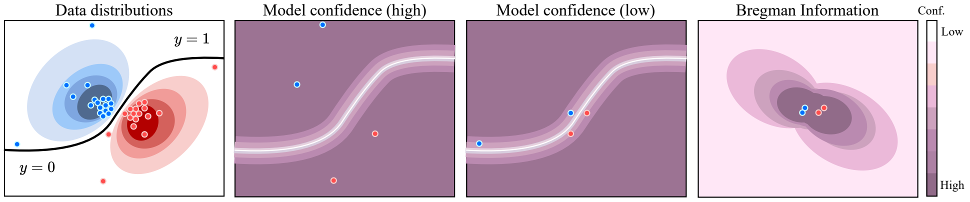

Figure 2: Illustration of the different aspects of uncertainty captured by model confidence scores and Bregman Information (BI). Close to the decision boundary we can notice how, due to the high aleatoric uncertainty, data points have low-confidence scores. Contrarily, due to high density of observed data, there is a low uncertainty about the data generating process resulting in a low BI.

3.2.1 Client-level

Memory management. For each client k, we introduce a fixed-size memory buffer $\mathcal{M}_k$. Similar to Chrysakis & Moens (2020), the memory population strategy is based on a class-balanced update, which is crucial to consider in case the stream of data is highly imbalanced. In fact, if classes are not equally represented in the memory, sampling from the memory may further deteriorate the predictive performance of the framework for under-represented classes. Differently from the populating strategy proposed in Chrysakis & Moens (2020), the criteria to decide which samples to keep in the memory is not random, but based on predictive uncertainty estimates in order to intentionally store samples with desired characteristics for each class. There are different ways to select the samples. We can decide to store a) the class-representative data points by selecting the least uncertain ones for each class (*bottom-k*); b) the easiest-to-forget samples by sampling the ones with high uncertainty (*top-k*).

From the second task on, we need a strategy for sampling from the memory a subset of data points used for replay (*replay set*). Since the memory represents a prototypical set of data points for each class, we assume that a random sampling is sufficient to extract informative data points from the memory. Following the standard practice, the number of samples in the replay set is equal to the batch size. It is important to note that the memory may already contain samples from the current task. In such cases, samples from the current task are excluded from the sampling process to ensure that the focus remains solely on those belonging to past tasks.

Predictive uncertainty estimation in the logit space. Different measure of predictive uncertainty can capture distinct aspects of a model's irreducible aleatoric uncertainty (inherent in the data) and its epistemic uncertainty (that stems from the finite training data and can be reduced by gathering more data). In online CL, the most commonly used measures are derived directly from the confidence scores of the model and mostly capture the irreducible aleatoric uncertainty (Wimmer et al., 2023). Here, we hypothesize it may be more beneficial for the model to replay instances which are representative in the sense that there is low uncertainty about the data generating process (we refer to this as low epistemic uncertainty, while acknowledging that there exist varying definitions). To not rely on specialized Bayesian models, we leverage a recently proposed bias-variance decomposition of the cross-entropy loss and compute a generalized variance term directly from the loss (Gruber & Buettner, 2023). Such bias-variance decomposition decomposes the expected prediction error (loss) into the model's bias, variance, and an irreducible error (noise term). The latter is related to aleatoric uncertainty, whereas the variance term can directly be related to epistemic uncertainty (Gruber et al., 2023). This is the first attempt of using this decomposition for memory management in FCL contexts.

Gruber & Buettner (2023) have recently shown that a bias-variance decomposition of cross-entropy loss gives rise to the Bregman Information as the variance term and measures the variability of the prediction in the logit space. We illustrate the different aspects of uncertainty captured by confidence scores and BI respectively in Figure 2. For example, data points close to the decision boundary have a low confidence score due to the inherently high aleatoric uncertainty; in contrast, due to the high density of observed data, there is actually a low uncertainty about the data generating process (DGP), resulting in a low BI (low epistemic uncertainty). Outliers far away from the decision boundary can have a high confidence score, but will have a high BI due to the high uncertainty regarding the DGP.

We hypothesize that it is samples with a low BI that are most useful for replay in online-FCL. To populate the memory with representative samples, we therefore propose using an uncertainty estimator based on Bregman Information (BI) (Gruber & Buettner, 2023). The authors demonstrate that BI at the sample level can be estimated through deep ensembles or test-time augmentation (TTA). However, to reduce the computational overhead at the local level, we employ TTA for computing the estimations. Let us consider the problem of multi-class classification (as in our case), where the standard loss is represented by the cross-entropy. Considering a set P of perturbations for a given data point x, we can compute the variance term of the classification loss $u(x)_{BI}$ as follows:

$$u(x)_{BI} = \frac{1}{P}\sum_{i=1}^{P} \text{LSE}(\hat{z}_i) - \text{LSE}\left(\frac{1}{P}\sum_{i=1}^{p} \hat{z}_i\right), \tag{1}$$

where $\hat{z}_i \in \mathbb{R}^c$ represent the logit predictions and $\text{LSE}(x_1, \ldots, x_n) = \ln \sum_{i=1}^{n} e^{x_i}$ the *LogSumExp* (LSE) function respectively. Intuitively, a large value of $u(x)_{BI}$ means that the logits predicted across the perturbations vary significantly, suggesting a high uncertainty of the prediction and the DGP at this point of the input space. The use of this estimator is motivated by the fact that, in comparison with other uncertainty scores such as entropy, smallest margin, or least confidence, there is no information loss in the estimation step. In fact, if we inspect these alternative metrics (see Appendix A.1 for the equations), we can notice that they either need a normalization step to move the logits in the probability space or rest on the largest activation value only. Furthermore, as reported in Gruber & Buettner (2023); Ovadia et al. (2019), and Tomani & Buettner (2021), common confidence scores are reliable only in case of well-calibrated models. In contrast, the BI-based estimation of the epistemic uncertainty is meaningful also under distribution shift and able to identify robust and representative samples (Gruber & Buettner, 2023).

3.2.2 Global-level

Communication rounds. Following the standard assumptions of the federated scenario, we allow local clients to share the model parameters solely. However, the proposed online framework poses additional challenges compared to the standard case. In the standard scenario, the communication round is performed at the end of several iterations over the same task dataset. This implies that the model has probably reached convergence and can effectively share the learned information during the communication round. In our case, a few new samples are available at each step and, to keep the model up-to-date, the communication round cannot be performed at the end of the task only. For this reason, as anticipated in Section 3.1, communications with the server are performed when a single mini-batch or multiple consecutive mini-batches are processed. Given the instability of the model when a new task starts, to ensure an effective and efficient information sharing, we propose to set a *burn-in* period. During this period, the local model learns independently for a certain number of batches without sharing and receiving information from the others. This guarantees that, when the local client starts to be involved in the communication rounds, the model has effectively learned relevant information on the current task. Additionally, since every parameter update degrades the predictive performance at local level (Qi et al., 2023), we propose to limit the number of communications rounds per task. Instead of performing a model averaging every time a batch is processed, we let the local models learn for q consecutive mini-batches before actively participating to the communication round.

Parameter averaging. Although clients work on the same task, they may receive data coming from different classes where some of them are over-represented compared to others. In this case, the local parameters collected for parameter averaging can be *biased* towards the most common classes. For this, we suggest to first create an aggregated model for each class available in the current round, and then compute $\boldsymbol{\theta}_g^r$ using the class-based models just created. This way, the current classes contribute equally during the updates of the global parameters. In case all the clients receive data containing classes equally distributed, the parameter averaging results in the standard computation, like e.g., FedAvg (McMahan et al., 2017). Importantly, if sharing the class information is not possible, we can rely on standard averaging strategies (e.g., FedAvg or FedProx (Li et al., 2020)) without hampering the performance since our methodology is flexible on this matter (see Table 9). Finally, once the new global parameters $\boldsymbol{\theta}_g^r$ are computed, to avoid a drastic change between the old parameters and the new ones, following Shenaj et al. (2023), we propose to average the newly computed parameters at round r with the previous parameters computed at round $r - 1$.

4 Experiments

4.1 Datasets and Settings

Datasets. To understand the behaviour of different algorithms in the online-FCL scenario, we use two datasets commonly used in the literature, namely, CIFAR10 and CIFAR100 (Krizhevsky et al., 2009), and TinyImageNet (Le & Yang, 2015). We randomly assign a set of classes to 5 tasks (10 for TinyImageNet) and we split the task data evenly among the clients such that the task sequence T_k^t for task t and client k does not share any data with the other clients. In this way, since the class assignment per task is different every time, we can identify the strategy providing greater flexibility and effectiveness irrespective of the composition of the tasks. Then, as anticipated in the introductory part, we assess the performance under more difficult and realistic conditions. Instead of using datasets that, to some extent, share similar characteristics with CIFAR (i.e., ImageNet) or represent unrealistic tasks (such as EMNIST), we validate our results on datasets for biomedical image analysis. Apart from the change in the domain (which in turn means different backgrounds in the images, different statistics, etc.), these datasets pose an additional challenge compared to the standard benchmark datasets, i.e., data imbalance. To reflect realistic conditions where recent tasks contain fewer data points than the older ones since the time to collect them is shorter, we assign classes to tasks based on the class size. We decide to focus on two biomedical datasets annotated by expert clinical pathologists; the colorectal cancer histology (CRC-Tissue) dataset (Kather et al., 2019; Yang et al., 2023) containing images divided in 8 classes of hematoxylin–eosin (HE)–stained slides taken from patients with colorectal cancer (CRC), and the kidney cortex cells (KC-Cell) dataset containing 8 classes of human kidney cortex tissue sections (Ljosa et al., 2012; Yang et al., 2023). Finally, to evaluate our approach on text classification tasks, we employ three different datasets: the 20NewsGroups dataset (Lang, 1995), DBPedia (Zhang et al., 2015), and Yahoo Answers (Zhang et al., 2015). As for vision tasks, we randomly assign classes to tasks every run. Statistics and more details about the used datasets can be found in the supplementary material (Appendix A.3).

Experimental settings. Given the novelty of the online setting, there are no direct competitors which we could refer to. We therefore investigate the most successful algorithms from the online-CL and the FCL literature. We compare our approach with standard baselines for FL, i.e., FedAvg (McMahan et al., 2017) and FedProx (Li et al., 2020), a standard memory-based approach for online-CL, namely Experience Replay (ER) (Chaudhry et al., 2019), and two state-of-the-art approaches for FCL with generative replay, namely MFCL (Babakniya et al., 2024) and FedCIL (Qi et al., 2023). Our decision to use ER is based on the fact that, despite its simplicity, it is surprisingly competitive when compared with more sophisticated and newer approaches as shown in recently conducted empirical surveys (Soutif-Cormerais et al., 2023). For FCL, we are particularly interested in MFCL because its data-free solution trains the generator on the server side. As such, we believe it may be potentially feasible in the online scenario, in contrast to other approaches such as FedCIL that train local generators at the client-side (Qi et al., 2023). In addition to the standard ER, we also include the class-balanced version (CBR) presented in Chrysakis & Moens (2020). Finally, to show the competitiveness of $u(x)_{BI}$, we also consider other uncertainty scores. In particular, least confidence (LC), margin sampling (MS), ratio of confidence (RC), and entropy (EN) (Shannon, 1948; Campbell et al., 2000; Culotta & McCallum, 2005).

In all the experiments for image classification, we employ a slim version of Resnet18 (He et al., 2016) – as done in previous works in online-CL (Kumari et al., 2022; Hurtado et al., 2023; Soutif-Cormerais et al., 2023) –, and use the SGD optimizer with a learning rate of 0.1. For text classification, we employ a simple MLP with a single fully connected hidden layer (512 units) and Adam optimizer with a learning rate of 0.01. To generate the perturbed inputs for estimating predictive uncertainty, we use two different strategies according to the different data modalities. For vision tasks, the perturbations are generated via standard augmentation. The list of augmentations used in our experiments is provided in Appendix A.2. For the natural language experiments, we leverage recent progress in foundation models and first create general-purpose latent representations of the input texts via a pre-trained sentence embedder. In particular, we employ *e5-small-v2* (Wang et al., 2022) (384 dimensions) via HuggingFace (Wolf et al., 2020). Then, for perturbing the generated vector representations, we add Gaussian noise from $\mathcal{N}(0, 0.1)$ to each latent dimension.

Following standard practice, we set the batch size equal to 10. The memory size was set to different values for each dataset in order to examine the performance of various memory configurations,

Table 1: Comparison of average last accuracy (A) and last forgetting (F) on CIFAR10 (5 tasks).

Score		M=200		M=500		M=1000	
ER		20.74 ± 2.53 48.48 ± 3.95		27.90 ± 2.03 30.18 ± 2.69		33.64 ± 0.72 24.3 ± 1.73	
CBR		22.82 ± 1.55 46.79 ± 2.92		24.17 ± 3.23 28.54 ± 2.22		32.67 ± 1.84 23.62 ± 2.63	
		Top	Bottom	Top	Bottom	Top	Bottom
LC	A(↑)	21.58 ± 1.56	20.92 ± 0.70	29.49 ± 1.46	27.38 ± 1.68	36.14 ± 2.44	31.94 ± 2.44
	F(↓)	50.78 ± 1.96	49.78 ± 3.16	36.13 ± 2.54	30.17 ± 2.52	21.97 ± 4.06	29.53 ± 4.46
MS	A(↑)	22.34 ± 1.04	21.88 ± 1.28	29.72 ± 1.98	28.96 ± 1.54	34.42 ± 0.91	32.28 ± 1.20
	F(↓)	49.38 ± 3.09	47.20 ± 4.97	35.16 ± 3.20	28.07 ± 2.71	29.58 ± 2.73	28.64 ± 4.34
RC	A(↑)	21.80 ± 1.65	22.75 ± 1.84	28.36 ± 2.07	**31.17** ± 1.75	35.08 ± 1.11	28.98 ± 2.26
	F(↓)	56.91 ± 1.90	45.05 ± 2.11	36.47 ± 2.84	31.86 ± 4.16	26.10 ± 1.76	28.39 ± 2.95
EN	A(↑)	21.15 ± 1.37	20.14 ± 1.41	28.63 ± 0.86	26.48 ± 1.59	**36.25** ± 1.22	26.42 ± 1.49
	F(↓)	53.30 ± 2.97	55.58 ± 3.49	36.70 ± 2.21	35.85 ± 3.17	25.52 ± 3.01	34.64 ± 3.40
BI	A(↑)	21.31 ± 1.46	**24.89** ± 0.83	26.57 ± 0.83	27.84 ± 2.31	35.12 ± 2.51	35.83 ± 2.60
	F(↓)	54.65 ± 1.83	**35.77** ± 4.13	42.86 ± 3.81	**24.59** ± 3.16	27.00 ± 2.99	**19.07** ± 2.17

including both large and small buffers. The burn-in period is set to 30. A communication round with the server is performed after $q = 5$ mini-batches. The number of clients is set to 5. For parameter averaging, we employ FedAvg (McMahan et al., 2017). Given the flexibility of the approach, as mentioned before, the model averaging strategy can be changed as desired. For evaluation, following recent works (Yoon et al., 2021a; Wuerkaixi et al., 2024), we use the average last accuracy (A) and average forgetting (F) (see appendix A.5 for the definitions). All the experiments are run on three different random seeds. For each dataset, experiments were run on a Linux machine using a single Quadro RTX 5000 and 16 GB RAM.

4.2 Empirical Results

From the experiments conducted on the CIFAR datasets, we can observe that the proposed approach is able to reduce CF consistently across different memory sizes and class-per-task assignments. The results in Tables 1 and 2 suggest that storing the class-representative samples (i.e., the least uncertain data points for each class) provides a benefit in terms of predictive performance gain and forgetting reduction. Our findings are substantiated in the classification tasks for biomedical images. Although the images originate from another domain, Table 3 confirms that in terms of CF, BI outperforms the considered memory-based baselines in most of the cases. Crucially, we also show the ability of our approach to work with imbalanced real-world data. Table 5 and Figure 3 summarise our findings; in comparison with both memory-based and generative-based methods, our simple-yet-elegant approach is able to perform consistently across tasks (left plot) and reduce CF on different datasets (right plot). Finally, we demonstrate that our method outperforms baselines also on non-vision data. Table 4 reports the results on the textual classification datasets. Here we find that the difference in terms of predictive accuracy is less marked. This is because the baselines have higher accuracy on the last tasks and poor performance on the first ones, while in our case the accuracy is kept more uniform across the tasks (reflected in a lower average forgetting) as shown in the left plot in Figure 3. Note that the use of generative baselines is restricted to image data only. Results for TinyImageNet are reported in Appendix A.4 - Table 7.

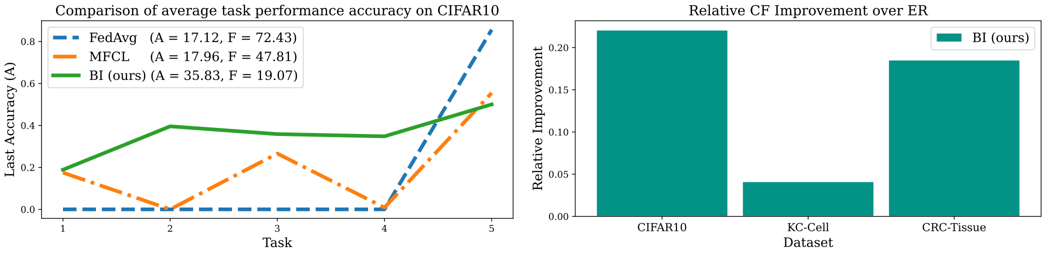

Figure 3: (left) Average accuracy per task in comparison with FL (FedAvg) and generative-based FCL baselines (MFCL); (right) Relative CF improvement over the CL baseline (ER).

Table 2: Comparison of last accuracy (A) and last forgetting (F) on CIFAR100 (5 tasks).

Score		M=1000		M=2000	
ER		10.33 ± 0.58		10.95 ± 0.4	
		6.89 ± 1.53		9.12 ± 0.96	
CBR		13.38 ± 0.58		11.41 ± 0.36	
		31.66 ± 0.30		9.37 ± 1.56	
		Top	Bottom	Top	Bottom
LC	A(↑)	12.71 ± 0.96	13.43 ± 1.00	13.50 ± 0.81	14.09 ± 0.47
	F(↓)	8.03 ± 1.88	6.95 ± 0.91	7.99 ± 1.34	7.75 ± 1.87
MS	A(↑)	13.34 ± 1.71	13.77 ± 1.31	13.83 ± 0.56	14.26 ± 0.72
	F(↓)	7.48 ± 2.04	6.84 ± 1.60	8.04 ± 0.99	8.01 ± 1.76
RC	A(↑)	13.30 ± 1.47	13.76 ± 1.24	13.28 ± 1.23	13.83 ± 0.69
	F(↓)	8.20 ± 1.91	6.97 ± 1.12	7.52 ± 1.70	7.70 ± 1.58
EN	A(↑)	12.77 ± 1.00	12.77 ± 0.71	13.81 ± 1.00	13.93 ± 0.76
	F(↓)	8.11 ± 1.14	7.90 ± 1.20	8.10 ± 1.51	7.50 ± 1.17
BI	A(↑)	12.89 ± 0.97	**14.04** ± 0.62	13.72 ± 0.70	**14.31** ± 0.76
	F(↓)	7.54 ± 1.43	**6.73** ± 1.76	8.06 ± 1.49	**6.96** ± 1.40

Table 3: Comparison of average last accuracy (A) and last forgetting (F) on (left) CRC-Tissue (4 tasks), and (right) KC-Cell (4 tasks).

CRC-Tissue

Score		M=80		M=120	
ER		49.79 ± 1.93		50.68 ± 2.69	
		20.31 ± 8.76		23.96 ± 4.78	
CBR		46.70 ± 1.34		49.38 ± 1.01	
		33.69 ± 3.16		19.19 ± 6.86	
		Top	Bottom	Top	Bottom
LC	A(↑)	47.66 ± 2.52	56.02 ± 4.89	56.49 ± 2.21	58.12 ± 2.94
	F(↓)	41.42 ± 3.64	13.25 ± 4.91	26.65 ± 6.30	11.13 ± 3.12
MS	A(↑)	47.87 ± 1.95	54.92 ± 2.93	52.82 ± 3.51	55.64 ± 4.42
	F(↓)	36.96 ± 4.74	17.24 ± 8.46	30.42 ± 4.05	9.59 ± 6.93
RC	A(↑)	50.42 ± 3.92	**57.50** ± 3.01	55.93 ± 5.85	58.81 ± 3.34
	F(↓)	32.95 ± 4.04	**12.91** ± 5.00	26.57 ± 6.35	8.71 ± 5.86
EN	A(↑)	49.18 ± 2.36	55.22 ± 4.05	53.80 ± 4.56	58.48 ± 3.06
	F(↓)	29.55 ± 6.39	14.95 ± 8.85	29.48 ± 12.65	8.16 ± 3.79
BI	A(↑)	49.18 ± 2.61	57.05 ± 3.65	50.16 ± 2.79	**62.33** ± 1.93
	F(↓)	36.01 ± 7.81	13.63 ± 8.04	37.28 ± 9.50	**7.99** ± 4.08

KC-Cell

Score		M=120		M=160	
ER		19.59 ± 1.32		**22.29** ± 0.71	
		65.00 ± 3.68		60.33 ± 6.0	
CBR		18.69 ± 1.56		20.38 ± 1.54	
		66.25 ± 2.36		62.90 ± 4.43	
		Top	Bottom	Top	Bottom
LC	A(↑)	17.94 ± 1.19	19.66 ± 0.83	19.18 ± 1.31	21.32 ± 1.66
	F(↓)	67.25 ± 3.88	65.16 ± 4.07	62.90 ± 2.48	62.84 ± 4.47
MS	A(↑)	17.46 ± 0.89	20.05 ± 1.79	19.30 ± 1.01	21.39 ± 0.96
	F(↓)	66.90 ± 3.52	62.22 ± 4.61	65.13 ± 5.05	65.63 ± 2.85
RC	A(↑)	18.14 ± 1.69	17.83 ± 1.20	19.59 ± 0.88	21.21 ± 1.23
	F(↓)	66.12 ± 3.44	68.04 ± 3.76	63.50 ± 3.76	63.27 ± 9.27
EN	A(↑)	17.64 ± 1.63	19.46 ± 1.54	18.94 ± 1.23	21.25 ± 1.17
	F(↓)	68.34 ± 4.37	64.66 ± 4.48	65.57 ± 4.24	64.63 ± 5.01
BI	A(↑)	16.63 ± 0.73	**20.91** ± 1.32	18.03 ± 0.61	21.61 ± 1.23
	F(↓)	72.63 ± 1.45	**61.03** ± 4.42	67.90 ± 3.89	**59.05**± 4.42

Table 4: Comparison of average last accuracy (A) and last forgetting (F) on text classification tasks.

Dataset			ER	CBR	LC	MS	RC	EN	BI
20NewsGroup	M=60	A(↑)	42.33 ± 1.66	**45.21** ± 1.86	42.90 ± 1.23	44.14 ± 1.63	44.15 ± 1.32	41.65 ± 1.82	44.92 ± 1.63
		F(↓)	31.12 ± 2.18	30.39 ± 0.80	32.05 ± 2.27	31.67 ± 2.57	31.61 ± 1.77	33.77 ± 1.43	**29.98** ± 1.37
	M=500	A(↑)	45.95 ± 1.76	46.22 ± 1.55	45.72 ± 2.28	46.46 ± 2.53	46.58 ± 2.83	44.92 ± 2.71	**46.72** ± 2.11
		F(↓)	27.43 ± 1.12	28.00 ± 0.98	28.58 ± 1.61	27.97 ± 2.36	28.02 ± 2.53	29.63 ± 2.45	**26.97** ± 2.36
DBPedia	M=60	A(↑)	75.99 ± 2.10	76.24 ± 1.21	74.73 ± 2.04	75.21 ± 1.92	75.36 ± 1.61	74.58 ± 1.88	**77.86** ± 1.95
		F(↓)	24.73 ± 2.70	23.06 ± 1.58	27.06 ± 2.30	26.44 ± 2.28	26.52 ± 1.77	27.14 ± 2.15	**23.06** ± 2.10
	M=100	A(↑)	76.78 ± 1.04	**79.78** ± 0.83	76.21 ± 1.58	76.18 ± 1.70	76.18 ± 1.50	76.33 ± 1.45	78.68 ± 1.08
		F(↓)	24.14 ± 1.18	**21.45** ± 1.49	25.15 ± 1.60	25.19 ± 1.83	25.29 ± 1.78	24.87 ± 1.83	21.79 ± 1.41
Yahoo Answers	M=500	A(↑)	45.80 ± 1.77	47.45 ± 2.86	47.87 ± 3.38	47.60 ± 3.38	48.25 ± 2.61	49.32 ± 3.61	**50.13** ± 1.69
		F(↓)	40.15 ± 2.91	36.38 ± 4.29	36.02 ± 6.81	35.69 ± 4.61	34.60 ± 3.49	33.67 ± 5.12	**32.65** ± 3.21
	M=1000	A(↑)	45.48 ± 2.11	44.62 ± 3.23	46.40 ± 3.07	47.00 ± 2.75	46.33 ± 3.68	46.70 ± 1.11	**47.03** ± 3.36
		F(↓)	39.40 ± 3.47	39.17 ± 4.16	37.15 ± 5.78	37.15 ± 4.35	38.08 ± 5.26	37.85 ± 1.78	**37.25** ± 4.17

5 Discussion and Conclusion

The novelty of the online scenario for FCL tasks raises questions on different aspects of the proposed approach and we next provide an in-depth discussion of key methodological choices and findings.

Ablation studies - Appendix A.7. We conduct several ablation studies to assess the impact of different hyperparameters (i.e., burn-in period, jump parameter, parameter averaging strategy, and augmentation set) on the learning performance of the proposed approach while also providing insights and explanations for usage best practices in this newly introduced context. One key aspect is the set of augmentations. From the results, the number of augmentations is important to improve the uncertainty estimation and, consequently, the performance of the approach. In terms of the choice of modality-specific augmentations, we rely on the large body of literature on TTA and show that stan-

Table 5: Comparison of average last accuracy (A) and last forgetting (F) on vision and textual tasks for standard FL, generative-based FCL, and our method.

Method		CIFAR10	CIFAR100	CRC-Tis.	KC-Cell	20News	DBPedia	Yahoo
FedAvg	A(↑)	16.90 ± 0.55	3.57 ± 0.33	22.26 ± 2.29	15.04 ± 0.83	6.34 ± 0.58	19.57 ± 0.30	14.18 ± 1.59
	F(↓)	78.29 ± 2.60	19.73 ± 0.99	87.39 ± 4.06	74.08 ± 3.70	58.84 ± 1.91	99.02 ± 0.32	70.90 ± 2.88
Weighted FedAvg	A(↑)	16.51 ± 0.77	4.22 ± 0.20	22.00 ± 2.40	14.85 ± 0.92	6.36 ± 0.56	19.23 ± 0.44	12.52 ± 1.60
	F(↓)	70.08 ± 1.57	20.18 ± 0.23	80.40 ± 5.15	76.21 ± 2.72	50.22 ± 2.41	97.64 ± 1.25	69.00 ± 2.11
FedProx	A(↑)	16.57 ± 0.50	3.73 ± 0.15	22.12 ± 1.26	14.79 ± 1.01	12.25 ± 0.40	19.58 ± 0.26	17.35± 1.34
	F(↓)	76.74 ± 1.12	20.26 ± 0.58	87.86 ± 4.78	73.38 ± 3.66	67.31 ± 3.14	99.06 ± 0.32	82.54 ± 0.55
MFCL	A(↑)	16.42 ± 0.32	4.60 ± 0.09	27.62 ± 7.99	16.32 ± 1.21	✗	✗	✗
	F(↓)	55.64 ± 3.03	13.69 ± 0.33	76.34 ± 12.68	70.88 ± 1.94			
FedCIL	A(↑)	16.26 ± 1.16	2.45 ± 0.28	22.44 ± 1.23	13.14 ± 1.31	✗	✗	✗
	F(↓)	49.87 ± 0.61	**4.99** ± 0.81	62.47 ± 1.35	**55.82**± 1.60			
BI (best)	A(↑)	**35.83** ± 2.60	**14.31** ± 0.76	**62.33** ± 1.93	**21.61** ± 1.23	**46.72** ± 2.11	**78.68** ± 1.08	**50.13** ± 1.69
	F(↓)	**19.07** ± 2.17	6.96 ± 1.40	**7.99** ± 4.08	59.05 ± 4.42	**26.97** ± 2.36	**21.79** ± 1.41	**32.65** ± 3.21

dard augmentations work well both in the medical domain and for standard benchmarks of natural images. For other modalities, we propose to perform the augmentation on the latent representation level by simply adding Gaussian noise.

Practicality and computational time - Appendices A.8 and A.9. In terms of practicality, for standard datasets like CIFAR10 and CIFAR100, the number of communication rounds per task with a jump parameter $q = 5$ is comparable to the one reported in FedCIL (40 communication rounds per task) and considerably smaller than MFCL (100 communication rounds per task). The same consideration is also valid for the training computational time; our approach has a comparable computational time compared to FedCIL and a considerably smaller one compared to MFCL.

Performance of generative-based approaches - Appendix A.10. The performance gap of generative-based solutions in our experiments stems from the different assumptions of the online scenario compared to the original setting. For instance, MFCL on CIFAR100 assumes 100 communication rounds for each task. This implies that the whole task dataset is processed 100 times, allowing an effective synthetic image generation. In our online scenario, instead, we have access to one batch of 10 images at every iteration. In this case, the effectiveness of generative-based solutions is limited by their need for large amounts of data. This is in line with the online-CL literature where the most successful solutions rely on memory-based approaches (Ma et al., 2022; Soutif-Cormerais et al., 2023). To illustrate this point, we generate some images using MFCL trained in the online setting (see Figure 5 in Appendix A.10). The synthetic images provide limited information about past tasks resulting in MFCL being better than standard FL approaches without memory, but not competitive compared to our solution (see Table 5).

Limitations. A limitation of the proposed approach is the need to use an ensembling technique (in our case, test-time augmentation) in order to compute the BI-based estimates. This may result in a reduced efficiency compared to, e.g., ER. Furthermore, the TTA-based estimation of the Bregman Information is only an estimate of the true unknown uncertainty; the estimator used throughout the experiments (Eq. 1) is only asymptotically unbiased and underestimates the theoretical quantity (Gruber & Buettner, 2023).

Conclusions. In this work, we address the challenges of real-world FCL, where new data arrive in streams of small chunks that cannot be fully retained after being processed during training. While current research in FCL focuses on generative-based solutions that imply an offline setting where generators are trained at the end of each task, we advocate that in realistic conditions complete task datasets may be unavailable and local models should be updated every time new data are received. For this, we devise and formalise a new scenario for the *online* problem in FCL. To solve CF, we introduce an effective memory-based baseline that combines uncertainty-aware updates with random replay, outperforming state-of-the-art methods. Unlike generative solutions, the newly proposed BI-based estimation for epistemic uncertainty is simple to implement and applicable across different data modalities. Furthermore, compared to standard uncertainty metrics, the empirical results show its superiority in reducing CF in standard settings as well as imbalanced and probing real-world scenarios taken from the biomedical domain. This confirms the ability of the proposed uncertainty-aware strategy to sample more robust and representative samples in challenging tasks and opens new frontiers in FCL, particularly for domains where data scarcity and imbalance are prevalent.

Acknowledgments

This work was supported by the The Federal Ministry for Economic Affairs and Climate Action of Germany (BMWK, Project OpenFLAAS 01MD23001E). Co-funded by the European Union (ERC, TAIPO, 101088594). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

Reproducibility Statement

Code availability. The code used to implement our approach and to conduct the experiments is available at `https://github.com/MLO-lab/online-FCL`.

Data accessibility. All the datasets employed in the experiments are publicly available. We provide detailed instructions on how we use them. In case some particular preprocessing step is required, the corresponding scripts are included in our code repository.

Hyperparameters. All hyperparameters and training details are described in the main paper. The effect of key hyperparameter choices is investigated in the ablation study reported in Appendix A.7.

Hardware and runtime. As mentioned in the main paper, our experiments were conducted on a Linux machine using a single Quadro RTX 5000 and 16 GB RAM. In Tables 14 and 15, we report the expected training runtime for our approach and the mini-batch processing time in seconds respectively.

Experiment instructions. The code repository includes a README file with instructions to run the experiments. Furthermore, to facilitate the comprehension of our approach, Appendix A.11 describes the pseudocode of the proposed methodology.

References

Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert gate: Lifelong learning with a network of experts. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3366–3375, 2017.

Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 139–154, 2018.

Rahaf Aljundi, Eugene Belilovsky, Tinne Tuytelaars, Laurent Charlin, Massimo Caccia, Min Lin, and Lucas Page-Caccia. Online continual learning with maximal interfered retrieval. *Advances in neural information processing systems*, 32, 2019.

Sara Babakniya, Zalan Fabian, Chaoyang He, Mahdi Soltanolkotabi, and Salman Avestimehr. A data-free approach to mitigate catastrophic forgetting in federated class incremental learning for vision tasks. *Advances in Neural Information Processing Systems*, 36, 2024.

Colin Campbell, Nello Cristianini, Alex Smola, et al. Query learning with large margin classifiers. In *ICML*, volume 20, pp. 0, 2000.

Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 532–547, 2018a.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. In *International Conference on Learning Representations*, 2018b.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, P Dokania, P Torr, and M Ranzato. Continual learning with tiny episodic memories. In *Workshop on Multi-Task and Lifelong Reinforcement Learning*, 2019.

Aristotelis Chrysakis and Marie-Francine Moens. Online continual learning from imbalanced data. In *International Conference on Machine Learning*, pp. 1952–1961. PMLR, 2020.

Aron Culotta and Andrew McCallum. Reducing labeling effort for structured prediction tasks. In *AAAI*, volume 5, pp. 746–751, 2005.

Jiahua Dong, Lixu Wang, Zhen Fang, Gan Sun, Shichao Xu, Xiao Wang, and Qi Zhu. Federated class-incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10164–10173, 2022.

Cornelia Gruber, Patrick Oliver Schenk, Malte Schierholz, Frauke Kreuter, and Göran Kauermann. Sources of uncertainty in machine learning–a statisticians' view. *arXiv preprint arXiv:2305.16703*, 2023.

Sebastian Gruber and Florian Buettner. Uncertainty estimates of predictions via a general bias-variance decomposition. In *International Conference on Artificial Intelligence and Statistics*, pp. 11331–11354. PMLR, 2023.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Xu He and Herbert Jaeger. Overcoming catastrophic interference using conceptor-aided backpropagation. In *International Conference on Learning Representations*, 2018.

Sean M Hendryx, Dharma Raj KC, Bradley Walls, and Clayton T Morrison. Federated reconnaissance: Efficient, distributed, class-incremental learning. *arXiv preprint arXiv:2109.00150*, 2021.

Julio Hurtado, Alain Raymond-Sáez, Vladimir Araujo, Vincenzo Lomonaco, Alvaro Soto, and Davide Bacciu. Memory population in continual learning via outlier elimination. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3481–3490, 2023.

Jakob Nikolas Kather, Johannes Krisam, Pornpimol Charoentong, Tom Luedde, Esther Herpel, Cleo-Aron Weis, Timo Gaiser, Alexander Marx, Nektarios A Valous, Dyke Ferber, et al. Predicting survival from colorectal cancer histology slides using deep learning: A retrospective multicenter study. *PLoS medicine*, 16(1):e1002730, 2019.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *online*, 2009.

Lilly Kumari, Shengjie Wang, Tianyi Zhou, and Jeff A Bilmes. Retrospective adversarial replay for continual learning. *Advances in Neural Information Processing Systems*, 35:28530–28544, 2022.

Ken Lang. Newsweeder: Learning to filter netnews. In *Machine learning proceedings 1995*, pp. 331–339. Elsevier, 1995.

Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

Sang-Woo Lee, Jin-Hwa Kim, Jaehyun Jun, Jung-Woo Ha, and Byoung-Tak Zhang. Overcoming catastrophic forgetting by incremental moment matching. *Advances in neural information processing systems*, 30, 2017.

Timothée Lesort, Hugo Caselles-Dupré, Michael Garcia-Ortiz, Andrei Stoian, and David Filliat. Generative models from the perspective of continual learning. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2019.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020.

Vebjorn Ljosa, Katherine L Sokolnicki, and Anne E Carpenter. Annotated high-throughput microscopy image sets for validation. *Nature methods*, 9(7):637–637, 2012.

Yuhang Ma, Zhongle Xie, Jue Wang, Ke Chen, and Lidan Shou. Continual federated learning based on knowledge distillation. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, volume 3, 2022.

Zheda Mai, Ruiwen Li, Jihwan Jeong, David Quispe, Hyunwoo Kim, and Scott Sanner. Online continual learning in image classification: An empirical survey. *Neurocomputing*, 469:28–51, 2022.

Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 7765–7773, 2018.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, 2017.

Cuong V Nguyen, Yingzhen Li, Thang D Bui, and Richard E Turner. Variational continual learning. In *International Conference on Learning Representations*, 2018.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems*, 32, 2019.

Daiqing Qi, Handong Zhao, and Sheng Li. Better generative replay for continual federated learning. In *The Eleventh International Conference on Learning Representations*, 2023.

Amal Rannen, Rahaf Aljundi, Matthew B Blaschko, and Tinne Tuytelaars. Encoder based lifelong learning. In *Proceedings of the IEEE international conference on computer vision*, pp. 1320–1328, 2017.

Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *International conference on machine learning*, pp. 4548–4557. PMLR, 2018.

Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

Donald Shenaj, Marco Toldo, Alberto Rigon, and Pietro Zanuttigh. Asynchronous federated continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5054–5062, 2023.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *Advances in neural information processing systems*, 30, 2017.

Albin Soutif-Cormerais, Antonio Carta, Andrea Cossu, Julio Hurtado, Vincenzo Lomonaco, Joost Van de Weijer, and Hamed Hemati. A comprehensive empirical evaluation on online continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3518–3528, 2023.

Xiaoyu Tao, Xinyuan Chang, Xiaopeng Hong, Xing Wei, and Yihong Gong. Topology-preserving class-incremental learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIX 16*, pp. 254–270. Springer, 2020.

Christian Tomani and Florian Buettner. Towards trustworthy predictions from deep neural networks with fast adversarial calibration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 9886–9896, 2021.

Gido M van de Ven, Tinne Tuytelaars, and Andreas S Tolias. Three types of incremental learning. *Nature Machine Intelligence*, 4(12):1185–1197, 2022.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, 2022.

Lisa Wimmer, Yusuf Sale, Paul Hofman, Bernd Bischl, and Eyke Hüllermeier. Quantifying aleatoric and epistemic uncertainty in machine learning: Are conditional entropy and mutual information appropriate measures? In *Uncertainty in Artificial Intelligence*, pp. 2282–2292. PMLR, 2023.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pp. 38–45, 2020.

Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 374–382, 2019.

Abudukelimu Wuerkaixi, Sen Cui, Jingfeng Zhang, Kunda Yan, Bo Han, Gang Niu, Lei Fang, Changshui Zhang, and Masashi Sugiyama. Accurate forgetting for heterogeneous federated continual learning. In *The Twelfth International Conference on Learning Representations*, 2024.

Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific Data*, 10(1):41, 2023.

Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. In *International Conference on Learning Representations*, 2018.

Jaehong Yoon, Wonyong Jeong, Giwoong Lee, Eunho Yang, and Sung Ju Hwang. Federated continual learning with weighted inter-client transfer. In *International Conference on Machine Learning*, pp. 12073–12086. PMLR, 2021a.

Jaehong Yoon, Divyam Madaan, Eunho Yang, and Sung Ju Hwang. Online coreset selection for rehearsal-based continual learning. In *International Conference on Learning Representations*, 2021b.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

A Appendix

A.1 Predictive Uncertainty Scores

The uncertainty metrics considered in our assessment are the followings:

- *Least Confidence (LC)* (Culotta & McCallum, 2005) measures the predictive uncertainty by looking at the samples with the smallest predicted class probability. If the probability associated with the most probable class $y_{(1)}$ is low, the model is less certain about the given sample.

$$u(x)_{LC} = 1 - \frac{1}{P}\sum_{i=1}^{P} p(y_{(1)} = c|\tilde{x}_i) \tag{2}$$

- *Margin Sampling (MS)* (Campbell et al., 2000) measures the predictive uncertainty by looking at the difference between the most probable predicted class $y_{(1)}$ and the second largest one $y_{(2)}$. If the two probabilities are similar, the model is uncertain.

$$u(x)_{MS} = 1 - \frac{1}{P}\sum_{i=1}^{P} \big(p(y_{(1)} = c|\tilde{x}_i) - p(y_{(2)} = c|\tilde{x}_i)\big) \tag{3}$$

- *Ratio of Confidence (RC)* (Campbell et al., 2000) is similar to MS. In this case, instead of computing the difference, the ratio between the probabilities of the two most probable classes is considered.

$$u(x)_{RC} = \frac{1}{P}\sum_{i=1}^{P} \left(\frac{p(y_{(2)} = c|\tilde{x}_i))}{p(y_{(1)} = c|\tilde{x}_i))}\right) \tag{4}$$

- *Entropy (EN)* (Shannon, 1948) considers, differently from the previously introduced metrics, the whole probability distribution. The entropy is computed as follows:

$$u(x)_{EN} = -\frac{1}{P}\sum_{i=1}^{P} \left(\sum_{j} p(y_j = c|\tilde{x}_i)\log(p(y_j = c|\tilde{x}_i))\right) \tag{5}$$

A.2 Set of Augmentations

As described in the main paper, we employ TTA to measure the epistemic uncertainty via BI and to compute all the other uncertainty estimates. The set of augmentations used in our experiments is presented in Figure 4. Each augmentation is applied singularly on the data points of interest.

```
transform_cands = [
    CutoutAfterToTensor(args, 1, 10),
    CutoutAfterToTensor(args, 1, 20),
    v2.RandomHorizontalFlip(),
    v2.RandomVerticalFlip(),
    v2.RandomRotation(degrees=10),
    v2.RandomRotation(45),
    v2.RandomRotation(90),
    v2.ColorJitter(brightness=0.1),
    v2.RandomPerspective(),
    v2.RandomAffine(degrees=20, translate=(0.1, 0.3), scale=(0.5, 0.75)),
    v2.RandomResizedCrop(args.input_size[1:], scale=(0.8, 1.0), ratio=(0.9, 1.1), antialias=True),
    v2.RandomInvert()
        ]
```

Figure 4: Augmentation set used sequentially for the calculation of uncertainty in the experiments.

A.3 Dataset Statistics

In Table 6, we report the statistics of the datasets used for the main experiments. We use dataset from different domains (CIFAR10, CIFAR100, CRC-Tissue, KC-Cell) and with different modalities (20NewsGroups, Yahoo Answers, DBPedia). For CRC-Tissue, in order to create tasks with decreasing size and equal number of classes, we remove the smallest class from our evaluation. For DBPedia and Yahoo Answers Zhang et al. (2015), we randomly select a subset of 25000 and 33600 samples respectively.

Table 6: Statistics of the datasets.

Number of	Classes	Samples	Tasks
CIFAR10	10	60000	5
CIFAR100	100	60000	5
CRC-Tissue	9	107180	4
KC-Cell	8	236386	4
TinyImageNet	200	100000	10
20NewsGroups	20	18828	5
DBPedia	14	33600	5
Yahoo Answers	10	25000	5

A.4 Additional results on TinyImageNet

Compared to CIFAR datasets, TinyImageNet (Le & Yang, 2015) consists of more classes (200) with larger images (64×64 instead of 32×32), providing a slightly different scenario compared to previously explored datasets. In Table 7, we report the results for all the baselines considered in our experiments (except FedCIL which does not consider images larger than 32×32). Similar to the empirical findings reported in the main paper, BI is able to maintain competitive predictive performance while having the smallest forgetting value.

Table 7: Comparison of average last accuracy (A) and last forgetting (F) on TinyImageNet (10 tasks) for all the considered baselines.

	A(↑)	F(↓)
ER	11.26 ± 1.31	6.31 ± 1.23
CBR	11.06 ± 1.65	6.44 ± 2.08
LC	10.16 ± 0.63	6.96 ± 1.20
MS	10.84 ± 1.08	6.27 ± 1.13
RC	10.88 ± 1.08	6.76 ± 1.17
EN	10.18 ± 1.02	6.69 ± 2.84
FedAvg	1.74 ± 0.30	12.49 ± 1.14
W. FedAvg	1.20 ± 0.29	10.96 ± 0.74
FedProx	1.50 ± 0.25	10.98 ± 0.51
FedCIL	✗	✗
MFCL	3.60 ± 0.49	21.58 ± 0.60
BI (ours)	11.04 ± 0.35	6.07 ± 0.74

A.5 Evaluation Metrics

In line with Yoon et al. (2021a) and Wuerkaixi et al. (2024), we employ the average *Last Accuracy* (A) and average *Last Forgetting* (F) – a federated adaptation of the metrics defined in Chaudhry et al. (2018a). *Last* refers to the measurement of the value at the end of the stream for all the clients. Suppose $a_k^{t,i}$ represents the accuracy of task i after learning task t on client k. The last accuracy A_k

at task T on client k is defined as $A_k = \frac{1}{T} \sum_i a_k^{T,i}$. Let K represent the total number of clients. The average last accuracy is then defined as:

$$A = \frac{1}{K} \sum_{k=1}^{K} A_k \tag{6}$$

The forgetting measures the difference between the peak accuracy and the final accuracy of each task. Usually, the peak accuracy is reached when the considered task is trained. After that, since the model is prone to focus more on the upcoming tasks, we observe a performance degradation on the previous tasks. For this, we want to keep the forgetting as low as possible. The average last forgetting is defined as follows:

$$F = \frac{1}{K} \sum_{k=1}^{K} F_k \tag{7}$$

where F_k is defined as

$$F_k = \frac{1}{T-1} \sum_{t=1}^{T-1} \max_{l \in \{1,\dots,T-1\}} a_k^{l,j} - a_k^{t,j}, \qquad \forall j < t. \tag{8}$$

A.6 Clarification about Figure 3

The notable performance gap on the last task graphically describes what is the effect of catastrophic forgetting (CF) on the learning model. One can see how approaches without replay mechanisms (such as FedAvg, the dashed blue line in the figure) mainly focus on the last task with very good performance ($> 80\%$ accuracy) while completely forgetting the knowledge of the previous ones ($\sim 0\%$ accuracy). This effect can be mitigated with generative-based replay mechanisms (e.g., MFCL). In our case, thanks to the proposed approach, the accuracy across tasks is kept more uniform (between 20% and 50%), leading to better average accuracy and reduced CF.

A.7 Ablation Study

In this section, we investigate the effect of various hyperparameters (i.e., burn-in period, jump parameter, parameter averaging strategy, set of augmentations, and number of tasks) on the learning capabilities of the proposed approach while also providing insights and explanations for best practices to use them in this newly proposed context.

Burn-in period. From the results reported in Table 8, the burn-in period seems to provide some benefit compared to the standard case (burn-in equal to 1) although its contribution is not crucial for improving the overall performance of the approach.

Table 8: Effect of the burn-in period on accuracy (A) and forgetting (F).

Burn-in	A(↑)	F(↓)
1	23.77	29.05
10	25.64	29.64
30	24.89	35.77
50	22.43	34.71
100	24.48	33.20

Parameter averaging strategy. The parameter averaging strategy can be changed without affecting particularly the performance of the approach as can be seen in Table 9. Overall, taking the average between the current and the previous global parameters (as proposed in Shenaj et al. (2023)) helps in reducing CF compared to the vanilla FedAvg.

Table 9: Effect of the parameter averaging strategy on accuracy (A) and forgetting (F).

Strategy	A(↑)	F(↓)
FedAvg	26.73	44.38
Weighted FedAvg	24.89	37.77
FedProx	26.67	38.32

Jump parameter. A value between 3 and 5 is a good trade-off for communication efficiency and good predictive performance. If the communication frequency is troublesome due to hardware constraints, one might use higher values at the price of losing pure predictive performance.

Table 10: Effect of the jump parameter q on accuracy (A) and forgetting (F).

q	A(↑)	F(↓)
1	21.15	16.55
3	37.83	34.84
5	24.89	35.77
10	22.75	46.46
50	21.99	37.86

Set of augmentations. To ablate the influence of the augmentations on the performance of our approach, starting from the set of augmentations listed in Appendix A.2, we randomly remove 4 and 2 augmentations from the list. The results reported in Table 11 suggest that removing augmentations decreases the performance of the proposed approach. This is probably because the higher is the number of augmentations used, the more precise is the uncertainty estimation resulting in better samples stored in the memory.

Table 11: Effect of the augmentation set on accuracy (A) and forgetting (F).

Nr. of augmentations	A(↑)	F(↓)
8	20.84	40.96
10	22.18	38.03
12	24.89	35.77

Number of tasks. For this experiment, we consider CIFAR100 as it allows us to create a larger number of tasks. From the results in Table 12, the increasing number of tasks seems to mainly influence catastrophic forgetting in all considered cases. The results look reasonable since, by increasing the number of tasks, it is more difficult to retain knowledge of the initial tasks. The same trend is also observed for the baselines included for comparison, with BI consistently outperforming ER and MFCL, with large gains especially over ER for large task numbers.

Table 12: Comparison of average last accuracy (A) and last forgetting (F) when increasing the number of tasks on CIFAR100.

Tasks		5	10	20	25
ER	A(↑)	10.95	8.59	9.43	9.09
	F(↓)	9.12	20.06	26.76	41.56
MFCL	A(↑)	4.60	4.43	3.98	3.58
	F(↓)	13.69	18.59	31.60	37.43
BI (ours)	A(↑)	14.31	14.56	14.46	13.71
	F(↓)	8.96	14.36	19.94	33.65

A.8 Practicality Analysis - Number of Communication Rounds

For standard datasets like CIFAR10 and CIFAR100, the number of communication rounds per task with a jump parameter $q = 5$ (as used in the main experiments) is comparable to the one proposed in FedCIL (40 communication rounds per task) and considerably smaller than MFCL (100 communication rounds per task). More in details, for each of the aforementioned datasets, each client possesses roughly 2000 images. With a batch size equal to 10, this results in 200 batches per task. Thus, removing the burn-in period of 30 batches, we are left with 170 batches. This means that, for a given task, the number of communication rounds with $q = 5$ is approximately 35. The frequency can be further reduced by increasing the jump parameter with a slight degradation in the predictive performance of the model (see results in the ablation study reported in Appendix A.7 Table 10.).

Table 13: Communication rounds for FCL approaches on CIFAR10 and CIFAR100.

	Communication rounds
MFCL	100
FedCIL	40
O-FCL (ours)	35

A.9 Time Complexity Analysis

Table 14 reports the runtime in seconds at the end of the training procedure on CIFAR10 for FedAvg (no replay mechanism), ER (random, memory-based), CBR (class-balanced, memory-based), MFCL (generative-based), FedCIL (generative-based), and BI (uncertainty-based, memory-based). From the results, the computational time of BI is in line with most of other uncertainty-based approaches and much shorter than generative approaches. The increase in runtime from CBR to BI stems from the requirement to generate TTA images. In comparison to generative approaches, the runtime of BI is similar or, compared to MFCL, considerably shorter.

Table 14: Runtime (in seconds) at the end of the learning procedure on CIFAR10.

	Runtime (in seconds)
FedAvg	~ 190
ER	~ 210
CBR	~ 300
MFCL	~ 3840
FedCIL	~ 600
BI (ours)	~ 660

In the online scenario, the total number of samples in a dataset does not affect the scalability of an algorithm. In fact, since the method does not use the whole dataset at once but rather process mini-batches sequentially, the only important factor is the time for processing a mini-batch. For this, an important factor that may affect the computational time is the image size. Table 15 reports the effect of the image size on the time for processing a mini-batch.

Table 15: Effect of the image size on the mini-batch processing time in seconds.

Image size	BI (ours)	MFCL
32 × 32	~ 0.20	~ 0.40
64 × 64	~ 0.42	~ 0.75
128 × 128	~ 1.32	✗
224 × 224	~ 4.19	✗

A.10 Performance Analysis of Generative-based Models

The proposed online-FCL (O-FCL) scenario is very different from the standard FCL scenario presented in, e.g., MFCL (Babakniya et al., 2024) and related works based on generative-based solutions. In the standard scenario, they assume to work on task datasets that can be iterated multiple times. For instance, let us consider MFCL on CIFAR100 in which the authors assume to have 100 communication rounds for each task. This implies that the whole task dataset is processed 100 times. Thus, after many iterations over the same task dataset, the generative model can effectively generate synthetic images for the given task. In our online scenario, instead, we only have access to one batch of 10 images at every iteration. In this setting, generative-based solutions are limited in their capabilities as they require many data points and many iterations over the training data for being effective. For this reason, we believe that the performance gap compared to the standard FCL setting is given by the inefficiency of generative models in online settings where we do not have the possibility to retrain the model on the same data multiple times. This is in line with the literature in online-CL where the most successful solutions rely on memory-based solutions (Ma et al., 2022; Soutif-Cormerais et al., 2023). To prove this point, we generated some images using MFCL at the end of the training procedure in the online setting (see Figure 5). Differently from the images reported in the original paper, we can see that no informative pattern has been discovered; the synthetic images provide limited information about the past information. This is also corroborated by the results of MFCL on the datasets for vision tasks reported in the main paper. From the results, we can see that MFCL is better than standard approaches without memory, but it is not competitive compared to memory-based approaches.

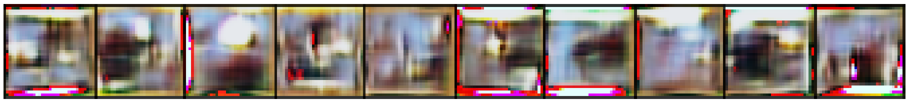

Figure 5: Synthetic images generated by MFCL (Babakniya et al., 2024) in the online setting.

A.11 PSEUDOCODE OF O-FCL

Algorithm 1 O-FCL learning scheme

```
Input K: number of clients, T: number of tasks, $q$: jump parameter, burn-in parameter, $bs$: batch size.
Notation $k$: current client; $\mathcal{M}_k$: client memory, $[\boldsymbol{C}]$: client set, $t$: current task, $b_k^t$ current batch size for client $k$ and task $t$, $\boldsymbol{m}$: memory sample (size equal to $bs$); $bn_k^t$: number of batches for client $k$ and task $t$, $\boldsymbol{\theta}_g$: global model parameters, $\boldsymbol{\theta}_k$: local model parameters.
while training not completed for all clients do

    Client training
    for $k \in [1, \dots, K]$ do                                        ▷ For each client
        $b_k^t, bn_k^t \leftarrow$ GETNEXTBATCH($\boldsymbol{C}_k$)                    ▷ Get one batch from $t$
        if $t = 0$ then
            $\boldsymbol{\theta}_k \leftarrow$ TRAIN($b_k^t$)
        else
            $\boldsymbol{m} \leftarrow$ SAMPLEFROMMEMORY($\mathcal{M}_k$)         ▷ Random sampling from memory
            $\boldsymbol{\theta}_k \leftarrow$ TRAIN($b_k^t$, $\boldsymbol{m}$)
        end if
        $\mathcal{M}_k \leftarrow$ UPDATEMEMORY($\mathcal{M}_k$, $b_k^t$)
    end for

    Parameter Averaging
    if $bn_k^t >$ burn-in and $bn_k^t \ \% \ q = 0$ then
        $\boldsymbol{\theta}_g \leftarrow$ AGGRMODEL($\boldsymbol{\theta}_k \quad \forall k \in [1, \dots, K]$)
        for $k \in [1, \dots, K]$ do
            $\boldsymbol{\theta}_k \leftarrow$ UPDATEPARAMETERS($\boldsymbol{\theta}_g$)
        end for
    end if
end while

function UPDATEMEMORY($\mathcal{M}_k$, $b_k^t$)
    for $c \in b_k^t$ do                                 ▷ For each label $c$ in the batch size
        $\boldsymbol{m}_c \leftarrow \mathcal{M}_k[y = c]$                 ▷ Extract samples with label $c$ from memory
        $\boldsymbol{a}_c \leftarrow \boldsymbol{m}_c \cup b_k^t[y = c]$                   ▷ Get candidate samples for label $c$
        $\boldsymbol{u}_c \leftarrow u_{BI}(\text{TTA}(\boldsymbol{a}_c))$                    ▷ Compute uncertainty with Eq. (1)
        $\mathcal{M}_k \leftarrow$ REPLACESAMPLES($\boldsymbol{u}_c$)                ▷ Update $\mathcal{M}_k$ based on $\boldsymbol{u}_c$
    end for
    return updated $\mathcal{M}_k$
end function
```