ILASR: Privacy-Preserving Incremental Learning for Automatic Speech Recognition at Production Scale

Gopinath Chennupati
Milind Rao
Gurpreet Chadha
Aaron Eakin
Amazon Alexa
USA

Anirudh Raju
Gautam Tiwari
Anit Kumar Sahu
Ariya Rastrow
Jasha Droppo
Amazon Alexa
USA

Andy Oberlin
Buddha Nandanoor
Prahalad Venkataramanan
Zheng Wu
Pankaj Sitpure
Amazon Alexa
USA

ABSTRACT
Incremental learning is one paradigm to enable model building and updating at scale with streaming data. For end-to-end automatic speech recognition (ASR) tasks, the absence of human annotated labels along with the need for privacy preserving policies for model building makes it a daunting challenge. Motivated by these challenges, in this paper we use a cloud based framework for production systems to demonstrate insights from privacy preserving incremental learning for automatic speech recognition (ILASR). By privacy preserving, we mean, usage of ephemeral data which are not human annotated. This system is a step forward for production level ASR models for incremental/continual learning that offers near real-time test-bed for experimentation in the cloud for end-to-end ASR, while adhering to privacy-preserving policies. We show that the proposed system can improve the production models significantly (3%) over a new time period of six months even in the absence of human annotated labels with varying levels of weak supervision and large batch sizes in incremental learning. This improvement is 20% over test sets with new words and phrases in the new time period. We demonstrate the effectiveness of model building in a privacy-preserving incremental fashion for ASR while further exploring the utility of having an effective teacher model and use of large batch sizes.

CCS CONCEPTS
• Computing methodologies → Speech recognition; Neural networks; Semi-supervised learning settings; • Security and privacy → Privacy-preserving protocols.

KEYWORDS
Incremental Learning, Automatic Speech Recognition, Privacy-preserving Machine Learning

1 INTRODUCTION
Privacy preserving machine learning [1] has been at forefront, due to both increased interest in privacy and the potential susceptibility of deep neural networks to leaks and attacks. Federated Learning (FL) [44] is a machine learning technique that involves training models on edge devices, where data need not leave the device, and can be heterogeneous and non-identically and independently distributed (non-IID). In FL, multiple model updates from a number of participating devices are aggregated. In spite of raw data not leaving the edge device, FL has found to be susceptible to gradient inversion attacks [65, 66]. In response, various privacy-preserving mechanisms such as differential privacy and secure aggregation [19, 58] have been proposed to counter data leakage and conform to privacy preserving mechanisms. Moreover, the lack of labels for the data present in the participating entities, makes FL more challenging for applications such as automatic speech recognition (ASR). Most research in FL until now focuses on training models from scratch. In this work, we focus on privacy-preserving incremental learning (IL), in the context of end-to-end production model building at scale over extended time periods. Incremental learning [8, 62] has been extensively used to incrementally update models on the fly instead of training them from scratch. Incremental learning as such is not privacy-preserving.

Despite the above advances, to the best of our knowledge, few frameworks exist for privacy-preserving incremental training of end-to-end automatic speech recognition models. Prior work on federated learning for speech-based tasks [13, 16, 23] and end-to-end ASR [18, 26], focus on standard benchmarks and not on large scale production data. Privacy-preserving IL on device for end-to-end ASR poses a number of challenges. Production-sized end-to-end ASR systems [11, 24] are expensive to train even in traditional environments.

1 e.g. LibriSpeech [47] is a small sized dataset (~ 1000 hours) recorded in a controlled environment.
We evaluate ILASR on three student recurrent neural network transducer (RNN-T) [24] architectures. The semi-supervised learning (SSL) approach produces machine transcripts using a larger teacher model. The students are pre-trained on in-house de-identified (SSL) approach produces machine transcripts using a larger teacher model. The students are pre-trained on in-house de-identified data until 2020. Through training in ILASR, we observe an improvement of \(3 - 7\%\) in word error rate (WER) over the pre-trained baselines when these students are trained incrementally on a new time period of six months in 2021. The improvement in WER is termed relative word error rate reduction (WERR). This increases to \(20\%\) on test sets with new words and phrases in 2021. Similarly, when the student models are trained incrementally each month, we observe WER improvements, as well the phenomenon where models get stale without further updates.

The paper is organized as follows: section 2 describes the essential concepts used in the paper; section 3 explains the proposed system; section 4 describes the experimental settings; section 5 presents the results; section 6 summarizes the related literature and finally, section 7 concludes and recommends future directions.

2 BACKGROUND

In this section we summarize the RNN-T architecture and large batch training with stochastic gradient descent (SGD).

2.1 RNN-T model architecture

Figure 1 shows the RNN-T [24] architecture used in real-time speech recognition. The model predicts the probability \(P(y|x)\) of labels \(y = (y_1, ..., y_T)\) given acoustic features \(x = (x_1, ..., x_T)\). It has an encoder, a prediction network, and a joint network. The encoder is analogous to an acoustic model that takes a sequence of acoustic input features and outputs encoded hidden representations. The prediction network corresponds to a language model that accepts the previous output label predictions, and maps them to hidden representations. The joint network is a feed forward DNN that takes both the encoder and prediction network outputs, and predicts the final output label probabilities with softmax normalization.

\[
P(y|x) = \text{softmax} \left( W^T \text{sigmoid}(Vh) + b \right)
\]

where \(W\) is the weight matrix, \(V\) is the projection matrix, \(h\) is the hidden representation, and \(b\) is the bias vector.

![RNN-T ASR model architecture](image)

Figure 1: RNN-T ASR model architecture

2.2 Overview of learning with large batch size

When training with SGD, mini batches with a well crafted decaying learning rate schedule are commonly used as opposed to using large batches. Previous work in [33] has demonstrated a generalization drop when using large batches, thus recommending mini-batch SGD with decaying learning rate. However, recent advances in large batch training both with a linear scaling rule of the learning rate [22] and constant learning rate[53], large batch training has been shown to achieve similar performance as its mini-batch counterpart. A recurrent observation in the literature [20, 33, 37, 41, 42, 52] is that large batch training (for ImageNet, > 1000) results in test accuracy degradation. Despite the warm-up in [22], for ImageNet, the best accuracies are observed up to a large mini-batch of 8192 images.

In this paper, we deal with the challenges of 1) training with large batches in incremental learning and 2) semi-supervised learning to alleviate unavailability of human annotation and labels. For automatic speech recognition (ASR), with large batch sizes (> 3e5 utterances) using a fixed learning rate schedule, we observe better test accuracies, as opposed to the degradation in literature, while training with teacher transcripts for the incremental audio data.
3 ILASR: INCREMENTAL LEARNING FOR AUTOMATIC SPEECH RECOGNITION

This section describes the ILASR architecture and the corresponding incremental learning algorithm. ILASR offers large scale end-to-end ASR training with the ability to incrementally update the models in user-defined time windows. ILASR automates the whole lifecycle of data generation, sampling, labeling, model development, evaluation and deployment for audio data in near real-time.

3.1 ILASR Architecture

Figure 2 shows the architectural overview of ILASR. The system comprises three primary components: (1) Data preprocessor – is a cloud runtime service that processes near real-time audio from device; (2) IL Core is responsible for model training, computing model updates and inference; and (3) IL Orchestrator aggregates the accumulated gradients, updates the model, performs evaluation and finalizes the model update based on the evaluation result.

Train launcher initiates the end-to-end ASR training in ILASR. The first step is data preprocessing to select a subset of devices and utterances to participate in the training loop. The selection could be random or based on heuristics aimed at improving the model in a particular way. Confidence scores obtained during inference are used [29, 30] coupled with heuristics such as presence of rare words or semantic tags and intents of interest. This selection can be extended to leverage weak signals from user feedback such as user indicating whether the action taken by the assistant is positive or negative or detecting friction such as repeated requests or cancellations. Acoustic features are extracted and augmented [48] for the selected utterances for training. Machine transcripts are generated using a teacher ASR model pre-trained using standard distributed training. The Conformer [25] based end-to-end ASR teacher model decodes the input audio (X) to produce machine transcripts (Y). These paired (X, Y) instances are used to train the model. The machine transcripts act as ground truth labels. ILASR produces transcriptions through secure automation without human intervention or review. The extracted features together with machine transcripts in this step are combined to train the student models using IL Core. The IL Core system has an application programming interface (API) that supports local gradient accumulation on each of the servers in the fleet, and an ASR inference engine. The IL Core API supports FedSGD and FedAvg [44] and can be extended to support other federated optimizers such as FedProx [51], FedMA [59], FedNova [60], and adaptive federated optimizer [50]. The IL Orchestrator coordinates training across the ILASR fleet. IL Orchestrator contains the gradient publisher, aggregator and updates the model incrementally. The gradient aggregator collects gradients from each of the IL Core instances, aggregates them and then applies them to the current model. Once the model update is done, the collected gradients are discarded and not stored in the system which helps with reducing the risk of gradient inversion attack. A periodic light-weight evaluation of the model ensures that the model is directionally improving. The global model is updated in a given round when the performance improves over that of the previous round. To reduce the probability of a model update resulting in worse performance, ILASR can be run in parallel with differing hyperparameters. In this scenario, one of the resulting models can be utilized should it result in improved performance. After a sufficient number of rounds, the final model is stored for the next model release after a detailed model validation step.

ILASR addresses security and privacy concerns with different levels of granularity. Since ILASR is a cloud-based system for privacy-preserving IL at scale, the audio encryption is two fold. In the first stage, TLS [15] encryption is applied on audio transmission followed by an application level key-master [40] encryption. Importantly, the audio is purged in a few minutes (≤ 10), within which the model updates are calculated.

3.2 ILASR: Incremental Learning

Algorithm 1 ILASR incremental learning algorithm

Require: K servers, L loss function, N number of local steps per round, B local batch size, η learning rate, T recent utterances pulled by server k in round r, D_{eval} eval set and D_{hit} past transcribed data if used for rehearsal training.

Ensure: w^g_0 incremented updated global model and w_r word error rate on the eval set after r rounds
1: Init. w^g_0 // start training with a pre-trained model
2: w_0 = asr_inference_engine(D_{eval}, w^g_0)
3: for each round r = 1, 2, . . . do
4: for each server k ∈ ILASR Fleet in parallel do
5: w^k_r = w^g_{r-1}
6: D_{train} ← (filter P_k based on utterance selection criteria and generate machine transcript, refer algorithm 2)
7: D_{train} ← (mix D_{train} and D_{hit} if D_{hit} is used for rehearsal, else just D_{train})
8: D_{train} ← (split D_{train} into N batches of size B)
9: for each batch b1 from b1 to bN do
10: w^k_r ← optimizer_k update(η, ∇L(w; b1))
11: end for
12: end for
13: w^g_r ← 1/K Σ_k w^k_r
14: w_r = asr_inference_engine(D_{eval}, w^g_r)
15: w^g_{r+1} = w^g_r if w_r > w_{r-1} // Revert to the previous model if not a better model.
16: end for
Algorithm 1 shows the incremental learning policy in ILASR framework. The new model obtained in each round is used only if it performs better than the model from the previous round. Parallel runs of the algorithm with differing hyper parameters to train an ensemble of incrementally updated models can ensure that there is at least one model that performs better than the model from the previous round. Another interesting consideration is the effect of catastrophic forgetfulness [17, 21, 43] in incremental learning of ILASR framework, where the previous learned behaviour of a model is forgotten with new updates. This can be mitigated with the rehearsal [2] of training on a subset of annotated historical data along with the new data.

We describe the SSL data generation method in algorithm 2. We randomly sample a subset of the audio in near real-time, to prepare a data pool \( \mathcal{P} \), and calculate target number of utterances \( |\mathcal{U}| \) to be sampled from \( \mathcal{P} \), where each of the utterances include a pre-calculated confidence value \([57]\). For each confidence bin, for example confidence in \([600, 700]\) where confidence is evaluated on a scale from 0 to 1000, utterances are filtered to conform to the confidence criterion. The randomly sampled utterances from above are set to get target number of utterances and sent to IL core for training, which are deleted as soon as the model takes a pass over it for the first time. Additional criteria such as presence of rare words, presence of desired semantic tags can also be utilized.

Algorithm 2: SSL data selection procedure

Require: \( r \) list of utterance confidence bins
Ensure: \( \mathcal{X} \) data set
1: \( \mathcal{P} = \text{random sample}(\mathcal{P}) \) // prepare a random pool of data
2: \( \mathcal{U} = \text{teacher_decode} (\mathcal{P}) \) // generate machine transcripts
3: \( Q = \text{filter_utterance}_\text{confidence}(\mathcal{P}) \). // prepare a random pool of data
4: \( Q[c] = \text{filter_utterances}(\mathcal{U}, c) \)
5: \( \mathcal{X} = \text{select_utterances}(Q) \) // can include additional criteria like presence of rare words or desired semantic tags
6: end for

4 EXPERIMENTS

We describe the datasets, model configurations and experimental settings used in this paper, to provide insights and study privacy-preserving incremental learning through ILASR.

4.1 Datasets

All speech data used for training and evaluation are de-identified.

Train sets: The audio streams are prepared into offline training datasets. The following training datasets are used for experimentation:

Pre-training datasets: A 480K-hour pre-training dataset is utilized for building pre-training models. This pre-trained model is used as a starting point for incremental training with the ILASR system. This comprises two datasets:

1. 120K-hour HT: Human-transcribed (HT) data from 2020 and previous years
2. 360K-hour SSL: Machine-transcribed data in 2020

Incremental training dataset: We consider the end of 2020 as the start date for incremental training of ASR models.

1. 180K-hour ILASR SSL: Machine-transcribed data is generated over a period of six months in 2021 (Jan to June) and is used for near real-time training of the ILASR system.

Test sets: We evaluate the models on in-house human transcribed (HT) test sets.

General: Includes three HT datasets from different time ranges representing the general use case. It comprises a 37-hour test set from 2021, a 10-hour test set from 2020 and a 96-hour test set from 2018 – 2019.

Rare: Includes three HT datasets from different time ranges, where the transcriptions contain at least one rare word. Rare words are those in the long-tail of the vocabulary determined by word frequency. This includes a 44-hour test set from 2021, a 44-hour test set from 2020, and a 27-hour test set from 2018 – 2019.

Delta: This consists of a 22-hour HT test set that records a change in frequency of words in 2021 over 2020. The transcriptions are filtered based on 1-gram, 2-grams and 3-grams that are 5x more frequent in 2021 than 2020. This test set captures changes in the data distribution and is very relevant to study the impact of incremental learning with ILASR.

Messaging: Includes two HT datasets that comprise of messaging and communications domain data. It includes a 2.7-hour HT test set from 2020 and a 45.5-hour HT test set from 2018 – 2019.

Monthly datasets (2021): We use six monthly test sets from Jan to June 2021 to evaluate the incremental learning setup of ILASR. Each of these datasets are referred to as (Jan, Feb, ... June) and each month has on average 70-hours of data. We further report results on 3-month datasets Jan – Mar including data from Jan, Feb, Mar and Apr – Jun including data from Apr, May, June.

4.2 Model details

Features: The audio features are 64 dimensional log-mel filter-bank energies \([46]\) computed over a 25ms window, with a 10ms shift. The features computed on 3 consecutive 10ms frames are stacked and sub-sampled to result in 192 dimensional features at a 30ms frame rate, and are provided as input to the ASR model. The ground truth transcripts are tokenized to 2500 sub-word units using a uni-gram language model \([35]\).

Models: Teacher models: Teacher models are used to generate SSL machine transcripts. We have three teacher models available: T3 is a teacher model (a conventional RNN-HMM hybrid ASR system\([6]\)) that is trained on 100K-hours of data until 2019 only. The machine-transcripts from T3 are utilized to bootstrap and provide transcripts for the more recent 360K-hour SSL pre-training dataset. The 480K-hour pre-training dataset, including the 360K-hour SSL dataset based on T3 and the 120K-hour HT dataset, is utilized to train two updated teacher models: (1) T1: A larger conformer based ASR architecture \([25]\) trained on 480K-hours. T1 has 122M parameters, an encoder with \(17 \times 512\) LSTM layers, 8 attention heads with 32 dimensional convolution kernel. The prediction network uses \(2 \times 1024\) LSTM layers. (2) T2 is a conventional RNN-HMM hybrid ASR system \([6]\) and is trained on the same 480K-hour dataset. Finally, the student models for all experiments in the paper are trained on SSL datasets that use the most recent T1 teacher model.
In section 5.1.3, for the purpose of ablations comparing various teachers, we train student models on SSL datasets that are based on T2 and T3. 

Student models: The student models are based on different LSTM based RNN-T architectures. These vary in the number of encoder layers and the feature frame rates. Two student models are described as follows. \textit{rnn\_60m} contains 60M parameters with 5 x 1024 LSTM encoder, 2 x 1024 LSTM prediction network and a feed-forward joint network with tanh activation. The input embeddings of the prediction network are 512 dimensional. SpecAugment \cite{park2019specaugment} is used on the audio features. \textit{rnn\_90m} contains 90M parameters with 8 x 1024 LSTM layer encoder, a prediction network of size 2 x 1024, and a feed-forward joint network with tanh activation. The input embeddings of the prediction network use 512 dimensional embeddings and a 2500 sub-word tokenizer from a uni-gram language model. SpecAugment is used on the audio features. The encoder uses an LSTM based time-reduced \cite{graves2014towards} RNN multi-layer (for speed of training and inference) with feature frame rate set to 3 layers. Each of these feature frame layers have 1536 units and the LSTM projection with a size of 512.

The models \textit{rnn\_90m} and \textit{rnn\_60m} are pre-trained on both the HT data of 120K hours and 340K hours of SSL data generated using the teacher (T1) decoded labels. The human transcribed data used in the pre-training utilizes data up to the end of 2020, while the SSL data is in 2020. For our experiments in this paper, we further train the above pre-trained RNN-T student models using a total amount of 180K hours of SSL data (teacher generated labels) available in a time-window of 6 months in 2021.

Training details: We use the following parameters to train both the teacher and student models. The system is run on a fleet consisting of 200 nodes. We adopt a learning rate schedule of warm-up where $lr = 1e^{-7}$ for the first 3000 steps, followed by constant learning rate of $5e^{-4}$ till 30k steps, then exponential decay ($lr = 1e^{-5}$) from 50k to 750k steps with Adam optimizer (hyperparameters are $\beta_1 = 0.9, \beta_2 = 0.99$).

We experiment with multiple large batch sizes (9k, 18k, 73k, 147k, 215k, 307k) through gradient accumulations. Note that these accumulations have an implicit effect of changing the gradient values due to the summation of gradients across a large batch. We process large batches without altering the lr schedule while accumulating the gradients. The performance of these models is measured in terms of relative word error rate reduction (WERR) over the corresponding baselines. WER is the ratio of edit distance to sequence length, where edit distance is the length of the shortest sequence of insert, delete and substitution operation on transforming a pre-dicted sequence to target.

| Time      | Test-set | ILASR             |
|-----------|----------|-------------------|
|           |          | replay | no replay |
| 2021      | Rare     | 0.72% | 0.66%    |
|           | Delta    | 20.10% | 23.99%   |
|           | General  | 1.23% | 0.41%    |
|           | Jan-Mar  | 1.25% | 1.50%    |
|           | Apr-Jun  | 2.73% | 3.09%    |
| 2020      | Rare     | 0.62% | 0.62%    |
|           | General  | 0.00% | -0.72%   |
|           | Message  | -0.83% | -2.84%   |
| 2018-2019 | Rare     | -0.63% | -0.63%   |
|           | General  | -1.21% | -2.6%    |
|           | Message  | -2.82% | -3.42%   |

Figure 3: Monthly WERR (%) for incremental learning in ILASR for \textit{rnn\_60m} on six of the monthly test sets (Jan–Jun) when measured relative to the starting model versus the one trained incrementally in each month.

Table 1: Relative % WER improvements from the initial model when trained with the ILASR system

5 RESULTS & DISCUSSION

In this section, we analyze the performance of incremental learning in ILASR. In particular, we analyze the performance of incremental learning in ILASR in terms of relative word error rate reduction (WERR) in comparison with the initial pre-trained student models as baselines.

From Table 1, we see that ILASR improves a strongly trained base model by up to 3% on test sets in 2021 which climbs to 20% on the delta dataset that consists of new or trending words and phrases. At the same time, performance on older general and tail test sets do not see much degradation.

Catastrophic forgetting is one of the issues incremental learning needs to circumvent in order to have consistent performance across both old and new data. In Table 1, we compare the performance of replay based incremental learning, where a sub-sampled portion of 120K-hour human-transcribed data is also consumed in model training while the no replay counterpart does not involve that. As demonstrated in Table 1, replay based training tends to outperform its no replay counterpart on older test sets as expected from IL literature.

Next, we evaluate the incrementally trained ILASR models on fine-grained test sets that are prepared in each of the six months (Jan-Jun) of 2021, see Figure 3. For all the evaluations in Figure 3, we report the WERR in each month relative to the initial pre-trained model (for example, WERR in May is the relative difference between the WERs of May model and the pre-trained). The results show incremental improvements in performance on all the six monthly test sets from month to month in the ILASR training. This suggests
that the incremental training helps in capturing the new trends in time periods while the model is adapting to the incremental changes in the data. It is also noteworthy that the incremental improvement does not come at the cost of catastrophic forgetting. More interestingly, the models trained with the data until May/June degrade the performance on June test-set, which improves after the model is trained on the data available from May/June. This clearly suggests the adaptive nature of capturing the shifts in data in the new time periods in ILASR.

To further strengthen the incremental learning claims, we analyze the incremental learning patterns for a longer duration in the time-periods between Jan – Sep in 2020. Figure 4 shows the learning patterns on a quarterly basis for the first three quarters (Q1–Q3) of 2020. In 2020Q1 and Q2, the WERR improves initially and then decreases as the incremental model training progresses on a month-over-month basis. The degradation (whilst better than the baseline) is a demonstration of forgetting as newer updates are prioritized over the months old test sets. Consequently, in 2020Q3, the performance improves without any downward trends, which is due to the fact that the models keep learning month over month while the test sets also belong to the same time-periods. These trends suggest that the proposed techniques help in incrementally improving the performance even in longer time-periods while limiting the regressions on the older eval data.

Next, we explore several design choices which play a key role in the performance of ILASR and share our insights in terms of the design choices.

5.1 Design Choices: ILASR

We explore the following design choices in the context of the ILASR framework: 1) effect of large batch sizes on performance of the student models; 2) temporal effects on processing the data in ILASR; 3) analyze the importance of different teacher models in ILASR.

5.1.1 Training is robust to large batch sizes. We use large batches in ILASR via gradient accumulations. As the effective batch size increases, the number of optimization or update steps reduces as the same amount of data is processed. Larger batch sizes would require fewer optimization steps and vice versa for the same amount of data. Use of large batches accelerates the training (shown in [64]), which is similar in ILASR. The reason large batch sizes are relevant in the ILASR system is that there are limitations about how quickly gradients can be aggregated and the global model distributed to the servers in the fleet. Hence, a limited number of update steps can take place in a time period compared to GPU-based offline distributed training. Moreover, as data arrives in a streaming fashion and is not persisted, it needs to be consumed as and when it arrives, in near real-time. For each of the limited number of updates, a large amount of streaming data is available.

We explore the trade-off between large batches and model performance. Table 2 shows the effect of large batches on performance of a student models trained in ILASR. The performance (WER) is relative to the corresponding pre-trained student model. This baseline is weaker, hence improvements are larger. We find that increasing the batch multiplier (effective batch size) has insignificant effect on WER. As batch sizes increase from 9K to 300K utterances, the difference in the accuracies is insignificant.

More importantly, this finding is in contrast to the test accuracy degradation effects reported in literature [20, 22, 33, 37, 41, 42, 52, 56] with the use of large batches. We observe that such degradation is not evident for model training in ILASR. Although, the attempts in the literature have no strong mathematical justification, Goyal et al. [22] reasoned the performance degradation to optimization issues, thereby using warm-up to mitigate the degradation. Similarly, in our case, we attribute the gains and/or no performance degradation to the following factor. The initialized models are pre-trained that have converged on the data from a previous time period as opposed to random initialization in the large batch training in

Table 2: Effect of large batches on the relative improvement in performance (in terms of WERR, %) of all the three models when fine-tuned in ILASR.

| Time      | Test-set | 9k  | 18k | 73k | 147k | 215k | 307k |
|-----------|----------|-----|-----|-----|------|------|------|
| 2021      | Jan–Mar  | 2.74 | 1.49 | 2.37 | 2.37 | 2.24 | 2.24 |
| 2018–2019 | Rare     | 3.58 | 3.58 | 3.72 | 3.65 | 3.78 | 4.72 |
|           | Message  | 6.46 | 6.05 | 9.46 | 9.46 | 9.28 | 9.37 |
|           | General  | 15.14| 15.12| 14.70| 14.56| 14.41| 14.26|

Figure 4: For the rnnt_60m, the pre-trained model is trained on the data available until 12/2019. Training this pre-trained model in incremental mode for the next nine months (Jan – Sep) in 2020. The x-axis shows the monthly incremental model, where the model from previous month is fine-tuned on the data in current month; y-axis shows the relative WER in each month w.r.t the initial pre-trained model. Each of the curves represent the test set of the corresponding month.
Table 3: Impact of the temporal order (chronological versus random) of processing the training data in ILASR for both with and without replay of the human transcriptions.

| Time     | Test-set | Chrono vs. random replay | no replay |
|----------|----------|--------------------------|-----------|
| 2021     | Rare     | -0.62%                   | -1.16%    |
|          | Delta    | -1.68%                   | -0.73%    |
|          | General  | 0.15%                    | 1.47%     |
|          | Jan-Mar  | -0.53%                   | -0.24%    |
|          | Apr-Jun  | -0.47%                   | 0.29%     |
| 2020     | Rare     | -0.56%                   | -0.90%    |
|          | General  | -0.55%                   | 1.61%     |
|          | Message  | 0.35%                    | 0.48%     |
| 2019-2019| Rare     | -0.46%                   | -1.26%    |
|          | General  | 0.32%                    | 0.67%     |
|          | Message  | -0.11%                   | -0.87%    |

literature, usually, these models are trained from scratch (despite the few initial epochs in warm-up) in the literature.

5.1.2 Impact of chronologically ordered data. One important aspect of IL is the data being processed in time as is available, chronologically. We analyze the effect of processing order (chronological vs random) for the six months in 2021. Note, random order is same as shuffling the data in regular distributed training of deep models. Chronological data is not IID across time as utterances have a correlation with the time of day (for example, requests to snooze alarms in the morning or turning smart lights on after sundown). We found that there is no difference in performance of processing the data chronologically as compared to randomly as depicted in Table 3. Moreover, in both the cases of chronological and randomized, the improvements over initial baselines are clearly evident (see Table 1).

Table 4: Performance (in terms of WERR, %) of the RNN-HMM hybrid ASR teacher (T2) and bidirectional RNN-HMM hybrid ASR (T3) based teacher models with respect to the Conformer teacher (T1). The negative (-) sign represents that T1 performs worse while the rest shows that T1 is the best performing teacher model.

| Time     | Test-set | T1 vs T3 | T1 vs T2 |
|----------|----------|----------|----------|
| 2021     | Jan–Mar  | 16.63%   | 0.14%    |
| 2018–2019| Rare     | 8.75%    | 12.02%   |
|          | Message  | 7.34%    | 14.92%   |
|          | General  | -0.89%   | 20.51%   |

5.1.3 Ablations with teachers and students. We experiment with three different teacher models that are trained for different time ranges with different architectures. This experiment helps us explore the importance of keeping an updated and more effective teacher. The three teachers are: T1 is the Conformer based that is explained earlier in section 4.2; T2 is a RNN-HMM conventional hybrid model [6]; T3 is a bidirectional RNN-HMM conventional hybrid ASR model. T1 and T2 are trained on the same amount of data until the end of 2020 while T3 is trained on the data (a total of ~100k hours of HT data) available till the end of 2019.

Table 4 compares the performance of the teacher models. On an average, T1 is better than the rest of the two teachers, T1 > T2 > T3 on new data reflecting the importance of keeping the teacher model up-to-date. Conformer based teacher, T1 is better than the rest of the remaining two teachers. The relative performance differences, when measured on the four standard test sets are, T1 is better than T2 and T3 with 11.85% and 7.96% WERR, respectively.

Table 5: The performance (in terms of WERR) of the student models when trained with the machine transcripts generated from each of the three different teacher models.

| Time     | Test-set | rntt_90m | rntt_60m |
|----------|----------|----------|----------|
| 2021     | Jan–Mar  | T1       | T2       | T3       | 7.78% | 6.27% | 2.87% |
| 2018–2019| Rare     | 5.14%    | 7.12%    | 5.60%    | 3.89% | 3.58% |
|          | General  | 4.12%    | 4.88%    | 7.12%    | 3.07% | 3.74% | 6.05% |
|          | Message  | 7.88%    | 7.30%    | 5.07%    | 5.03% | 6.55% |

Table 5 shows the WERR of two student models (rntt_90m and rntt_60m) when trained using the machine transcripts generated from the three teacher models. We observe that both the students are in similar terms of performance. On an average, for rntt_90m, T1 based training is better than T2 and T3, with 4.66% and 3.25% WERR, respectively. For rntt_60m, T1 is better than T2 and T3 with 5.96 and 2.18% relative WERR improvement respectively. The improvements are larger than in Table 1 as these experiments were done with 3 months of data using a weaker baseline. In fact, both the students have same order of performance as the teachers, that is T1 > T2 > T3 even after training in IL on new data. More important, the magnitude of improvement in student (true for both the student models) training is not of same scale as the difference in teachers. For example, Conformer based teacher (T1) is better than T3 by 7.96%, whereas rntt_90m student trained with Conformer transcripts (T1) is 3.25% better than the one trained with T3 transcripts. This suggests that better teacher models result in improving the student performance but the difference (same student trained with different teacher models) is narrower. In other words, a significantly better teacher model can have a limited impact in improving students models in ILASR.

6 RELATED WORK

SGD gradients mini and large: Stochastic gradient descent (SGD) drives the training of neural nets with mini batches. Large mini batches [22, 27, 53, 64] reduce the number of updates with a large step size. Simply increasing the batch size reduces the test accuracy [33] as the gradients get integrated. Test set accuracy can be improved with large batches that are proportional to the learning rate. This simple linear scaling is inefficient, which necessitates a warm-up phase [22]. Instead of decaying the learning rate, increasing the batch size during training [53] helps to reduce the communication steps to update the model and improves the test accuracy. Federated averaging [44] (FedAvg) follows a similar strategy of
We proposed the ILASR framework for privacy preserving incremental learning of end-to-end automatic speech recognition systems. ILASR is a big step forward for production level ASR systems, especially for automatic incremental updates of these systems. In this study of near-real time training with ILASR, we learned that even the converged production level ASR models: 1) can be improved significantly in an incremental fashion with 3% general improvements that can go up to 20% on test sets with new words or phrases; 2) training with large batches arising as a result of communication constraints does not result in degradation; 3) memory replay training is effective at mitigating catastrophic forgetting on older test sets; 4) there is no significant impact of chronological versus random processing of data in IL for speech recognition over a period of six months; and finally; 5) having a significant improvement in teacher models used to generate machine transcripts does not translate to the same scale of improvements in students.

In the future, we will explore the utility of noisy students for iterative self-learning instead of relying on teacher models in ILASR. Real-time resource-constrained on-device speech recognition is still a hard challenge. Here, we plan to further explore different directions such as finding the best hyper parameters [34], controlling leaky gradients [66], stopping gradient inversion and data leakage attacks [58], personalizing ASR depending on the device context, and using smaller teacher models or self-labelling that can be run on device. Approximate gradient computation techniques may be required with severe compute resource limitations. Further, exploring methods of integrating weak supervision information from inferred or explicit user feedback from a session of interactions as well as externally updated language models are avenues of further research.

**REFERENCES**

[1] Al-Rubaie, M. and Chang, J. M. Privacy-preserving machine learning: Threats and solutions. IEEE Security & Privacy, 17(3):49–58, 2019.
[2] Anthony, R. Catastrophic forgetting, rehearsal and pseudoexperience. Connection Science, 7(2):123–146, 1995.
[3] Baevski, A., Zhou, Y., Mohamed, A., and Auli, M. wav2vec 2.0: A framework for self-supervised learning of speech representations. Advances in Neural Information Processing Systems, 33:12449–12460, 2020.
[4] Baevski, A., Hsu, W.-N., Xu, Q., Babu, A., Gu, J., and Auli, M. Data2vec: A general framework for self-supervised learning in speech, vision and language. arXiv preprint arXiv:2202.03555, 2022.
[5] Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., Kiddon, C., Konečný, J., Mazzocchi, S., McMahan, H. B., et al. Towards federated learning at scale: System design. arXiv preprint arXiv:1902.01046, 2019.
[6] Bourlard, H. A. and Morgan, N. Connectionist speech recognition: a hybrid approach, volume 247. Springer Science & Business Media, 2012.
[7] Cai, H., Gan, C., Zhu, L., and Han, S. Tinytl: Reduce memory, not parameters for efficient on-device learning. Advances in Neural Information Processing Systems, 33:11285–11297, 2020.
[8] Castro, F. M., Marin-Jimenez, M. J., Guili, N., Schmid, C., and Alahari, K. End-to-end incremental learning. In Proceedings of the European conference on computer vision (ECCV). pp. 233–248, 2018.
[9] Chen, K. and Huo, Q. Scalable training of deep learning machines by incremental block training with intra-block parallel optimization and blockwise model-update filtering. In 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5880–5884, 2016. doi: 10.1109/ICASSP.2016.7472805.
[10] Chen, S., Wang, C., Chen, Z., Wu, Y., Liu, S., Shen, Z., Li, J., Kanda, N., Yoshio, T., Xiao, X., et al. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. arXiv preprint arXiv:2110.13900, 2021.
[11] Chu, C.-C., Samath, T. N., Wu, Y., Prabhavalkar, R., Nguyen, P., Chen, Z., Kannan, A., Weiss, R. J., Rao, K., Gonnia, E., et al. State-of-the-art speech recognition with efficient and large-volume speech processing. arXiv preprint arXiv:2110.12468, 2021.
sequence-to-sequence models. In _2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)_ pp. 4774–4778. IEEE, 2018.

[12] Chung, Y.-A. and Glass, J. Generative pre-training for speech with autoregressive predictive coding. In _ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)_ pp. 6429–6433. IEEE, 2020.

[13] Cui, X., Lu, S., and Kingsbury, B. Federated acoustic modeling for automatic speech recognition. _CoRR_, abs/2102.04209, 2021.

[14] Dey, S., Motlicek, P., Bui, T., and Dernoncourt, F. Exploiting semi-supervised training through a dropout regularization in end-to-end speech recognition. _arXiv preprint arXiv:1908.05227_, 2019.

[15] Dierks, T. and Riesco, E. The transport layer security (tls) protocol version 1.2. _Dive CCN, 2013._

[16] Dimitriadis, D., Kumatani, K., Gmyr, R., Gaur, and Eksimler, S. E. A federated learning approach in training acoustic models. In _Proc. Interspeech_, 2020.

[17] French, R. M. Catastrophic forgetting in connectionist networks. _Trends in cognitive sciences_, 3(4):128–135, 1999.

[18] Gao, Y., Parcollet, T., Fernandez-Marcues, J., de Gusmao, P. P. B., Beutel, D. J., and Lane, N. D. End-to-end speech recognition from federated acoustic models. 2021.

[19] Geyer, R. C., Klein, T., and Nabi, M. Differentially private federated learning: A client level perspective. _arXiv preprint arXiv:1712.07557_, 2017.

[20] Golmant, N., Vemuri, N., Yao, Z., Feinberg, V., Gholami, A., Rothauge, K., Mahoney, M. W., and Gonzalez, J. On the computational inefficiency of large batch sizes for stochastic gradient descent. _arXiv preprint arXiv:1811.12943_, 2018.

[21] Goodfellow, I. J., Mirza, M., Xiao, D., Courville, A., and Bengio, Y. An empirical investigation of catastrophic forgetting in gradient-based neural networks. _arXiv preprint arXiv:1312.6213_, 2013.

[22] Goyal, P., Dollé, P., Girshick, R., Noordhuis, P., Wesolowski, L., Ykola, A., Tulloch, A., Jia, Y., and He, K. Accurate, large minibatch SGD: Training imagenet in 1 hour. _arXiv preprint arXiv:1706.02677_, 2017.

[23] Granqvist, F., Seigel, M., van Dalen, R., Cahnii, A., Shum, S., and Paulih, M. Improving on-device speaker verification using federated learning with privacy. 2020.

[24] Graves, A. Sequence transduction with recurrent neural networks. _arXiv preprint arXiv:1211.3712_, 2012.

[25] Guanli, A., Qin, J., Chiu, C.-C., Parmar, N., Zhang, Y., Yu, J., Han, W., Wang, S., Zhang, Z., Wu, Y., et al. Conformer: Convolution-augmented transformer for speech recognition. _arXiv preprint arXiv:2005.08100_, 2020.

[26] Guani, D., Beaufays, F., and Motta, G. Training speech recognition models with federated learning: A quality/cost framework. In _ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)_ pp. 3080–3084. IEEE, 2021.

[27] Hoffer, E., Hubara, I., and Soudry, D. Train longer, generalize better: closing the generalization gap in large batch training of neural networks. _arXiv preprint arXiv:1705.08741_, 2017.

[28] Hsu, W.-N., Bolte, B., Tsai, Y.-H. H., Lakhotia, K., Salakhutdinov, R., and Mohamed, S. H. An empirical investigation of transfer learning with DNNs for speech recognition. In _ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)_ pp. 5206–5210. IEEE, 2019.

[29] Park, D. S., Chan, W., Zhang, Y., Chiu, C.-C., Zoph, B., Cubuk, E. D., and Le, Q. V. SpecAugment: A simple data augmentation method for automatic speech recognition. _arXiv preprint arXiv:1904.08779_, 2019.

[30] Parthasarathi, S. H. K. and Strom, N. Lessons from building acoustic models with a million hours of speech. In _ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)_ pp. 6670–6674. IEEE, 2019.

[31] Reddi, S., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Koncny, J., Kumar, S., and McMahan, H. B. Adaptive federated optimization. _arXiv preprint arXiv:2003.06295_, 2020.

[32] Sahu, A. K., Li, T., Sanjabi, M., Zaheer, M., Talwalkar, A., and Smith, V. On the convergence of federated optimization in heterogeneous networks. _arXiv preprint arXiv:1812.06127_, 3:3, 2018.

[33] Shih, C. J., Lee, J., Antognoni, J., Sohl-Dickstein, J., Frostig, R., and Dahl, G. E. Measuring the effects of data parallelism on neural network training. _arXiv preprint arXiv:1811.03600_, 2018.

[34] Smith, S. L., Kidermans, P.-J., Ying, C., and Le, Q. V. Don’t decay the learning rate, increase the batch size. 2018.

[35] Soltau, H., Hao, L., and Sak, H. Reducing the computational complexity for whole word models. _2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)_ pp. 63–68, 2017.

[36] Strom, N. Scalable distributed training using commodity gpus computing. In _Sixteenth Annual Conference of the International Speech Communication Association_, 2015.

[37] Sun, F., Feng, W., Han, R., Yan, S., and Wen, Y. Optimizing network performance for distributed dn training on gpus clusters. ImageNet/alexnet training in 1.5 minutes, 2019.

[38] Swarup, P., Raas, M., Carnella, S., Mallick, S. H., and Hoffmeister, B. Improving asr confidence scores for alexa using acoustic and hypothesis embeddings. In _ICLR_, 2017.

[39] Triastcyn, A. and Fallings, B. Federated generative privacy. _IEEE Intelligent Systems_, 35(4):50–57, 2020.

[40] Wang, H., Yurochkin, M., Sun, Y., Papailiopoulos, D., and Khazaei, Y. Federated learning with matched averaging. _arXiv preprint arXiv:2002.06440_, 2020.

[41] Wang, J., Liu, Q., Liang, H., Joshi, G., and Poor, H. V. Tackling the objective inconsistency problem in heterogeneous federated optimization. _arXiv preprint arXiv:2007.05781_, 2020.

[42] Weninger, F., Mana, F., Gemello, R., Andrés-Ferrer, J., and Zhan, P. Semi-supervised learning with data augmentation for end-to-end asr. _arXiv preprint arXiv:2007.13876_, 2020.

[43] Wu, Y., Chen, Y., Wang, L., Ye, Y., Liu, Z., Guo, Y., and Fu, Y. Large scale incremental learning with data augmentation for end-to-end speech recognition. _arXiv preprint arXiv:1908.08456_, 2019.

[44] Xiao, A., Puegn, C., and Mohamed, A. Contrastive semi-supervised learning for asr. In _ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)_ pp. 3870–3874. IEEE, 2021.

[45] You, Y., Gitman, I., and Ginsburg, B. Scaling SGD batch size to 32k for imagenet training. _arXiv preprint arXiv:1708.03888_, 6:12, 2017.

[46] Zhao, B., Mopuri, K. R., and Bilen, H. Idlg: Improved deep leakage from gradients. _arXiv preprint arXiv:2001.02620_, 2020.

[47] Zhu, L., Liu, Z., and Han, S. Deep leakage from gradients. _Advances in Neural Information Processing Systems_, 32, 2019.