The Future of Large Language Model Pre-training is Federated

Lorenzo Sani[1,2,†] lex Iacob[1,2] Zeyu Cao[1,*] Bill Marino[1,*]

Yan Gao[1,2] Tomas Paulik[1] Wanru Zhao[1] William F. Shen[1]

Preslav leksandrov[1] Xinchi Qiu[1] Nicholas D. Lane[1,2]

bstract

Generative pre-trained large language models (LLMs) have demonstrated impressive performance over a wide range of tasks, thanks to the unprecedented amount of data they have been trained on. s established scaling laws indicate, LLMs' future performance improvement depends on the amount of computing and data sources we can leverage for pre-training. Federated learning (FL) has the potential to unleash the majority of the planet's data and computational resources, which are underutilized by the data-center-focused training methodology of current LLM practice. Our work presents a robust, flexible, reproducible FL approach that enables large-scale collaboration across institutions to train LLMs. This would mobilize more computational and data resources while matching or potentially exceeding centralized performance. We further show the effectiveness of the federated training scales with model size and present our approach for training a billion-scale federated LLM using limited resources. This will help data-rich actors to become the protagonists of LLMs pre-training instead of leaving the stage to compute-rich actors alone.

1 Introduction

The impressive performance of generative pre-trained large language models (LLMs) and their multi-modal derivations largely owes to their capacity to learn representations at scale [1]. Thus, a very small number of well-resourced tech companies and institutions are using increasingly powerful computing facilities in the race to scale up LLMs and dataset sizes to achieve state-of-the-art (SOT) performance. The thousands of hours of training to convergence on thousands of specialized and well-connected hardware accelerators in a single data center incur a high energy and monetary cost [2]. Distributing training across multiple data centers in sparse geographical locations, for those companies who could afford such a thing, would drive the cost even higher due to communication overheads [3, 4].

†Lead and corresponding author: ls985@cam.ac.uk
*Equal contributors
1Department of Computer Science and Technology, University of Cambridge
2Flower Labs

Preprint. Under review.

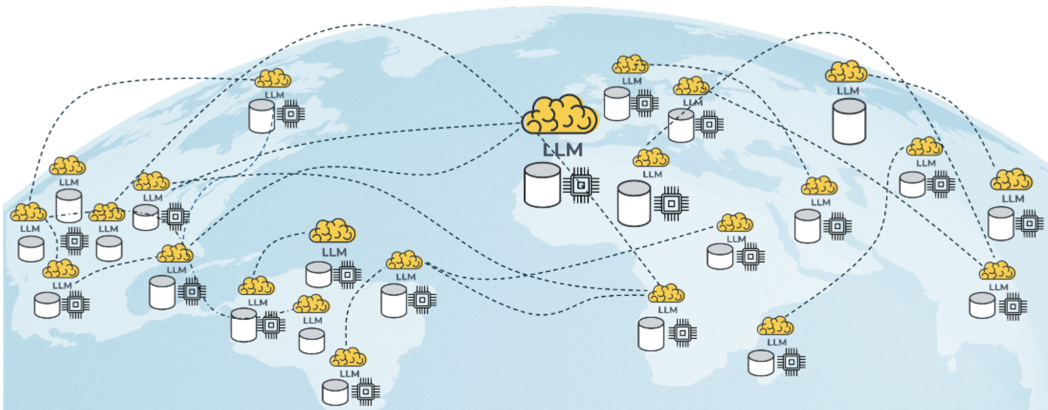


Figure 1: hypothetical representation of the available data silos around the world. While scraping data from the web has taken foundation models quite far, most data remains under private entities' control. These organizations can collaborate in the federated generative pre-training of large language models to exploit their data towards the common goal of training LLMs they control. The collaborative nature of FL and its low communication requirements make this possible with only moderately powerful hardware, eliminating the prohibitive costs of pre-training.

Hoffmann et al. [5] showed that the effective performance improvement of increasingly large LLMs requires increasingly extensive training datasets. Most of the world's data sources are unevenly distributed among private actors who often don't want to share it, even if the data regulations of their respective jurisdictions permit such sharing. Since no organization independently owns the rights to a sufficient amount of text data, the multi-terabyte datasets used in these procedures must be obtained from publicly available sources. These may include materials potentially protected by intellectual property laws [6] or be otherwise problematic [7, 8, 9, 10]. The potential means of collecting non-public data may require cutting independent deals with data providers [11] or taking model training directly to these private data sources. The further improvement of LLMs will require more extensive high-quality language data than what has been estimated [12] to be publicly available in the future, as early as 2 26. Given the limits on dataset growth coupled with near-unbounded model growth, increases in memorization and data leakage amongst current LLMs [13, 14] have started to become evident.

The next generation of LLMs and foundation models (FMs) will benefit from effectively leveraging more data and computational sources than the centralized paradigm currently makes available. Therefore, we argue that making FMs future even brighter requires shifting the dominant training paradigm to a collaborative federated approach, where nodes control considerable but not immense computational or data resources according to their abilities. Using federated learning (FL), we can expand the quality of our models by gaining access to previously untapped data and computational sources [15]. This will, in turn, allow us to increase the size of models that can be efficiently trained compared to the centralized paradigm and to avoid both memorization and data leakage.

s shown in Stich [16], Lin et al. [17], McMahan et al. [18], FL can relax the synchronization requirements of stochastic gradient descent (SGD) to accommodate such poorly connected nodes. lso, Douillard et al. [19], Liu et al. [20], Douillard et al. [21] showed that Local SGD could substantially reduce the communication overhead of training LLMs in data center settings with homogeneous and heterogeneous computational nodes. dvancing beyond their useful independent insights, we argue for an entirely federated approach to LLM training that can automatically balance the workload of multiple actors and privately merge the knowledge derived from their local datasets without divulging them directly to any other participant.

We are the **first** [22] to succeed in generative pre-training a **billion-scale model** in a **heterogeneous federated setting**. This work presents a complete system for pre-training federated LLMs in a collaborative, reproducible, and scalable fashion that can address the main challenges for a flexible execution in **arbitrary FL cross-silo settings**. It is built on the **open-source** FL framework *Flower* [23] and shall be released to the public after light refactoring. Furthermore, we make this available as a **training recipe** with a fully **transparent** experimental configuration accompanying our results. We emphasize that federated generative pre-training can be done with **affordable** hardware configurations by interlinking single nodes containing 1-8 GPUs from standard cloud providers rather than renting entire data centers at once. We demonstrate that pre-training LLMs can be democratized through FL technologies, including data-rich actors **independent** of their **connectivity and computing resources**.

Between the numerous insights discovered in developing this novel federated system, we show, for the first time, that larger federated LLMs find a consensus across clients more easily than their smaller versions, contrary to our expectations. Our **results** show that:

1. Federated LLM training offers **competitive performance** with centralized training but with far **less communication overhead**.
2. Under our system, **larger models require less frequent communication** while receiving a greater boost in generalization performance than smaller ones.
3. By utilizing our execution engine, *Pollen* [24], to balance loads across computational nodes, we can establish a training pipeline that is **faster and more robust than** its **centralized** counterpart.

2 The Landscape of LLM Training

Generative pre-trained large language models (LLMs) have demonstrated powerful performance across various natural language processing tasks, leading to rapid and widespread adoption. They are trained on massive corpora of text, incorporating mixtures of low-quality web-scraped data and high-quality curated datasets [25, 26]. Recently, several institutions have released their pre-trained LLMs, either in the closed form, such as GPT-4 [26], Chinchilla [5], and Gemini [27], or the open-sourced form, such as BLOOM [28], LLaMa [29, 30], and Falcon [31]. The scaling laws identified by Kaplan et al. [1], Hoffmann et al. [5] dictate that model size and dataset size should be increased in equal measure to improve model performance best. These suggest a future race between entities interested in developing state-of-the-art LLMs to grab as many compute and data sources as possible. The promising direction in which this field is headed can become even more luminous by gaining the trust of private entities, which possess an unprecedented breadth of knowledge and computing resources [15].

In Section 2.1, we will describe the current landscape for generative pre-training of LLMs with particular attention to the techniques for centralized distributed training. We will discuss federated learning (FL) and its impact as a communication efficient technique in Section 2.2. Section 2.3, for completeness, presents the effort the community has put into matching FL with LLMs fine-tuning. However, we highlight that our work tackles the far more challenging problem of federated pre-training of LLMs.

2.1 Centralized Distributed Optimization

LLM pre-training is based on two denoising pillars: leveraging huge batch sizes and very long contexts, i.e., sequence length of a single input sample. The combination of model and batch sizes forces training LLMs to scale SGD beyond the confines of a single GPU. Thus, it is often necessary to process more training samples in parallel and to split the model across GPUs if it is incapable of fitting within the memory of one GPU. In the following, we present a subset of optimizations and techniques that have been proposed to make this challenging training recipe possible. s we will see in Section 2.2, Federated Learning follows as a natural next step in the sequence of distributed training optimizations previously adopted by the field. Given the importance of such optimizations even for local training in federated learning, our work supports most of them.

2.1.1 Data and Model Parallelism

The number of trainable parameters and the size of the datasets make LLM training very sensitive to the stochastic fluctuations of the optimizer used, thus requiring a strong and robust regularization achieved by the denoising properties of very large batch sizes. To enable training with larger effective batch size, Data Parallelism replicates the model N times across different devices and feeds each replica with training batches in parallel; thus, each replica produces the gradients of its training micro-batch locally and accumulates them via simple summation. This accumulation is eventually followed by a synchronization step where these gradients are averaged and applied to the model. Since the batch size of an ML algorithm plays a crucial role in the convergence of the training procedure [32, 33], this synchronization step happens after each replica has accumulated sufficient micro-batches to match the desired true batch size. Modern data parallelism implementations such as the one used by PyTorch Distributed [34] use the Ring llReduce algorithm popularized by Horovod [35], implemented with low-level collective communication libraries like NCCL, to reduce the gradients across replicas. Ring llReduce scales proportionally to the amount of data being transmitted and the latency of the slowest connection between workers in the ring, making it highly sensitive to the network topology connecting GPUs.

In the case of LLM training, the model size often mandates computing gradients over batch sizes larger than what GPUs can support, even with Data Parallelism. Thus, they are forced to use a micro-batch size dictated by the number of training samples that fit in the GPU, which is, in turn, limited by the model's VR M occupancy and activations. These factors result in a complex interplay between the model's memory consumption and the training pipeline's efficiency. This is exacerbated when considering how models may be partitioned across GPUs to reduce their VR M consumption.

For sufficiently large models, the parameters must be split across GPU workers so that they fit in VR M. Traditionally, this was achieved via Model Parallelism (MP) [36, 37], splitting individual tensors and their computation vertically across GPUs and communicating as needed. While model parallelism can reduce the per-GPU memory requirements linearly by the model parallelism degree N_m, it induces communication overheads that are tolerable in a single-machine context with high inter-GPU bandwidth but cannot scale to multi-machine contexts.

2.1.2 Fully Sharded Data Parallelism

n alternative approach is to shard the model into equally-sized units amongst GPUs, with units potentially containing multiple layers, and then materialize the units, as necessary, to compute the activations during the forward pass via collective communication. This form of fully-sharded data parallelism [38, 39] reduces memory consumption linearly in the data-parallelism degree N while increasing communication by 1.5 compared to standard data parallelism. It is also possible to combine this methodology with other techniques for reducing memory consumption, such as model parallelism, activation checkpointing [40], or CPU offloading [41]. ctivation checkpointing functions by recomputing activations during the backward pass rather than saving them in memory. CPU offloading refers to offloading either data to system R M or operations to the CPU.

Given the memory requirements, it is important to consider the minimum number of GPUs necessary to train one model with reasonable efficiency, e.g., without extreme CPU offloading. This provides a lower bound on the hardware that an actor requires to participate in any distributed training of an LLM and is generally determined by the model size and the micro-batch size. By accounting for each actor's independent resources and manipulating the amount of local computation they do together with the micro-batch size, our method can relax the lower bound within reasonable limits, thus allowing even actors with weak hardware to participate in distributed training.

2.1.3 Bottlenecks for generative pre-training of LLMs

High-quality public language data is liable for exhaustion within the next decade, while low-quality language data may be exhausted in a few decades [12]. When taken together with a growing interest from both individuals and corporations in constraining what data can be scraped from the internet or used to train an LLM, this limits the size of models that can be efficiently trained. Circumventing this either requires costly independent deals with data providers [11], leaps in the effectiveness of synthetic data generation for model training [42], or significant improvements in the data efficiency of ML optimization.

Similarly, hardware accelerators with enough memory and throughput to support LLM training are scarce and increasingly unaffordable to all actors except for the best-funded organizations. Hundreds to thousands of such accelerators are required with extremely high monetary costs for training and inference [28, 43]. Moreover, such accelerators must be largely uniform in specifications to avoid stragglers and need to be extremely well-connected both within a computational node and across nodes due to the synchronization requirements of data parallel or fully shared data parallel SGD [34, 38, 39]. These constraints pose significant system challenges that can be mitigated by extremely expensive configurations or by developing heterogeneity-aware pipelines that exploit the resources available despite their heterogeneity. The difficulties described above scale with model size, as splitting the model across the memory of several GPUs further increases communication demands [38, 39] and the optimization gaps.

2.1.4 Mitigation of LLMs demands

The wider I community has gone to great lengths to circumvent the high requirements of LLM utilization and thus expand the data pool we can draw upon. Such endeavors have widely focused on locally running efficient inference with pre-trained model weights [29, 30], quantization [44], or parameter-efficient fine-tuning [45]. The recent work of Borzunov et al. [46] proposes Petal, which enables wide-scale collaboration for inference and parameter-efficient fine-tuning over the Internet by joining the resources of multiple parties. Their work assumes that clients provide inference jobs or data for fine-tuning, and servers execute the LLM inference/fine-tuning in a distributed pipeline parallel fashion. To enable efficient fine-tuning, their system makes clients responsible for holding the trainable parameters, with servers merely running forward passes and returning gradients for the pre-trained weights they store.

We argue that while methods exploiting pre-trained weights are highly beneficial to the wider community, they do not resolve the bottleneck of pre-training. Thus, the community is bound to the decisions made by actors capable of training LLMs regarding their structure and the data they use, and they may suffer downstream consequences for such reliance. This may happen since the pre-trained model, further fine-tuned for a specific downstream task, is the dominant upper bound for the downstream model's performances. Thus, we propose developing systems capable of the distributed *pre-training* of LLMs that can accommodate the variety of hardware and data available in the I community. This may include training from scratch if the model's size permits, starting from pre-initialized weights and retraining the entire model instead of only a limited subset of parameters.

2.2 Federated Learning and Local SGD

Traditional machine learning involves using a central server that hosts the machine learning models and all the data in one place. In contrast, in Federated Learning (FL) [18] frameworks, client devices collaboratively learn a shared global model using their local compute and data.

FL aims to collaboratively learn a global model while keeping private data on the device. FL training occurs over multiple communication rounds. During each round, a fraction of the clients are selected and receive the global model from the server. Those selected clients then perform local training with their local data before sending the updated models back to the central server. Finally, the central server aggregates these updates via averaging [18] into a **pseudo-gradient**. It then uses a federated optimizer [47] to update its model based on the pseudo-gradient, creating a new global model. Then, this three-stage process is repeated.

Federated optimization has several properties that make it suitable as a new paradigm for LLM training: (a) it does not require the private data of participants to be directly shared, (b) it can naturally incorporate Differential Privacy [48] or Secure ggregation [49] to compile with privacy regulations at an actor level, (c) it allows for more control over the optimization and has less restriction on the connectivity as each data-source can be associated with a series of updates. Crucially, since FL allows previously unseen data to be accessed during training, it reduces the likelihood of data memorization and leakage, which have become increasingly common as model size has increased [13, 14].

Despite these advantages, FL comes with two major challenges in the form of data and systems heterogeneity [50]. Data heterogeneity refers to the tendency of naturally generated and partitioned data to fail the IID assumption, which is common in centralized ML optimization. Systems heterogeneity refers to the ability of client hardware to vary in terms of computational ability, communication efficiency, or availability [51]. Both forms of heterogeneity are highly relevant for the distributed training of LLMs. Data heterogeneity may arise from participants holding texts in different languages, belonging to different genres, or varying in complexity. Systems heterogeneity is primarily present in the computational ability of the GPUs held by a specific client, their VR M, and number, as well as the communication efficiency of said client.

Local SGD [16, 52] is a data-parallel training paradigm where each replica applies independent gradient updates to its parameters for several local steps before averaging parameters rather than gradients. While mathematically equivalent to the Fed vg [18], the context in which it is applied, lowering the communication costs in centralized training, lacks the hardware and potential data heterogeneity specific to FL clients. Unlike previous work [19], we intend to go beyond local SGD towards fully federated training inclusive of the broadest possible range of participants.

2.3 Federated Fine-tuning and Parameter Efficient Fine-tuning of LLMs

Until now, full federated pre-trained LLMs have not been accomplished because researchers could not solve the dual challenges of its communication overhead and pre-training large models on resource-challenged devices. That said, researchers, eager to reap the benefits of federated learning, have nonetheless made progress on federating downstream LLM training tasks whose computational and communication demands are lower, such as fine-tuning, parameter-efficient fine-tuning (PEFT), and prompt-tuning.

For example, Hilmkil et al. [53] use FL to fine-tune all parameters of LBERT [54] and BERT [55], reaching 90% of the accuracy achieved by a centrally trained model on text classification tasks. Meanwhile, Riedel et al. [56] found that BERT fine-tuned in an FL scenario could perform as well as a centralized model on multilingual text classification. Wang et al. [57] and Weller et al. [58] also conducted federated fine-tuning on private data but preceded it with centralized pre-training on public data, finding that this workflow improved fine-tuning accuracy, even when the pre-training was only being done on a 1% sampling of a large public corpus or if the client-side data was non-IID, respectively.

Much progress has also been made on federated PEFT, whose computational and communication hurdles are lower than those of federated fine-tuning. Researchers have shown that a model that has been subject to federated PEFT can outperform the original pre-trained model [59], outperform siloed client models [60], and even outperform federated fine-tuning [61, 62], including in non-IID scenarios [63], but with far lower computation and communication costs because clients only need to update and transmit the smaller set of parameters. Federated LoR , for example, may consume as little as 0.058% of the communication cost of federated full fine-tuning [60]. To reduce these costs even further, Xu et al. [64] add Noise Contrastive Estimation to reduce memory demands andXu et al. [65]add backpropagation-free training to improve memory and time efficiency. Meanwhile, to address the greater impact that non-IID client distributions can have on federated PEFT performance, Babakniya et al. [66] precede federated LoR with a round of federated efficient sparse fine-tuning, reducing the performance gap while keeping combined training time low. Differently, Kim et al. [67] abandon a global model altogether, instead using FL to train client-specific models that benefit from sharing data via FL but are more resilient to client drift.

Researchers have also demonstrated that federated prompt tuning, wherein clients tune a set of continuous soft prompts that are appended to input prompts, can perform as well as the PTM [61], the siloed client models [60] and even federated full fine-tuning [68, 69] while lowering communication costs. ccompanied by an adaptive optimization method, it may also reduce the impact of client drift due to non-IID data [70, 71]. P-tuning, a variety of prompt tuning that concatenates discrete text prompts with the continuous prompt embeddings, was also shown to perform as well as the PTM [61] or local transfer tuning [71] on certain tasks, while significantly reducing the number of model parameters that clients must train and transmit.

3 Design Principles for Federated Generative Pre-Training of LLMs

We propose a federated LLM generative pre-training paradigm to create new pre-trained models over which the previously mentioned fine-tuning techniques may be applied. Thus, we aim to disentangle the broad, distributed community of researchers and practitioners that build upon LLMs from the willingness of large organizations to provide open-source weights.

This section proposes a series of inclusive principles necessary for such federations to be effective and incorporate them into our system design. The principles we chose are meant to tackle the foundational issues of federated LLM pre-training with a particular focus on data and hardware inclusivity, robustness, and efficiency.

Broad ccess to Data: The ability to train an LLM should depend on the data that an actor or set of actors possess rather than unrestricted access to hardware. Thus, we aim to transform organizations that right now can serve as data providers only [11, 72], when their ownership of the data is respected, into active participants in the LLM training process. We believe that incorporating such actors directly into the training process and offering them an incentive to participate, obtaining a model performing well on their data, is the natural next step in the proliferation of generative I generally and LLMs in particular.

Limited Communication Requirements: Training should be possible without the strong synchronization requirements of standard data-parallel training [38] to accommodate geographically distributed and poorly connected participants. Our federated learning solutions allow orders-of-magnitude reductions in communication frequency compared to centralized solutions. While such improvements provide an efficiency boost to all actors in the federation, they also provide particular benefits to data sources that may have been underrepresented in the past. For example, incorporating private data silos from regions with a lower internet presence may help alleviate the challenge of NLP for low-resource languages [73, 74].

Broad Hardware Inclusivity: Organizations with valuable data must be able to participate in the federated network, even with limited hardware resources. We define the minimum set of hardware resources based on the memory requirements of the model; i.e., their GPUs must have sufficient VR M to hold the model and at least one sample. If a client suffers from either computational or communication limitations, we account for it by adjusting the maximum round duration or allowing them to modulate the amount of local training they undertake. Thus, our system can accommodate a broad set of potential participants with hardware varying from a few medium-powered GPUs to small data centers.

Scalable Local Training Pipelines: Given that the participants in federated training are likely to vary in terms of model-size requirements, for example, if they prefer smaller models capable of faster training more specific to their data or more appropriate for the amount of data held by each participant, a federated training system must be effective across a wide range of model sizes. Thus, we build our system to support local training pipelines appropriate for all scales from billions-sized models with fully-shaded data parallelism [38, 39], to simple data parallel training [34] and CPU offloading [41]. While this work focuses on the most challenging large-scale setting, our codebase is highly adaptable to all requirements.

4 Evaluation

We have experimentally validated the efficacy of our federated training recipe at a series of increasing model scales while investigating multiple impactful choices for both the federated and local optimization.

For all our experiments, we use a version of C4 [76] randomly split into 8 equally-sized clients trained across a variable number of rounds depending on the model size. ll experiments use 5

Size	**Server**			**Local**					
	#Rounds	η	μ	η				MAX	**Batch Size**
75M	5	.1	.9	4.	1 ^{4}	1	1 ^{6}	88	256
125M	25	.1	.9	3.	1 ^{4}	1	1 ^{5}	15	256
35 M	4	.1	.9	3.	1 ^{4}	1	1	13 4	256
1.3B	16	.7	.9	2.	1 ^{4}	1	1	24 8	512

Table 1: Our scalability experiments' model sizes and afferent hyperparameter settings. We use common parameters for the damW [75] optimizer for the local training. We reset the local optimizer states every round as it helps convergence after aggregation. The centralized model uses the same local hyperparameters.

local steps per round executed by all the clients in the federation. We distinguish between *parallel* steps and *sequential* steps since the model parameters are averaged after each round, and each of the 5 local steps done during that round is done in parallel, as in standard data-parallel training [34]. Our setting respects the conditions of standard cross-silo [50] FL in which the orchestrator samples the entire batch of equally capable clients in every round. Our experiments used diverse hardware resources whose specifications allow for a reasonably fast execution. Our models have been trained on a combination of heterogeneous servers equipped with NVIDI 40, 100, and H100 GPUs. Due to FL's lax synchronization requirements, these heterogeneous hardware accelerators could collaborate despite being located in different countries.

Larger Models Improve Consensus: Our experiments focused on scaling the model size to determine if federated optimization can train models approximating the size of those commonly used today [29, 30]. s shown in Table 1, we trained models ranging in size from 75 million parameters to 1.3 billion.

We observed that the stability of the training procedure improves directly with the model size, with federated optimization reaching a better consensus across client models. s shown in Fig. 2, while the centralized model's final performance is slightly greater for small model sizes, it becomes identical for the 1.3B model. Furthermore, client training perplexities showcase a shift from a transient oscillatory phase to a convergence phase where federated optimization acts as a regularizer, improving performance. Thus, while aggregation increases local perplexity in the early stages, it decreases it at convergence. This transition is much quicker for larger models, with the 1.3B spending only 4 rounds in the pre-convergence phase, while the 75M requires over 2 rounds to reach a consensus across client models. Similarly, while the 75M model continues to experience occasional turbulences in convergence after round 2 , the larger models, such as the 35 M, obtain **full** client model alignment.

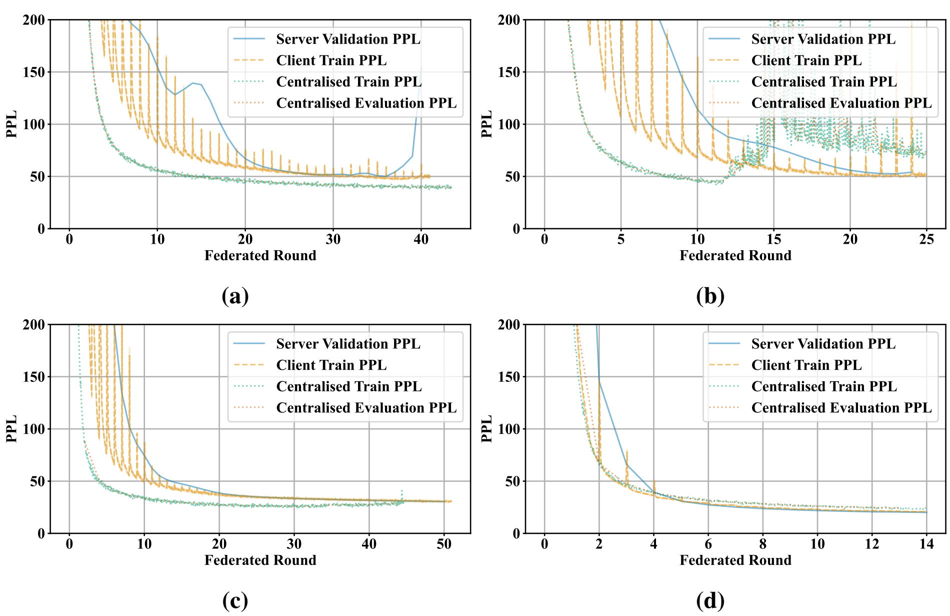
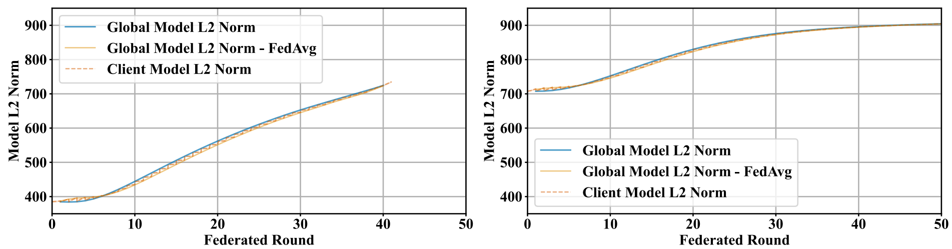


Figure 2: Perplexity comparison between the server model evaluated on the centralized validation set, the train and test perplexities for a centralized baseline, and the training perplexities of clients for our 75M (a), 125M (b), 35 M (c), and 1.3B experiments. Crucially, the stability of federated training increases with model size. For example, the centralized model outperforms the 75M federated while performing near-identically for the 1.3B case. While federated aggregation initially causes large spikes in client perplexity, these subside as the clients reach a consensus on the model parameters, which happens much quicker for larger models. Following this transitory phase, aggregation applies a regularizing effect on the model performance, allowing a better model to be trained than would be possible for a single client. The server validation perplexity is a soft upper bound for the spikes, with the gap between train and validation perplexities decreasing over time.

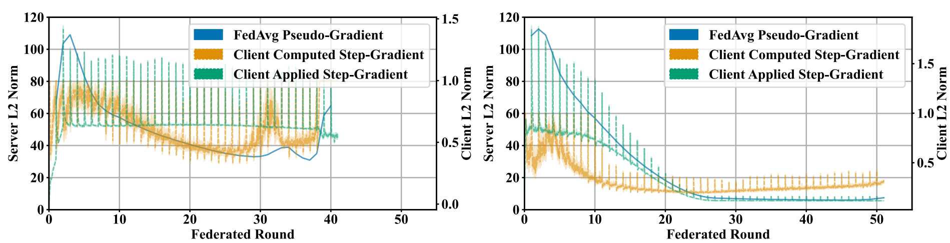
Interplay of Client and Server Models: We further investigate the training dynamics of the models of size 75 and 35   million trainable parameters. In particular, we are interested in understanding how the client and server models interplay as the FL converges.   s Fig. 3 shows, the server model initially "pulls back" the clients's model norms through the aggregation.   fter a few rounds, the federated aggregation starts incrementing the norm of the averaged clients' models until the global and local models converge to the same norm. Figure 3 showcases how the client and server optimizations interplay to agree on an initial set of global model parameters and then converge to an optimum.



(a) (b)

Figure 3: The l_2 norms of the global model, client models, and the average of client models for our 75M (a) and 35 M (b) experiments. While in the transitory early stage, the server model grows slower than the aggregate of client models, it is consistently larger in later stages. This reflects both clients reaching a consensus and the ability of the momentum mechanism to stabilize the optimization trajectory.

Federated Optimization   ligns Client Gradients: The cause of client model convergence can be observed in Fig. 4, which showcases the relationship between the norms of the per-round **pseudo-gradient** (i.e, average client update) and the local client gradients applied at every SGD step. While the pseudo-gradient starts much larger than the applied local gradients, it reaches a similar or much smaller size as the client models converge.



(a) (b)

Figure 4: The l_2 norms, for our 75M (a) and 35 M (b) experiments, of the Fed vg Pseudo-Gradient (average of client deltas relative to the server model of the previous round), the client model gradients computed on a per-step basis, and the client gradients applied to the model when considering learning rate, weight decay, and clipping. The decay in the Fed vg pseudo-gradient is much faster than in the local gradients. We postulate that averaging cancels conflicting and potentially noisy optimization directions.

5 Future Work

This work presents a complete system for generative pre-training of LLMs across a federation of clients, each participating with its own computing and data sources. In this section, we present a summary of the future direction and possibilities this work will allow. We plan to fine-tune these models on established benchmark tasks to assess further their performance in the broad range of downstream applications the participants may be interested in. We also plan to scale up the federated setting presented in this work regarding the population and model size. In addition, we will conduct further investigations into the heterogeneity of data sources, such as different languages or genres, and how this inherent property of FL impacts the capabilities of these federated models. These investigations will illuminate the potential advantages of collaboration compared to the isolated centralized paradigm. We expect FL's benefits to grow only as we incorporate more extensive private datasets into the training procedure. Since FL has been introduced as a privacy-preserving by-design technique, we will analyze this paradigm's potential privacy-preservation strengths and weaknesses.

s we advance the proposed system, we will delineate and develop further optimizations to mitigate the few overheads remaining in the execution runtime. This will eventually make our proposal more efficient and effective for allowing an even broader participation of the FL.

6 Conclusions

We successfully demonstrate the potential of LLM federated generative pre-training by obtaining the first federated billion-scale model fully trained in a heterogeneous FL setting. The complete system presented in this work allows for collaborative, reproducible, and scalable pre-training of LLMs, releasing the computational and data resources spread across the planet. Our system is built on an **open-source** framework [23] and will soon be made publicly available. Furthermore, we fully disclose our training recipe to enable future development and collaborations. Our work advances LLM pre-training by democratizing it through federated technologies, allowing for data-rich actors to pool together resources despite their individual compute and connectivity capabilities. Given current trends, we believe that the future of LLM pre-training lies in the data sources that FL can bring together.

References

[1] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, lec Radford, Jeffrey Wu, and Dario modei. Scaling laws for neural language models. *CoRR*, abs/2001.08361, 2020. URL `https://arxiv.org/abs/2001.08361`.

[2] R. F. Berriel, . T. Lopes, . Rodrigues, F. M. Varejão, and T. Oliveira-Santos. Monthly energy consumption forecast: deep learning approach. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 4283–4290, 2017.

[3] Xuanyu Cao, Tamer Basar, Suhas N. Diggavi, Yonina C. Eldar, Khaled B. Letaief, H. Vincent Poor, and Junshan Zhang. Communication-efficient distributed learning: n overview. *IEEE J. Sel. reas Commun.*, 41(4):851–873, 2023. doi: 10.1109/JS C.2023.3242710. URL `https://doi.org/10.1109/JSAC.2023.3242710`.

[4] Elia Guerra, Francesc Wilhelmi, Marco Miozzo, and Paolo Dini. The cost of training machine learning models over distributed data sources. *IEEE Open J. Commun. Soc.*, 4:1111–1126, 2023. doi: 10.1109/OJCOMS.2023.3274394. URL `https://doi.org/10.1109/OJCOMS.2023.3274394`.

[5] Jordan Hoffmann, Sebastian Borgeaud, rthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa nne Hendricks, Johannes Welbl, idan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, urelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models. *CoRR*, abs/2203.15556, 2022. doi: 10.48550/ RXIV.2203.15556. URL `https://doi.org/10.48550/arXiv.2203.15556`.

[6] Michael M Grynbaum and Ryan Mac. The times sues openai and microsoft over a.i. use of copyrighted work, Dec 2023. URL `https://www.nytimes.com/2023/12/27/business/media/new york times open ai microsoft lawsuit.html`.

[7] Vasilisa Bashlovkina, Zhaobin Kuang, Riley Matthews, Edward Clifford, Yennie Jun, William W. Cohen, and Simon Baumgartner. Trusted source alignment in large language models. *CoRR*, abs/2311.06697, 2023. doi: 10.48550/ RXIV.2311.06697. URL `https://doi.org/10.48550/arXiv.2311.06697`.

[8] Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross J. nderson. The curse of recursion: Training on generated data makes models forget. *CoRR*, abs/2305.17493, 2023. doi: 10.48550/ RXIV.2305.17493. URL `https://doi.org/10.48550/arXiv.2305.17493`.

[9] Meera Desai, Irene V Pasquetto, bigail Z Jacobs, and Dallas Card. n archival perspective on pretraining data. *Patterns (N Y)*, 5(4):100966, March 2024.

[10] Florian Tramèr, Gautam Kamath, and Nicholas Carlini. Considerations for differentially private learning with large-scale public pretraining. *CoRR*, abs/2212.06470, 2022. doi: 10.48550/ RXIV.2212.06470. URL `https://doi.org/10.48550/arXiv.2212.06470`.

[11] Open I, Dec 2023. URL `https://openai.com/blog/axel springer partnership`.

[12] Pablo Villalobos, Jaime Sevilla, Lennart Heim, Tamay Besiroglu, Marius Hobbhahn, and nson Ho. Will we run out of data? an analysis of the limits of scaling datasets in machine learning. *CoRR*, abs/2211.04325, 2022. doi: 10.48550/ RXIV.2211.04325. URL `https://doi.org/10.48550/arXiv.2211.04325`.

[13] Sara bdali, Richard narfi, C. J. Barberan, and Jia He. Securing large language models: Threats, vulnerabilities and responsible practices. *CoRR*, abs/2403.12503, 2024. doi: 10.48550/ RXIV.2403.12503. URL `https://doi.org/10.48550/arXiv.2403.12503`.

[14] Jaydeep Borkar. What can we learn from data leakage and unlearning for law? *CoRR*, abs/2307.10476, 2023. doi: 10.48550/ RXIV.2307.10476. URL `https://doi.org/10.48550/arXiv.2307.10476`.

[15] Sixing Yu, J. Pablo Muñoz, and li Jannesari. Federated foundation models: Privacy-preserving and collaborative learning for large models. *CoRR*, abs/2305.11414, 2023. doi: 10.48550/ RXIV.2305.11414. URL `https://doi.org/10.48550/arXiv.2305.11414`.

[16] Sebastian U. Stich. Local SGD converges fast and communicates little. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, L , US , May 6-9, 2019*. OpenReview.net, 2019. URL `https://openreview.net/forum?id=S1g2JnRcFX`.

[17] Tao Lin, Sebastian U. Stich, Kumar Kshitij Patel, and Martin Jaggi. Don't use large mini-batches, use local SGD. In *8th International Conference on Learning Representations, ICLR 2020, ddis baba, Ethiopia, pril 26-30, 2020*. OpenReview.net, 2020. URL `https://openreview.net/forum?id=B1eyO1BFPr`.

[18] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise guera y rcas. Communication-efficient learning of deep networks from decentralized data. In *rtificial intelligence and statistics*. PMLR, 2017.

[19] rthur Douillard, Qixuang Feng, ndrei . Rusu, Rachita Chhaparia, Yani Donchev, dhiguna Kuncoro, Marc' urelio Ranzato, rthur Szlam, and Jiajun Shen. Diloco: Distributed low-communication training of language models. *CoRR*, abs/2311.08105, 2023. doi: 10.48550/ RXIV.2311.08105. URL `https://doi.org/10.48550/arXiv.2311.08105`.

[20] Bo Liu, Rachita Chhaparia, rthur Douillard, Satyen Kale, ndrei . Rusu, Jiajun Shen, rthur Szlam, and Marc' urelio Ranzato. synchronous local-sgd training for language modeling. *CoRR*, abs/2401.09135, 2024. doi: 10.48550/ RXIV.2401.09135. URL `https://doi.org/10.48550/arXiv.2401.09135`.

[21] rthur Douillard, Qixuan Feng, ndrei . Rusu, dhiguna Kuncoro, Yani Donchev, Rachita Chhaparia, Ionel Gog, Marc' urelio Ranzato, Jiajun Shen, and rthur Szlam. Dipaco: Distributed path composition. *CoRR*, abs/2403.10616, 2024. doi: 10.48550/ RXIV.2403.10616. URL `https://doi.org/10.48550/arXiv.2403.10616`.

[22] Nicholas D. Lane, Lorenzo Sani, and lex Iacob. Introducing FlowerLLM, 2024. URL `https://flower.ai/blog/2024 03 14 introducing flowerllm/`.

[23] Daniel J. Beutel, Taner Topal, khil Mathur, Xinchi Qiu, Titouan Parcollet, and Nicholas D. Lane. Flower: friendly federated learning research framework. *CoRR*, abs/2007.14390, 2020. URL `https://arxiv.org/abs/2007.14390`.

[24] Lorenzo Sani, Pedro Porto Buarque de Gusmão, lex Iacob, Wanru Zhao, Xinchi Qiu, Yan Gao, Javier Fernandez-Marques, and Nicholas Donald Lane. Pollen: High-throughput simulation of federated learning via resource-aware client placement, 2024.

[25] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, rvind Neelakantan, Pranav Shyam, Girish Sastry, manda skell, Sandhini garwal, riel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, ditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, lec Radford, Ilya Sutskever, and Dario modei. Language models are few-shot learners, 2020.

[26] Open I. GPT-4 technical report. *CoRR*, abs/2303.08774, 2023. doi: 10.48550/ RXIV.2303.08774. URL `https://doi.org/10.48550/arXiv.2303.08774`.

[27] Rohan nil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste layrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, ndrew M. Dai, nja Hauth, Katie Millican, David Silver, Slav Petrov, Melvin Johnson, Ioannis ntonoglou, Julian Schrittwieser, melia Glaese, Jilin Chen, Emily Pitler, Timothy P. Lillicrap, ngeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul Ronald Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem youb, Megha Goel, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, naïs White, nders ndreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, and et al. Gemini: family of highly capable multimodal models. *CoRR*, abs/2312.11805, 2023. doi: 10.48550/ RXIV.2312.11805. URL `https://doi.org/10.48550/arXiv.2312.11805`.

[28] Teven Le Scao, ngela Fan, Christopher kiki, Ellie Pavlick, Suzana Ilic, Daniel Hesslow, Roman Castagné, lexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, lexander M. Rush, Stella Biderman, lbert Webson, Pawan Sasanka mmanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, lbert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, ngelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, aron Gokaslan, di Simhi, itor Soroa, lham Fikri ji, mit lfassy, nna Rogers, riel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa delani, and et al. BLOOM: 176b-parameter open-access multilingual language model. *CoRR*, abs/2211.05100, 2022. doi: 10.48550/ RXIV.2211.05100. URL `https://doi.org/10.48550/arXiv.2211.05100`.

[29] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie- nne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal zhar, urelien Rodriguez, rmand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023.

[30] Hugo Touvron, Louis Martin, Kevin Stone, Peter lbert, mjad lmahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, nthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, rtem Korenev, Punit Singh Koura, Marie- nne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, ndrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, lan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, dina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, ngela Fan, Melanie Kambadur, Sharan Narang,

urelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

[31] Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, lessandro Cappelli, Hamza lobeidli, Baptiste Pannier, Ebtesam lmazrouei, and Julien Launay. The refinedweb dataset for falcon llm: Outperforming curated corpora with web data, and web data only, 2023.

[32] Sam McCandlish, Jared Kaplan, Dario modei, and Open I Dota Team. n empirical model of large-batch training. *CoRR*, abs/1812.06162, 2018. URL `http://arxiv.org/abs/1812.06162`.

[33] Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemyslaw Debiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Christopher Hesse, Rafal Józefowicz, Scott Gray, Catherine Olsson, Jakub Pachocki, Michael Petrov, Henrique Pondé de Oliveira Pinto, Jonathan Raiman, Tim Salimans, Jeremy Schlatter, Jonas Schneider, Szymon Sidor, Ilya Sutskever, Jie Tang, Filip Wolski, and Susan Zhang. Dota 2 with large scale deep reinforcement learning. *CoRR*, abs/1912.06680, 2019. URL `http://arxiv.org/abs/1912.06680`.

[34] Shen Li, Yanli Zhao, Rohan Varma, Omkar Salpekar, Pieter Noordhuis, Teng Li, dam Paszke, Jeff Smith, Brian Vaughan, Pritam Damania, and Soumith Chintala. Pytorch distributed: Experiences on accelerating data parallel training. *Proc. VLDB Endow.*, 13(12): 3005–3018, aug 2020. ISSN 2150-8097. doi: 10.14778/3415478.3415530. URL `https://doi.org/10.14778/3415478.3415530`.

[35] lexander Sergeev and Mike Del Balso. Horovod: fast and easy distributed deep learning in tensorflow. *CoRR*, abs/1802.05799, 2018. URL `http://arxiv.org/abs/1802.05799`.

[36] Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *CoRR*, abs/1909.08053, 2019. URL `http://arxiv.org/abs/1909.08053`.

[37] Noam Shazeer, Youlong Cheng, Niki Parmar, Dustin Tran, shish Vaswani, Penporn Koanantakool, Peter Hawkins, HyoukJoong Lee, Mingsheng Hong, Cliff Young, Ryan Sepassi, and Blake . Hechtman. Mesh-tensorflow: Deep learning for supercomputers. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *dvances in Neural Information Processing Systems 31: nnual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 10435–10444, 2018. URL `https://proceedings.neurips.cc/paper/2018/hash/3a37abdeefe1dab1b30f7c5c7e581b93 Abstract.html`.

[38] Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: memory optimizations toward training trillion parameter models. In Christine Cuicchi, Irene Qualters, and William T. Kramer, editors, *Proceedings of the International Conference for High Performance Computing, Networking, Storage and nalysis, SC 2020, Virtual Event / tlanta, Georgia, US , November 9-19, 2020*, page 20. IEEE/ CM, 2020. doi: 10.1109/SC41405.2020.00024. URL `https://doi.org/10.1109/SC41405.2020.00024`.

[39] Yanli Zhao, ndrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, lban Desmaison, Can Balioglu, Pritam Damania, Bernard Nguyen, Geeta Chauhan, Yuchen Hao, jit Mathews, and Shen Li. Pytorch FSDP: experiences on scaling fully sharded data parallel. *Proc. VLDB Endow.*, 16(12):3848–3860, 2023. doi: 10.14778/3611540.3611569. URL `https://www.vldb.org/pvldb/vol16/p3848 huang.pdf`.

[40] Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *CoRR*, abs/1604.06174, 2016. URL `http://arxiv.org/abs/1604.06174`.

[41] Jie Ren, Samyam Rajbhandari, Reza Yazdani minabadi, Olatunji Ruwase, Shuangyan Yang, Minjia Zhang, Dong Li, and Yuxiong He. Zero-offload: Democratizing billion-scale model training. In Irina Calciu and Geoff Kuenning, editors, *2021 USENIX nnual Technical Conference, USENIX TC 2021, July 14-16, 2021*, pages 551–564. USENIX ssociation, 2021. URL `https://www.usenix.org/conference/atc21/presentation/ren jie`.

[42] Yingzhou Lu, Huazheng Wang, and Wenqi Wei. Machine learning for synthetic data generation: a review. *CoRR*, abs/2302.04062, 2023. doi: 10.48550/ RXIV.2302.04062. URL `https://doi.org/10.48550/arXiv.2302.04062`.

[43] La Javaness. `https://lajavaness.medium.com/llm large language model cost analysis d5022bb43e9e`, 2023.

[44] Shuming Ma, Hongyu Wang, Lingxiao Ma, Lei Wang, Wenhui Wang, Shaohan Huang, Li Dong, Ruiping Wang, Jilong Xue, and Furu Wei. The era of 1-bit llms: ll large language models are in 1.58 bits. *CoRR*, abs/2402.17764, 2024. doi: 10.48550/ RXIV.2402.17764. URL `https://doi.org/10.48550/arXiv.2402.17764`.

[45] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan llen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, pril 25-29, 2022*. OpenReview.net, 2022. URL `https://openreview.net/forum?id=nZeVKeeFYf9`.

[46] lexander Borzunov, Max Ryabinin, rtem Chumachenko, Dmitry Baranchuk, Tim Dettmers, Younes Belkada, Pavel Samygin, and Colin Raffel. Distributed inference and fine-tuning of large language models over the internet, 2023.

[47] Zhouyuan Huo, Qian Yang, Bin Gu, Lawrence Carin, and Heng Huang. Faster on-device training using new federated momentum algorithm. *CoRR*, abs/2002.02090, 2020. URL `https://arxiv.org/abs/2002.02090`.

[48] H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=BJ0hF1Z0b`.

[49] K. . Bonawitz, Vladimir Ivanov, Ben Kreuter, ntonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, aron Segal, and Karn Seth. Practical secure aggregation for federated learning on user-held data. In *NIPS Workshop on Private Multi-Party Machine Learning*, 2016. URL `https://arxiv.org/abs/1611.04482`.

[50] Peter Kairouz, H. Brendan McMahan, Brendan vent, urélien Bellet, Mehdi Bennis, rjun Nitin Bhagoji, Kallista . Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, drià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaïd Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, leksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, yfer Özgür, Rasmus Pagh, Hang Qi, Daniel Ramage, Ramesh Raskar, Mariana Raykova, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, nanda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. dvances and open problems in federated learning. *Found. Trends Mach. Learn.*, 14(1-2):1–210, 2021. doi: 10.1561/2200000083. URL `https://doi.org/10.1561/2200000083`.

[51] Kallista . Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, lex Ingerman, Vladimir Ivanov, Chloé Kiddon, Jakub Konečný, Stefano Mazzocchi, Brendan McMahan, Timon Van Overveldt, David Petrou, Daniel Ramage, and Jason Roselander. Towards federated learning at scale: System design. In meet Talwalkar, Virginia Smith, and Matei Zaharia, editors, *Proceedings of Machine Learning and Systems 2019, MLSys 2019, Stanford, C , US , March 31 - pril 2, 2019*. mlsys.org, 2019. URL `https://proceedings.mlsys.org/book/271.pdf`.

[52] Jose Javier Gonzalez Ortiz, Jonathan Frankle, Mike Rabbat, ri S. Morcos, and Nicolas Ballas. Trade-offs of local SGD at scale: n empirical study. *CoRR*, abs/2110.08133, 2021. URL `https://arxiv.org/abs/2110.08133`.

[53] grin Hilmkil, Sebastian Callh, Matteo Barbieri, Leon René Sütfeld, Edvin Listo Zec, and Olof Mogren. Scaling federated learning for fine-tuning of large language models. In Elisabeth Métais, Farid Meziane, Helmut Horacek, and Epaminondas Kapetanios, editors, *Natural Language Processing and Information Systems - 26th International Conference on plications of Natural Language to Information Systems, NLDB 2021, Saarbrücken, Germany, June 23-25, 2021, Proceedings*, volume 12801 of *Lecture Notes in Computer Science*, pages 15–23. Springer, 2021. doi: 10.1007/978-3-030-80599-9_2. URL `https://doi.org/10.1007/978 3 030 80599 9_2`.

[54] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. LBERT: lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, ddis baba, Ethiopia, pril 26-30, 2020*. OpenReview.net, 2020. URL `https://openreview.net/forum?id=H1eA7AEtvS`.

[55] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North merican Chapter of the ssociation for Computational Linguistics: Human Language Technologies, N CL-HLT 2019, Minneapolis, MN, US , June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. ssociation for Computational Linguistics, 2019. doi: 10.18653/V1/N19-1423. URL `https://doi.org/10.18653/v1/n19 1423`.

[56] Pascal Riedel, Manfred Reichert, Reinhold von Schwerin, lexander Hafner, Daniel Schaudt, and Gaurav Singh. Performance analysis of federated learning algorithms for multilingual protest news detection using pre-trained distilbert and BERT. *IEEE ccess*, 11:134009–134022, 2023. doi: 10.1109/ CCESS.2023.3334910. URL `https://doi.org/10.1109/ACCESS.2023.3334910`.

[57] Boxin Wang, Jacky Yibo Zhang, Yuan Cao, Bo Li, H. Brendan McMahan, Sewoong Oh, Zheng Xu, and Manzil Zaheer. Can public large language models help private cross-device federated learning? *CoRR*, abs/2305.12132, 2023. doi: 10.48550/ RXIV.2305.12132. URL `https://doi.org/10.48550/arXiv.2305.12132`.

[58] Orion Weller, Marc Marone, Vladimir Braverman, Dawn J. Lawrie, and Benjamin Van Durme. Pretrained models for multilingual federated learning. In Marine Carpuat, Marie-Catherine de Marneffe, and Iván Vladimir Meza Ruíz, editors, *Proceedings of the 2022 Conference of the North merican Chapter of the ssociation for Computational Linguistics: Human Language Technologies, N CL 2022, Seattle, W , United States, July 10-15, 2022*, pages 1413–1421. ssociation for Computational Linguistics, 2022. doi: 10.18653/V1/2022.N CL-M IN.101. URL `https://doi.org/10.18653/v1/2022.naacl main.101`.

[59] Jianyi Zhang, Saeed Vahidian, Martin Kuo, Chunyuan Li, Ruiyi Zhang, Guoyin Wang, and Yiran Chen. Towards building the federated GPT: federated instruction tuning. *CoRR*, abs/2305.05644, 2023. doi: 10.48550/ RXIV.2305.05644. URL `https://doi.org/10.48550/arXiv.2305.05644`.

[60] Tao Fan, Yan Kang, Guoqiang Ma, Weijing Chen, Wenbin Wei, Lixin Fan, and Qiang Yang. F TE-LLM: industrial grade federated learning framework for large language models. *CoRR*, abs/2310.10049, 2023. doi: 10.48550/ RXIV.2310.10049. URL `https://doi.org/10.48550/arXiv.2310.10049`.

[61] Weirui Kuang, Bingchen Qian, Zitao Li, Daoyuan Chen, Dawei Gao, Xuchen Pan, Yuexiang Xie, Yaliang Li, Bolin Ding, and Jingren Zhou. Federatedscope-llm: comprehensive package for fine-tuning large language models in federated learning. *CoRR*, abs/2309.00363, 2023. doi: 10.48550/ RXIV.2309.00363. URL `https://doi.org/10.48550/arXiv.2309.00363`.

[62] Jingang Jiang, Xiangyang Liu, and Chenyou Fan. Low-parameter federated learning with large language models. *CoRR*, abs/2307.13896, 2023. doi: 10.48550/ RXIV.2307.13896. URL `https://doi.org/10.48550/arXiv.2307.13896`.

[63] Shubham Malaviya, Manish Shukla, and Sachin Lodha. Reducing communication overhead in federated learning for pre-trained language models using parameter-efficient finetuning. In Sarath Chandar, Razvan Pascanu, Hanie Sedghi, and Doina Precup, editors, *Conference on Lifelong Learning gents, 22-25 ugust 2023, McGill University, Montréal, Québec, Canada*, volume 232 of *Proceedings of Machine Learning Research*, pages 456–469. PMLR, 2023. URL `https://proceedings.mlr.press/v232/malaviya23a.html`.

[64] Mingbin Xu, Congzheng Song, Ye Tian, Neha grawal, Filip Granqvist, Rogier C. van Dalen, Xiao Zhang, rturo rgueta, Shiyi Han, Yaqiao Deng, Leo Liu, nmol Walia, and lex Jin. Training large-vocabulary neural language models by private federated learning for resource-constrained devices. In *IEEE International Conference on coustics, Speech and Signal Processing IC SSP 2023, Rhodes Island, Greece, June 4-10, 2023*, pages 1–5. IEEE, 2023. doi: 10.1109/IC SSP49357.2023.10096570. URL `https://doi.org/10.1109/ICASSP49357.2023.10096570`.

[65] Mengwei Xu, Dongqi Cai, Yaozong Wu, Xiang Li, and Shangguang Wang. Fwdllm: Efficient fedllm using forward gradient, 2024.

[66] Sara Babakniya, hmed Roushdy Elkordy, Yahya H. Ezzeldin, Qingfeng Liu, Kee-Bong Song, Mostafa El-Khamy, and Salman vestimehr. Slora: Federated parameter efficient fine-tuning of language models. *CoRR*, abs/2308.06522, 2023. doi: 10.48550/ RXIV.2308.06522. URL `https://doi.org/10.48550/arXiv.2308.06522`.

[67] Yeachan Kim, Junho Kim, Wing-Lam Mok, Jun-Hyung Park, and SangKeun Lee. Client-customized adaptation for parameter-efficient federated learning. In nna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the ssociation for Computational Linguistics: CL 2023, Toronto, Canada, July 9-14, 2023*, pages 1159–1172. ssociation for Computational Linguistics, 2023. doi: 10.18653/V1/2023.FINDINGS- CL.75. URL `https://doi.org/10.18653/v1/2023.findings acl.75`.

[68] Brian Lester, Rami l-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 3045–3059. ssociation for Computational Linguistics, 2021. doi: 10.18653/V1/2021.EMNLP-M IN.243. URL `https://doi.org/10.18653/v1/2021.emnlp main.243`.

[69] Haodong Zhao, Wei Du, Fangqi Li, Peixuan Li, and Gongshen Liu. Fedprompt: Communication-efficient and privacy-preserving prompt tuning in federated learning. In *IEEE International Conference on coustics, Speech and Signal Processing IC SSP 2023, Rhodes Island, Greece, June 4-10, 2023*, pages 1–5. IEEE, 2023. doi: 10.1109/IC SSP49357.2023.10095356. URL `https://doi.org/10.1109/ICASSP49357.2023.10095356`.

[70] Tianshi Che, Ji Liu, Yang Zhou, Jiaxiang Ren, Jiwen Zhou, Victor S. Sheng, Huaiyu Dai, and Dejing Dou. Federated learning of large language models with parameter-efficient prompt tuning and adaptive optimization. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 7871–7888. ssociation for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-M IN.488. URL `https://doi.org/10.18653/v1/2023.emnlp main.488`.

[71] Wanru Zhao, Royson Lee, Yihong Chen, Xinchi Qiu, Yan Gao, Hongxiang Fan, and Nicholas Donald Lane. Breaking physical and linguistic borders: Multilingual federated prompt tuning for low-resource languages. In *The Twelfth International Conference on Learning Representations*, 2024.

[72] Sahil Patel and Stephanie Palazzolo. Open I offers publishers as little as $1 million a year — the information, Jan 2024. URL `https://www.theinformation.com/articles/openai offers publishers as little as 1 million a year`.

[73] lexandre Magueresse, Vincent Carles, and Evan Heetderks. Low-resource languages: review of past work and future challenges. *CoRR*, abs/2006.07264, 2020. URL `https://arxiv.org/abs/2006.07264`.

[74] Surangika Ranathunga, En-Shiun nnie Lee, Marjana Prifti Skenduli, Ravi Shekhar, Mehreen lam, and Rishemjit Kaur. Neural machine translation for low-resource languages: survey. *CM Comput. Surv.*, 55(11):229:1–229:37, 2023. doi: 10.1145/3567592. URL `https://doi.org/10.1145/3567592`.

[75] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, L , US , May 6-9, 2019*. OpenReview.net, 2019. URL `https://openreview.net/forum?id=Bkg6RiCqY7`.

[76] Colin Raffel, Noam Shazeer, dam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020. URL `http://jmlr.org/papers/v21/20 074.html`.