Coordinated Replay Sample Selection for Continual Federated Learning

Jack H. Good[1]*, **Jimit Majmudar**[2], **Christophe Dupuy**[2], **Jixuan Wang**[2],
Charith Peris[2], **Clement Chung**[2], **Richard Zemel**[2], **Rahul Gupta**[2]
[1]Carnegie Mellon University
[2]Amazon Alexa AI
jhgood@cs.cmu.edu
{mjimit, dupuychr, wjixuan, perisc, chungcle, rzemel, gupra}@amazon.com

Abstract

Continual Federated Learning (CFL) combines Federated Learning (FL), the decentralized learning of a central model on a number of client devices that may not communicate their data, and Continual Learning (CL), the learning of a model from a continual stream of data without keeping the entire history. In CL, the main challenge is *forgetting* what was learned from past data. While replay-based algorithms that keep a small pool of past training data are effective to reduce forgetting, only simple replay sample selection strategies have been applied to CFL in prior work, and no previous work has explored coordination among clients for better sample selection. To bridge this gap, we adapt a replay sample selection objective based on loss gradient diversity to CFL and propose a new relaxation-based selection of samples to optimize the objective. Next, we propose a practical algorithm to coordinate gradient-based replay sample selection across clients without communicating private data. We benchmark our coordinated and uncoordinated replay sample selection algorithms against random sampling-based baselines with language models trained on a large scale de-identified real-world text dataset. We show that gradient-based sample selection methods both boost performance and reduce forgetting compared to random sampling methods, with our coordination method showing gains early in the low replay size regime (when the budget for storing past data is small).

1 Introduction

The ubiquity of personal devices with a network connection, such as smart phones, watches, and home devices, offer a rich source of data for learning problems such as language modeling or facial recognition. The conventional approach is to collect all the data into one location and use dedicated hardware to learn a model; however, the privacy risk associated with communicating personal data makes this approach unsuitable for many applications. *Federated learning* (FL) offers a solution by learning a central model via distributed training across user-owned devices, without communicating any data to the central server.

In addition, the devices may produce a continual stream of data and, due to storage constraints and/or privacy restrictions, be able to keep only a limited amount of data at a time. Thus *continual federated learning* (CFL) has recently emerged as a prominent topic in machine learning research. CFL incorporates methods from *continual learning* (CL), where a model is periodically fine-tuned on new data. The main challenge for CL is *catastrophic forgetting*, a phenomenon where fine-tuning on new data causes a reduction of performance on past data. This is harmful to long-term generalization, especially when different time periods comprise different tasks, or when the data distribution shifts over time or presents seasonality.

Among various methods, *episodic replay*, wherein a small, fixed-size *replay buffer* of past data is kept and used for fine-tuning along with new data, has proven to be among the most effective strategies to reduce forgetting and improve performance of the final model in both CL (Verwimp et al., 2021) and CFL (Guo et al., 2021; Dupuy et al., 2023). However, only basic replay sample selection strategies, including random sampling and iCaRL (Rebuffi et al., 2017), have been applied to CFL (Guo et al., 2021). To bridge this gap, we adopt the selection objective from gradient-based sample selection (GSS) (Aljundi et al., 2019b), a more recent approach that selects replay samples based on the diversity of their gradients. We propose a new relaxation-based selection method that results in selections closer to optimal compared to methods from prior work.

Any replay sample selection method from CL can be used for CFL by applying it independently

* Work done while the author was an intern at Amazon

at each client. However, CFL presents a yet-unexplored opportunity for the central server to coordinate the selection of replay samples across clients, that is, choose samples such that the union of all clients' replay buffers, rather than each individual buffer, is optimal. The main challenge is that, to ensure privacy, the data cannot be communicated to the server, so selection techniques from CL can not be applied directly. Building on our relaxation-based selection approach, we propose the first server-coordinated replay sample selection approach for CFL. By introducing auxiliary variables that make the objective of the relaxation separable across clients, we enable an alternating minimization (more generally called block coordinate descent) process whereby the optimization alternates between the server and the clients in parallel, all while maintaining communication volume and privacy very similar to standard FL training.

Our novel contributions are 1) a relaxation-based approach to select replay samples that maximize loss gradient diversity; 2) a practical algorithm for coordinated selection of replay samples to maximize gradient diversity jointly across many clients without sacrificing privacy or substantially increasing communication or computation cost; and 3) an empirical analysis of the effect of these strategies on performance and forgetting on a language modeling problem using real-world voice assistant data with heterogeneity across clients and time periods.

2 Related work

FedAvg (McMahan et al., 2017) is a standard FL algorithm wherein the server sends an initial model to a random sample of clients, each client in parallel fine-tunes the model with its local data and sends it back to the server, and the server averages their weights to get a new central model. This is repeated for a number of rounds. If the clients are heterogeneous (have non-i.i.d. data distributions), then the weight averaging results in *client drift*. As a result, convergence rates of algorithms based on FedAvg generally get worse with client heterogeneity (Wang et al., 2019; Karimireddy et al., 2020; Li et al., 2020; Reddi et al., 2020). Several variations of FedAvg have been proposed to address challenges such as client drift (Zhao et al., 2018; Wang et al., 2019; Li et al., 2020; Reddi et al., 2020; Karimireddy et al., 2020). The replay sample selection strategies proposed in this paper are orthogonal to the particulars of the FL algorithm; for our evaluation, we use standard FedAvg.

"Continual learning" can refer to several related problems, but in this work, we consider the problem of learning a single task without forgetting from a continual stream of data, usually by periodic fine-tuning, with some limitations such as hardware capacity precluding the retention of the full history of data. The distribution of data may shift over time. Common approaches to reduce forgetting are to apply regularization penalizing the difference in weights between the current model previous models (Kirkpatrick et al., 2017); keep a small set of historical data and project loss gradients such that they do not increase the loss on these historical data (Lopez-Paz and Ranzato, 2017; Chaudhry et al., 2019; Guo et al., 2020); or keep a small set of historical data to include during training (Rebuffi et al., 2017; Aljundi et al., 2019b,a; Borsos et al., 2020). The last approach, called *episodic replay* or *rehearsal*, has been shown to be especially effective to reduce forgetting in both CL (Verwimp et al., 2021) and CFL (Guo et al., 2021; Dupuy et al., 2023). In particular, gradient-based sample selection (GSS) (Aljundi et al., 2019b) is an episodic replay strategy that chooses replay samples to maximize the diversity of the loss gradients. It is shown to outperform other strategies and is the foundation for our proposed CFL methods.

Continual federated learning (CFL) is a setting where each client receives a continual stream of data and federated learning is periodically applied to update a central model. This setting faces challenges of both heterogeneity across clients, as in FL, and heterogeneity across time steps, as in CL. CFL works that focus on improving performance by reducing forgetting, like this one, include the following: (Yao and Sun, 2020) applies model regularization methods from CL to FL, but focuses on improving generalization of FL by reducing client drift; (Guo et al., 2021) proposes a general CFL framework with convergence analysis and applies CL techniques including model regularization, generative data augmentation, and episodic replay strategies including naive random sampling and iCaRL (Rebuffi et al., 2017), finding that episodic replay outperforms the other CL strategies by a wide margin, with the naive method being superior; (Usmanova et al., 2021) uses a distillation strategy with both central and past local models as teachers for new local models; (Jiang et al., 2021) uses parameter masking to preserve and reuse knowledge; and (Casado et al., 2020) proposes a different take on CFL using lightweight models with ensemble methods, focusing mainly on practical limitations of low-power devices, but also discussing applicability to single-task CL problems with distribution shift. To the best of our knowledge, we are the first to apply gradient-based replay sample selection methods to CFL and the first to propose a server-coordinated approach. Other CFL works focus on FL challenges such as client interference (Yoon et al., 2021) or variable sampling rate, device capabilities, latency, and availability issues (Chen et al., 2020).

3 Problem Formulation

In FL, each client $m \in [M]$ has a set X_m of samples of size $n_m = |X_m|$, and we aim to find a model w that solves the optimization problem

$$\min_w \sum_{m \in [M]} \frac{n_m}{n} \ell(X_m; w) \tag{1}$$

where ℓ indicates a client-level aggregate loss function and $n = \sum_m n_m$ is the total number of samples. In CFL, the samples are further split into T consecutive time periods, so each client $m \in [M]$ and time period $t \in [T]$ has samples $X_{m,t}$ of size $n_{m,t} = |X_{m,t}|$, and we aim to find a model w that minimizes

$$\min_w \sum_{m,t} \frac{n_{m,t}}{n} \ell(X_{m,t}; w) \tag{2}$$

with $n = \sum_{m,t} n_{m,t}$ the total number of samples. Since data is generated sequentially and that user-owned devices typically have limited storage, at time period t each client only has access to the data generated during t and a small subset of the past data. Thus in a CFL setting, we learn a series of models $w_1, \ldots, w_T$, with the goal that w_T minimizes (2); each w_t, for $t \in [T]$, is trained on $X_{1,t}, \ldots, X_{M,t}$ using Federated Learning with initialization from w_{t-1}, except w_1, which is initialized randomly or pre-trained, e.g., on publicly available data.

4 Episodic Replay Strategies

For each t, w_t is trained on $X_{1,t}, \ldots, X_{M,t}$, so we may expect that w_t minimizes $\sum_m \frac{n_{m,t}}{n_t} \ell(X_{m,t}; w_t)$; however, it is not necessarily true that w_t minimizes $\sum_m \frac{n_{m,t'}}{n_{t'}} \ell(X_{m,t'}; w_t)$ for $t' < t$ because training on later data can result in *forgetting*. *Episodic replay* is a simple and effective remedy whereby, at each time period t, each client m has a *replay buffer* $R_{m,t}$ containing at most N_m data from $X_{m,1}, \ldots, X_{m,t-1}$, where N_m is the replay buffer size for client m. Then w_t is trained on $X_{1,t} \cup R_{1,t}, \ldots, X_{M,t} \cup R_{M,t}$ using federated learning. The purpose of the replay buffer is to alleviate forgetting and ultimately result in a good performing model across time periods, and it has been shown in numerous CL (Rebuffi et al., 2017; Aljundi et al., 2019b,a; Borsos et al., 2020; Verwimp et al., 2021) and CFL (Guo et al., 2021; Dupuy et al., 2023) works that episodic replay is effective in accomplishing that. The defining feature of an episodic replay strategy is how $R_{m,t+1}$ is selected from $X_{m,t} \cup R_{m,t}$.

We next describe several such sample selection strategies, which we call *uncoordinated* if the selection is made independently at each client, or *coordinated* if the selection is made jointly across clients.

4.1 Random sample selection

The most basic approach to replay sample selection is random sampling, which is always uncoordinated. We consider three baseline methods based on random sampling: naive uniform, approximation of uniform, and fixed proportion proposed in (Dupuy et al., 2023), that we also describe in Appendix A.

4.2 Uncoordinated gradient-based selection

Replay sample selection from CL can be adapted for uncoordinated sample selection in CFL by applying them independently at each client. Thus, to simplify notation for uncoordinated strategies, we can omit the client index m. We adopt the strategy of (Aljundi et al., 2019b) to select data into the replay buffer with high diversity of loss gradients, that is, the gradient of the loss function with respect to model parameters, as used to train the model. At period t, we compute the loss gradients after training model w_t on $X_t \cup R_t$. For a given client at the end of period, let $g_i \in \mathbb{R}^d$ be the loss gradient for sample $i \in [n'_t]$ for model w_t, with d the number of model parameters, and let $n'_t = |X_t \cup R_t|$ be the size of the data and replay buffer at time t. As per (Aljundi et al., 2019b), we select the replay buffer R_{t+1} to minimize the cosine similarity of gradients for selected samples.

$$\begin{aligned} \min_{R} \quad & \sum_{i,j \in R} \frac{\langle g_i, g_j \rangle}{\|g_i\| \|g_j\|} \\ \text{s.t.} \quad & R \subseteq X_t \cup R_t \\ & |R| = N \end{aligned} \tag{3}$$

This is generally NP-Hard to solve exactly (Aljundi et al., 2019b). As a result, (Aljundi et al., 2019b) proposes two methods to find approximate solution, one using a greedy heuristic and the other using online clustering, both of which are designed for efficiency in an online learning setting. We propose a different approximation: introduce variables x_i, $i \in [n'_t]$ and equivalently write Problem (3) as

$$\begin{aligned} \min_{x} \quad & x^T G^T G x \\ \text{s.t.} \quad & x_i \in \{0, 1\} \text{ for all } i \\ & \sum_i x_i = N \end{aligned} \tag{4}$$

where $G \in \mathbb{R}^{d \times n'_t}$ is the matrix of gradient directions defined by $G_{:,i} = g_i / \|g_i\|$, and let $R_{t+1} = \{i \mid x^*_i = 1\}$ for solution x^* to Problem (4). We relax the domain of x_i from $\{0, 1\}$ to $[0, 1]$. The resulting problem is convex quadratic minimization and efficient to solve; we finally let R_{t+1} be the set of data with the top-N values in the solution x^*.

Because the diagonal of $G^T G$ is 1, and because with high-dimensional gradients the off-diagonal elements of G tend to be near 0, x^* tends to have values mostly close to the average N/n'_t, so the solution resulting from the top-N operation may be poor. To alleviate this, we set the diagonal of $G^T G$ to zero, which is equivalent to removing the $i = j$ terms in the sum of Problem (3), which always sum to N, so this does not change the minimizer. In the relaxation, however, it tends to result in x^* values that are mostly 0 and 1, reducing error from the top-N selection, but causing the relaxation to possibly be non-convex. We find that both versions of our relaxation result in better solutions in practice than the heuristics from (Aljundi et al., 2019b) (see Figure 1 in Section 5), with the non-convex outperforming the convex relaxation. Therefore we use the non-convex relaxation of Problem (4) for uncoordinated gradient-based replay sample selection. This relaxation-based formulation also makes possible the coordinated selection strategy proposed in the next section.

Due to the high-dimension of the gradients, it is best in practice to compute $G^T G$ first and solve the relaxation of Problem (4) as written; however, the relaxed problem can also be expressed more intuitively as

$$\begin{aligned} \min_{x} \quad & \|Gx\|^2 \\ \text{s.t.} \quad & x_i \in [0, 1] \text{ for all } i \\ & \sum_i x_i = N \end{aligned} \tag{5}$$

and interpreted as choosing the data with the minimal-magnitude sum of gradient directions for selected data. This will help motivate the coordinated formulation proposed in the next section.

4.3 Coordinated sample selection

A coordinated sample selection strategy aims for the union of all clients' replay buffers, rather than each clients' individual buffer, to be optimal. For example, in uncoordinated selection, many clients may choose similar samples for replay, which results in suboptimal representation for training, but coordinated selection aims for diversity across clients. This means clients cannot independently make selections, and because client data (hence gradients) may not be communicated to the server, replay sample selection methods for CL cannot necessarily be adapted directly into coordinated CFL methods.

To make the gradient diversity objective of (5) coordinated, we sum over data in the union of all clients' selected replay samples instead of an individual client's.

$$\begin{aligned} \min_{x_1, \ldots, x_M} \quad & \left\| \sum_m G_m x_m \right\|^2 \\ \text{s.t.} \quad & x_{m,i} \in [0, 1] \text{ for all } m, i \\ & \sum_i x_{m,i} = N_m \text{ for all } m \end{aligned} \tag{6}$$

The obvious approach is to have each client m send G_m to the server and solve Problem (6) there; however, not only can Problem (6) be resource-intensive to solve centrally with many clients, but this also introduces a very large communication cost, as each column of G_m is the size of the model itself. More importantly, communicating gradients puts client data at risk since individual gradients are vulnerable to privacy attack (Zhu et al., 2019). Therefore, the goal is to solve Problem (6) without substantial increase in communication or computation cost, and without communicating data, gradients, or anything else that reduces privacy.

We propose an alternating minimization process whereby an objective is minimized alternatively at the server and in parallel at the clients. Define auxiliary variables $h_1, \ldots, h_M$ such that $h_m := G_m x_m - \frac{1}{M}\sum_{n\in[M]} G_n x_n$. Then we have

$$\left\| \sum_{n\in[M]} G_n x_n \right\|^2 = M^2 \left\| G_m x_m - h_m \right\|^2$$

for each $m \in [M]$. Adding over all $m \in [M]$, Problem (6) can be equivalently written as

$$\begin{aligned}
\min_{\substack{x_1,\ldots,x_M\\ h_1,\ldots,h_M}} \quad & M \sum_{m\in[M]} \left\| G_m x_m - h_m \right\|^2 \\
\text{s.t.} \quad & x_{m,i} \in [0,1] \text{ for all } m, i \\
& \sum_i x_{m,i} = N_m \text{ for all } m \\
& h_m = G_m x_m - \frac{1}{M}\sum_n G_n x_n .
\end{aligned} \tag{7}$$

Next, relax Problem (7) to

$$\begin{aligned}
\min_{\substack{x_1,\ldots,x_M\\ h_1,\ldots,h_M}} \quad & M \sum_{m} \left\| G_m x_m - h_m \right\|^2 \\
\text{s.t.} \quad & x_{m,i} \in [0,1] \text{ for all } m, i \\
& \sum_i x_{m,i} = N_m \text{ for all } m \\
& \sum_m h_m = 0.
\end{aligned} \tag{8}$$

Problem (8) is a relaxation of Problem (7) because the feasible set of the latter is a subset of the former. Theorem 1, proven in Appendix B, shows that this relaxation is tight.

Theorem 1. *If $x_1^*, \ldots, x_M^*, h_1^*, \ldots, h_M^*$ is an optimal solution of (8), then it is also an optimal solution of (7).*

As a consequence of Theorem 1, we can determine an optimal solution of the original coordinated problem (6) by solving (8). Moreover, if we fix h and consider minimization only over x, then Problem (8) is separable over the M clients. This means we can use an alternating minimization (more generally called block coordinate descent (Wright, 2015)) algorithm where each client m optimizes w.r.t. x_m given h_m in parallel and sends the resulting $G_m x_m^*$ to the server, then the server optimizes w.r.t. $h_1, \ldots, h_M$ given $G_1 x_1, \cdots, G_M x_M$ and sends each resulting h_m^* to client m. We initialize with $h_m = 0$ for all m so that the selection at zero iterations is the same as uncoordinated. Pseudocode is given in Algorithm 1. It is shown by (Luo and Tseng, 1993) that block coordinate descent of a quadratic function over a convex polyhedron converges at least linearly to a stationary point, and in our case, that function is convex, so this alternating process improves at every iteration and converges at least linearly to an optimum of the coordinated objective on the relaxed domain.

Despite this, neither the data itself nor individual gradients need to be communicated. What is communicated is targets h_m and weighted sum loss gradients $G_m x_m$. Each is just one gradient-sized vector rather than one per local data point as in sending the gradients themselves. Thus the communication cost per iteration is the same as FedAVG. The number of iterations can be chosen up-front as a hyperparameter to trade off optimality of the selection with number of rounds and total volume of communication, or there could be a stopping condition such as a threshold on change in loss indicating convergence. As for privacy, FedAVG itself already makes a weighted combination of gradients public when run with one batch per client; it is simply the difference between the model parameters sent to the client and the parameters the client sends back to the server. In this sense, this algorithm is no less private than general FedAVG.

Algorithm 1 Coordinated replay sample selection.

```
at each client m:
    G_m ← gradients at X_t ∪ R_t
    h_m ← 0
repeat
    at each client m:
        x_m ← min_x   ||G_m x − h_m||^2
                 s.t.  x_i ∈ [0,1] for all i
                       Σ_i x_i = N_m
        send G_m^T x_m to the server
    at the server:
        h_m ← G_m x_m − (1/M) Σ_{n=1}^{M} G_n x_n
        send each h_m to client m
until convergence or max iterations
at each client m:
    select R_{t+1} from X_t ∪ R_t by top-N_m of x_m
```

To efficiently solve the minimization at clients when gradients are large, write $\|G_m x - h_m\|^2 = x^T G_m^T G_m x + h_m^T h_m - 2h_m^T G_m x$ and pre-compute $G_m^T G_m$ and $h_m^T G_m$. Also, G_m is the same at each iteration of the alternating minimization, so $G_m^T G_m$

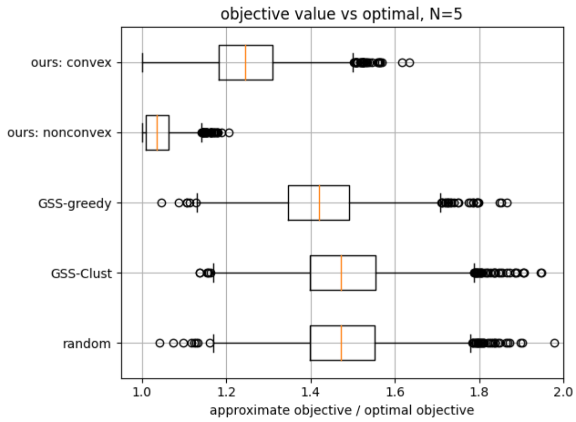


Figure 1: Distribution of objective values vs. optimal for approximate sample selection strategies.

may be computed just once.

4.3.1 Intuitive interpretation

This alternating minimization process has an intuitive interpretation. The goal is to choose replay data such that their sum of loss gradient directions across clients is close to zero. The server sends a "target sum gradient" h_m to each client m, which is initially zero. Each client independently chooses data so that its sum gradient $G_m x_m$ is as close as possible to its target h_m, then sends the result $G_m x_m$ back to the server. The server adjusts the targets h_m to be as close as possible to the sum gradients actually returned by the clients, while maintaining that $\sum_m h_m = 0$. In this sense, the back-and-forth process searches for the sum gradient assignments h_m that sum to zero, and therefore targets the coordinated gradient diversity objective, while being the most individually achievable by clients given their respective data.

5 Experiments

We run experiments to demonstrate the quality of our relaxation-based sample selection and the performance of models trained using CFL with the proposed sample selection strategies. Additional experimental details and results are in Appendix C.

5.1 Near-optimality of relaxation-based selection

We empirically compare our relaxation-based sample selection to the heuristic selection strategies proposed by (Aljundi et al., 2019b), as well as a random selection baseline. We use randomly drawn vectors $g_i \in \mathbb{R}^{300}$ and select $N = 5$ out of $n = 50$ data. We repeat the selection process 5000 times. For each approach, we assess the quality of the selection by comparing the resulting objective value as in Problem (3) to the optimal value obtained by brute-force search (which is possible because N and n are small).

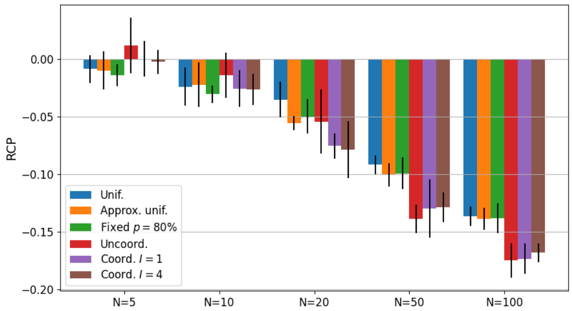
The distribution of objective ratios for each method is shown in Figure 1. Our relaxations achieve the best objective values, with the non-convex relaxation being superior; we expect this is because, with the convex relaxation, many x^* values are close to the mean, resulting in error during the top-N operation that is not present with the non-convex relaxation, where x^* values are close to 0 and 1. In terms of objective value, the heuristic selection strategies from (Aljundi et al., 2019b) are only slightly better than random.

5.2 Comparison of sample selection methods

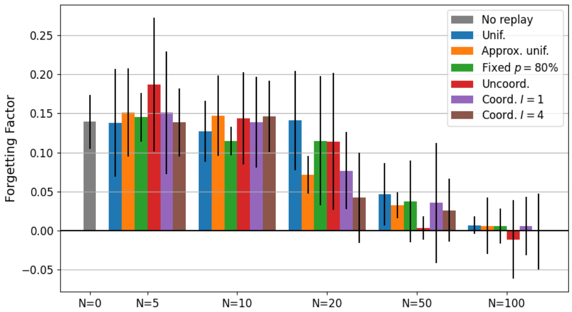
We compare CFL models learned using various replay buffer sizes and sample selection strategies, including the proposed coordinated and uncoordinated strategies as well as baseline strategies using random sampling. We train a model with the TinyBERT architecture to a masked language modeling (MLM) task, where the performance metric is perplexity (lower is better). We choose TinyBERT (Jiao et al., 2020) because distilled models with smaller footprints are more suitable for FL applications. We use 5 data sets, each of which comprises of automated transcriptions of utterances from a random sample of 1000 voice assistant users split into 10 time periods of 5 weeks each: the first 4 weeks are used for training and the remaining 1 week for testing. Additional experiment details in Appendix C.2.

All results are given in terms of relative change in perplexity (RCP), that is, the relative change in perplexity for the experimental model with respect to the model trained without episodic replay ($N = 0$). We also report the forgetting factor, defined as the difference in performance between the latest model and the best performance of the previous models on the same test set (Dupuy et al., 2023). A zero or negative value means that the latest model does not present forgetting on this test set; a positive value means that a past model performs better than the latest model on this test set, which indicates forgetting.

Figure 2 shows the overall performance and forgetting factor for each replay buffer size and sample selection strategy. As expected, we see that the error and forgetting both decrease as the re-



(a) All-period test set perplexity.



(b) All-period test set forgetting.

Figure 2: Relative change in perplexity (RCP) of models learned with various replay sample selection strategies. Error bars show standard deviation over 5 different samples of clients.

play buffer size N increases; at $N = 100$, there is close to no forgetting on average. We also see that gradient-based sample selection increasingly outperforms random sample selection as N increases. Coordinated sample selection appears to outperform uncoordinated sample selection with a low replay budget, $N \leq 20$. There does not seem to be a notable difference between 1 and 4 iterations of coordinated optimization, suggesting that most of the benefit from coordinated selection is achieved after just one iteration.

Figure 3 shows the $N = 20$ RCP results for each

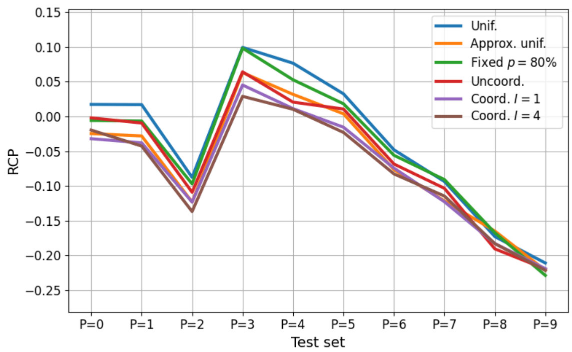


Figure 3: Performance on each period for $N = 20$.

period of the test set, relative to the all-period test perplexity for the no-replay model. As expected, with some exception, performance is generally better on more recent periods. Also, the performance gap between methods is larger on earlier time periods, with the coordinated methods consistently performing best on each time period except the most recent ones. Results for other N are shown in Appendix C.2.

6 Discussion

We proposed a new relaxation for gradient-based selection of replay samples in continual learning. Based on this, we proposed the first algorithm for coordinated replay sample selection in continual federated learning, which converges to the optimal selection under our relaxation while maintaining privacy and low communication cost. Our experiments show that, compared to random sampling, the gradient-based selection of replay samples improves performance of the final model for various replay buffer sizes, and coordinated selection improves for small buffer sizes.

7 Limitations

The reproducibility of this work is limited because the data used for some experiments is not public. Moreover, training language models in a large CFL setting is extremely demanding of both time and computational resources.

Acknowledgements

We thank Saleh Soltan for creating the BERT embeddings and encoder that were used in this work.

References

Rahaf Aljundi, Lucas Caccia, Eugene Belilovsky, Massimo Caccia, Min Lin, Laurent Charlin, and Tinne Tuytelaars. 2019a. *Online Continual Learning with Maximally Interfered Retrieval*. Curran Associates Inc., Red Hook, NY, USA.

Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. 2019b. Gradient based sample selection for online continual learning. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Zalán Borsos, Mojmir Mutny, and Andreas Krause. 2020. Coresets via bilevel optimization for continual learning and streaming. In *Advances in Neural Information Processing Systems*, volume 33, pages 14879–14890. Curran Associates, Inc.

Fernando E. Casado, Dylan Lema, Roberto Iglesias, Carlos V. Regueiro, and Senén Barro. 2020. Federated and continual learning for classification tasks in a society of devices.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. 2019. Efficient lifelong learning with a-GEM. In *International Conference on Learning Representations*.

Y. Chen, Y. Ning, M. Slawski, and H. Rangwala. 2020. Asynchronous online federated learning for edge devices with non-iid data. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 15–24, Los Alamitos, CA, USA. IEEE Computer Society.

Christophe Dupuy, Jimit Majmudar, Jixuan Wang, Tanya Roosta, Rahul Gupta, Clement Chung, Jie Ding, and Salman Avestimehr. 2023. Quantifying catastrophic forgetting in continual federated learning. In *ICASSP 2023*.

Yongxin Guo, Tao Lin, and Xiaoying Tang. 2021. Towards federated learning on time-evolving heterogeneous data.

Yunhui Guo, Mingrui Liu, Tianbao Yang, and Tajana Rosing. 2020. Improved schemes for episodic memory-based lifelong learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 1023–1035. Curran Associates, Inc.

Ziyue Jiang, Yi Ren, Ming Lei, and Zhou Zhao. 2021. Fedspeech: Federated text-to-speech with continual learning. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 3829–3835. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. TinyBERT: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174, Online. Association for Computational Linguistics.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. 2020. SCAFFOLD: Stochastic controlled averaging for federated learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5132–5143. PMLR.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526.

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

David Lopez-Paz and Marc' Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Zhi-Quan Luo and Paul Tseng. 1993. Error bounds and convergence analysis of feasible descent methods: a general approach. *Annals of Operations Research*, 46(1):157–178.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282. PMLR.

S. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert. 2017. icarl: Incremental classifier and representation learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5533–5542, Los Alamitos, CA, USA. IEEE Computer Society.

Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. 2020. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*.

Anastasiia Usmanova, François Portet, Philippe Lalanda, and German Vega. 2021. A distillation-based approach integrating continual learning and federated learning for pervasive services.

Eli Verwimp, Matthias De Lange, and Tinne Tuytelaars. 2021. Rehearsal revealed: The limits and merits of revisiting samples in continual learning. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9365–9374.

Shiqiang Wang, Tiffany Tuor, Theodoros Salonidis, Kin K. Leung, Christian Makaya, Ting He, and Kevin Chan. 2019. Adaptive federated learning in resource constrained edge computing systems. *IEEE Journal on Selected Areas in Communications*, 37(6):1205–1221.

Stephen J. Wright. 2015. Coordinate descent algorithms. *Mathematical Programming*, 151(1):3–34.

Xin Yao and Lifeng Sun. 2020. Continual local training for better initialization of federated models. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 1736–1740.

Jaehong Yoon, Wonyong Jeong, Giwoong Lee, Eunho Yang, and Sung Ju Hwang. 2021. Federated continual learning with weighted inter-client transfer. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12073–12086. PMLR.

Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. 2018. Federated learning with non-iid data.

Ligeng Zhu, Zhijian Liu, and Song Han. 2019. Deep leakage from gradients. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Appendix

A Random sampling strategies

Here we describe the replay sample selection strategies based on random sampling, which were omitted from the main text to comply with page limits.

Naive uniform: each client samples N data uniformly at random from $X_t \cup R_t$. This method is "naive" because the likelihood of selecting examples from the earliest periods decreases with time, which suggests higher vulnerability to catastrophic forgetting.

Approximation of uniform: each client samples $Nn_t/n_{\leq t}$ data uniformly from X_t and $Nn_{<t}/n_{\leq t}$ data uniformly from R_t. In this way, R_{t+1} approximates a uniform sample from $X_{\leq t}$, the set of all data seen so far. While this allows early time periods to continue to be represented, the representation of each individual period reduces over time; after many time steps, the number of samples from even the most recent time period approaches 0.

Fixed proportion $p \in (0,1)$: each client samples pN data uniformly from X_t and $(1-p)N$ data uniformly from R_t. Like naive uniform, the buffer contains fewer data from earlier periods, but the decrease is controlled by the chosen p instead of customer activity.

B Proof of Theorem 1

Theorem 1. *If $x_1^*, \dots, x_M^*, h_1^*, \dots, h_M^*$ is an optimal solution of (8), then it is also an optimal solution of (7).*

Proof. We first show that $x_1^*, \dots, x_M^*, h_1^*, \dots, h_M^*$ is a feasible solution of (7). Since (8) is convex, using the KKT optimality conditions, $x_1^*, \dots, x_M^*, h_1^*, \dots, h_M^*$ is optimal for (8) if and only if it is feasible for (8) and there exist non-negative vectors $u_1^*, \dots, u_M^*, v_1^*, \dots, v_M^*$, vector w^*, and scalars $\alpha_1^*, \dots, \alpha_M^*$ satisfying

- $x_{m,i}^* u_{m,i}^* = 0$ and $(1 - x_{m,i}^*) u_{m,i}^* = 0$ for all $m \in [M], i \in [N_m]$,
- $2MG_m^T(G_m x_m^* - h_m^*) - u_m^* + v_m^* + \alpha_m^* e = 0$ for all $m \in [M]$,
- $-2M(G_m x_m^* - h_m^*) + w^* = 0$ for all $m \in [M]$.

Adding the last equation over all $m \in [M]$, we get $w^* = 2\sum_m (G_m x_m^* - h_m^*)$ and therefore, for all $m \in [M]$,

$$\begin{aligned} h_m^* &= G_m x_m^* - \frac{w^*}{2M} \\ &= G_m x_m^* - \frac{1}{M}\sum_m (G_m x_m^* - h_m^*) \\ &= G_m x_m^* - \frac{1}{M}\sum_m G_m x_m^*. \\ &\left(\because \sum_{m\in[M]} h_m^* = 0\right) \end{aligned}$$

This shows that $x_1^*, \dots, x_M^*, h_1^*, \dots, h_M^*$ is a feasible solution of (7).

Next we show the optimality of $x_1^*, \dots, x_M^*, h_1^*, \dots, h_M^*$ for (7). Suppose for contradiction that $x_1', \dots, x_M', h_1', \dots, h_M'$ is a feasible solution of (7) such that

$$\sum_m \|G_m x_m' - h_m'\|^2 < \sum_m \|G_m x_m^* - h_m^*\|^2.$$

This contradicts the optimality of $x_1^*, \dots, x_M^*, h_1^*, \dots, h_M^*$ for (8) since $x_1', \dots, x_M', h_1', \dots, h_M'$ is a feasible solution of (7). □

C Experiment Details and Additional Results

This section contains additional details and results for the experiments.

C.1 Near-optimality of relaxation-based selection

For these experiments, the vectors $g_i \in \mathbb{R}^d$ with $d = 300$, $i \in [n]$ were generated for each of $M = 1000$ clients by sampling from a random Gaussian mixture as follows. Let the number of centers be

$n_c = p + 1$ with $p \sim \text{Poisson}(4)$, then sample centers $c_{k,j} \sim \mathcal{N}(0, 1)$ for $k \in [n_c]$, $j \in [300]$ and normalize such that each $\{c_{k,j} \mid k \in [n_c]\}$ has mean 0 and standard deviation 1. Let $w \sim \text{Dir}(\mathbf{1}_{n_c})$, where $\mathbf{1}_{n_c}$ is the vector of length n_c whose elements are 1, then for each $i \in [n]$, sample $k \sim \text{Categorical}(w)$ and $g_{i,j} \sim \mathcal{N}(c_{k,j}, 1)$. Finally, normalize the g_i so that $\{g_{i,j} \mid i \in [n]\}$ has mean 0 and standard deviation 1.

The results for $n = 50$ were shown in the main text; we show results for additional n in Figure 4. We see that the relative gap between the optimal and approximate selection increases with n for all methods; however, the relative difference between the approximate methods is similar regardless of n.

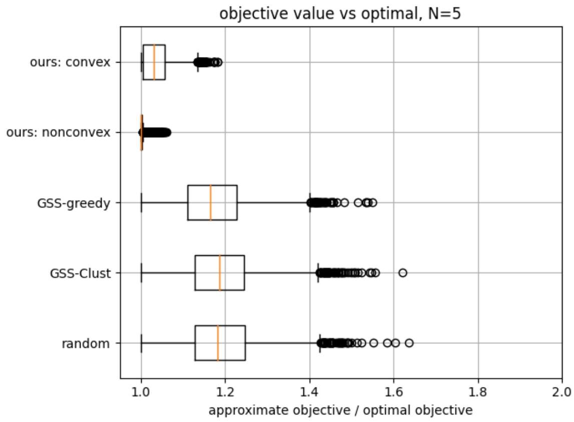
C.2 Comparison of sample selection methods

We use a TinyBERT model (Jiao et al., 2020) for our experiments, with L=4, H=312, A=12 and feed-forward/filter size=1200 where we denote the number of layers (i.e., Transformer blocks) as L, the hidden size as H, and the number of self-attention heads as A.

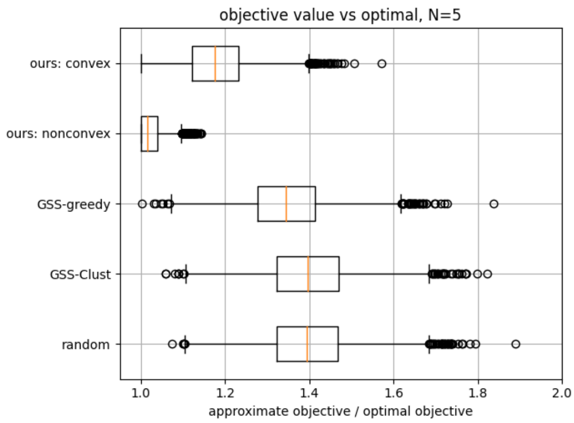
We ran 1555 parallelized experiments using p3.16x instances. Our training time per period per instance was approximately 21 minutes. Note that there was wide variance in training time values given that experiments for earlier periods take less time than experiments for later periods because of the replay buffer increasing training data size.

Figure 5 shows the overall results in the form of a scatterplot. This contains the same information as Figure 2, but visualized differently. There is a clear strong correlation between forgetting factor and performance; this support the idea that the models with better replay improve performance by reducing forgetting.

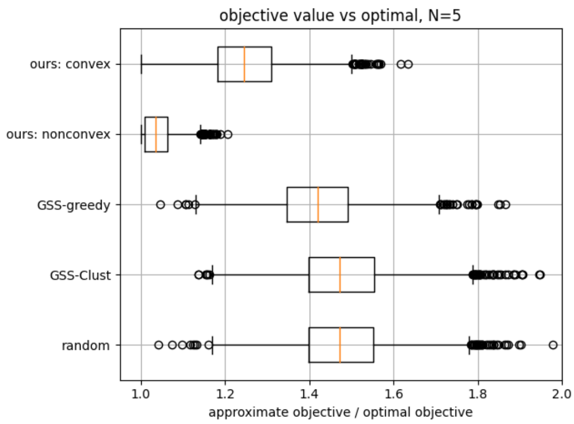
Figure 6 shows the performance and forgetting broken down by the period of the test set, as in Figure 3, but for both evaluation metrics and for all N.



(a) $n = 10$



(b) $n = 30$



(c) $n = 50$

Figure 4: Distribution of objective values vs. optimal for approximate sample selection strategies.

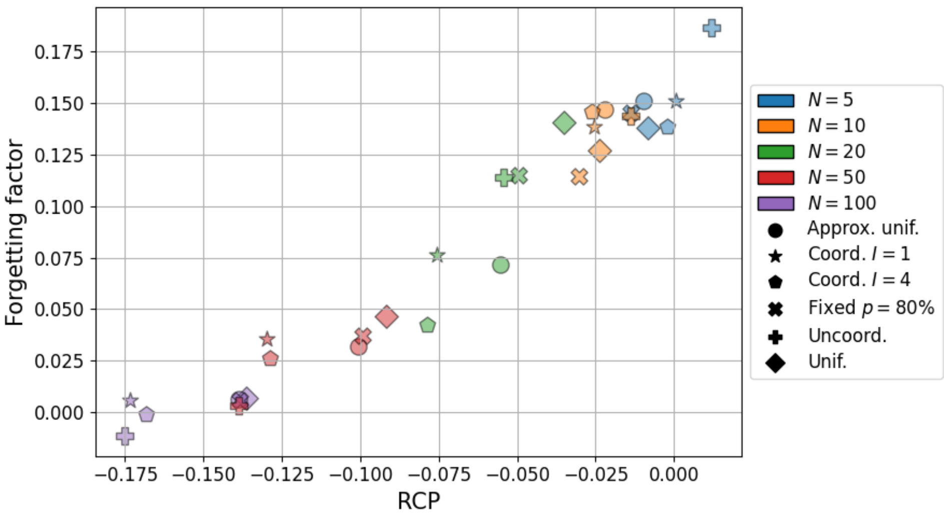
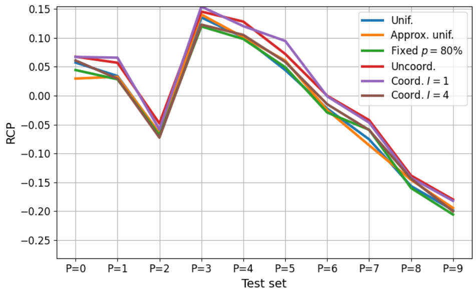
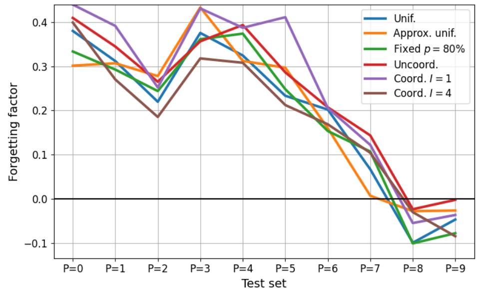


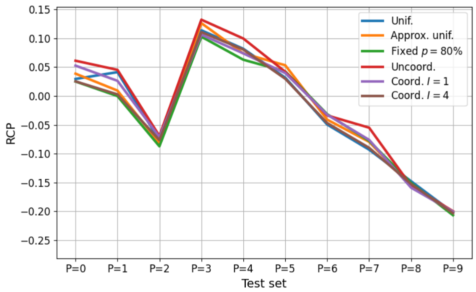
Figure 5: Performance and forgetting results displayed as a scatterplot.



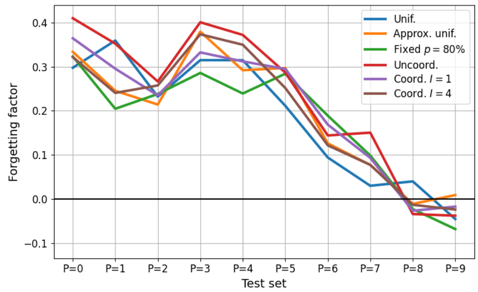
(a) $N = 5$ performance.



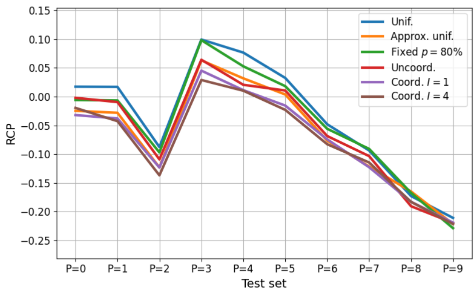
(b) $N = 5$ forgetting.



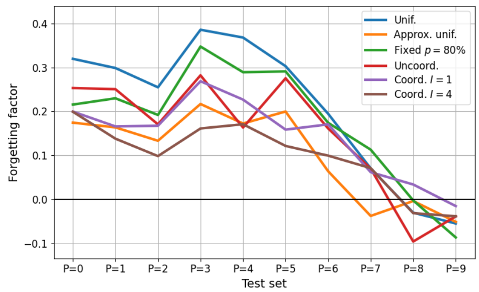
(c) $N = 10$ performance.



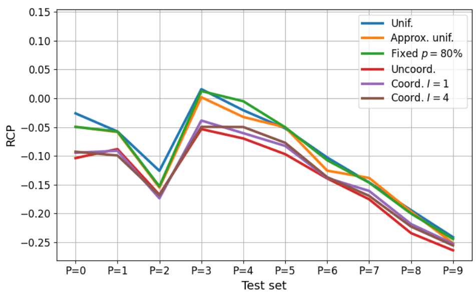
(d) $N = 10$ forgetting.



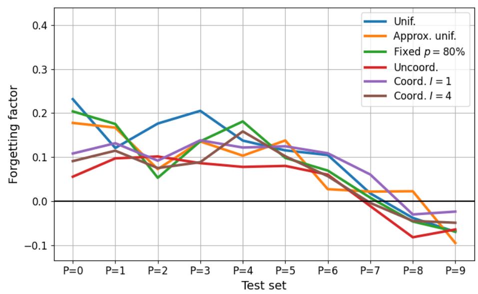
(e) $N = 20$ performance.



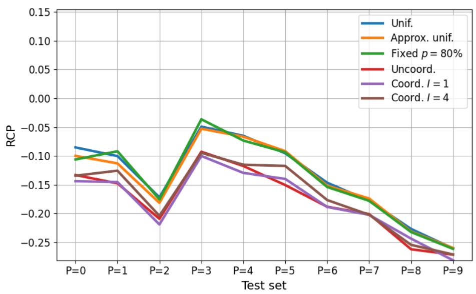
(f) $N = 20$ forgetting.



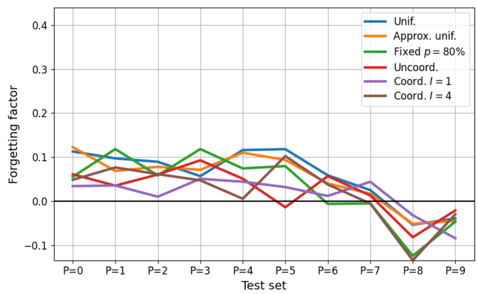
(g) $N = 50$ performance.



(h) $N = 50$ forgetting.



(i) $N = 100$ performance.



(j) $N = 100$ forgetting.

Figure 6: Period test results for all N.