FedSpaLLM: Federated Pruning of Large Language Models

Guangji Bai[1,2*] **Yijiang Li**[1] **Zilinghan Li**[1] **Liang Zhao**[2] **Kibaek Kim**[1†]
[1]Argonne National Laboratory [2]Emory University
`{guangji.bai,liang.zhao}@emory.edu`
`zl52@illinois.edu`, `{yijiang.li,kimk}@anl.gov`

Abstract

Large Language Models (LLMs) achieve state-of-the-art performance but are challenging to deploy due to their high computational and storage demands. Pruning can reduce model size, yet existing methods assume public access to calibration data, which is impractical for privacy-sensitive applications. To address the challenge of pruning LLMs in privacy-preserving settings, we propose **FedSpaLLM**, the first federated learning framework designed specifically for pruning LLMs. FedSpaLLM enables clients to locally prune their models based on private data while accounting for system heterogeneity and maintaining communication efficiency. Our framework introduces several key innovations: (1) a novel ℓ_0-norm aggregation function that ensures only non-zero weights are averaged across clients, preserving important model parameters; (2) an adaptive mask expansion technique that meets global sparsity targets while accommodating client-specific pruning decisions; and (3) a layer sampling strategy that reduces communication overhead and personalizes the pruning process based on client resources. Extensive experiments show that FedSpaLLM improves pruning performance in diverse federated settings. The source code can be found at `https://github.com/BaiTheBest/FedSpaLLM`.

1 Introduction

Large Language Models (LLMs) such as GPT (OpenAI, 2023) and LlaMA (Touvron et al., 2023) have recently transformed the field of Natural Language Processing (NLP) due to their ability to perform exceptionally well across a variety of complex language benchmarks. However, the increasing scale of these models, which can contain billions of parameters, also brings significant computational and storage costs. The high memory and inference costs make it challenging to deploy LLMs in real-world applications where resources are constrained (Bai et al., 2024a). Consequently, there has been an increasing interest in *model compression* techniques such as pruning, quantization, and knowledge distillation, which aim to reduce the computational load while maintaining model performance (Zhu et al., 2023). Among these techniques, *pruning* has emerged as a highly effective approach for reducing the size and complexity of LLMs by introducing sparsity into the models.

Despite the recent success of LLM pruning methods, existing approaches predominantly assume that the calibration data used for pruning is publicly available (Frantar and Alistarh, 2023; Sun et al., 2023). However, in many real-world scenarios, especially when dealing with sensitive applications like medical agents or financial systems, the data used for pruning might be private and cannot be shared openly (Ren et al., 2024). On the other hand, Federated Learning (FL), a distributed machine learning technique that enables multiple clients to collaboratively train models without sharing their private data, has gained significant popularity in traditional machine learning (Zhang et al., 2021). However, most works on LLMs in FL settings have focused on fine-tuning. Due to the intrinsic differences between fine-tuning and pruning, existing FL-based fine-tuning methods cannot handle the problem of pruning LLMs with private data.

To address the challenges posed by pruning LLMs in federated settings, where private data cannot be shared and heterogeneity exists among clients, we propose a novel method called **FedSpaLLM** (Federated Sparse LLM). FedSpaLLM is the first framework that allows pruning LLMs under a federated learning setting with resource heterogeneity. Our method allows each client to prune its local model based on its data while maintaining privacy and accommodating diverse computational resources.

[*]This work was done during Guangji Bai's internship at Argonne National Laboratory.
[†]Corresponding Author

Our contributions are summarized as follows:

- **Federated pruning framework for LLMs**: We present the first framework for pruning LLMs in FL, allowing collaborative pruning with private local data.
- **ℓ_0-norm aggregation**: We introduce a novel aggregation function that preserves important weights by averaging only non-zero elements across client models.
- **Adaptive mask expansion**: We propose a mask expansion technique to meet global sparsity targets while accounting for client-specific pruning.
- **Layer sampling**: We develop a resource-aware layer sampling strategy, enabling personalized pruning and reducing communication costs.
- **Extensive evaluation**: We conduct comprehensive experiments, showing that FedSpaLLM improves both pruning efficiency and model performance in heterogeneous federated environments.

2 Related Work

Pruning of LLMs. *Pruning* regained prominence in the late 2010s for reducing inference costs (Han et al., 2015). LLM pruning can be categorized into *structured* and *unstructured* pruning.

Unstructured pruning removes individual parameters without regard to model structure, often using thresholds to nullify smaller weights. SparseGPT (Frantar and Alistarh, 2023) achieves up to 60% parameter reduction in LLMs with minimal performance loss. Wanda (Sun et al., 2023) introduces a pruning criterion based on both weight magnitude and activations, particularly effective in linear layers. DynaTran (Tuli and Jha, 2023) dynamically prunes activations at runtime, enhanced by a custom ASIC architecture.

Structured pruning removes groups of parameters such as filters or attention heads. LLM-Pruner (Ma et al., 2023) combines first-order data and Hessian information for structured pruning, while LoSparse (Li et al., 2023) uses low-rank and sparse approximations to balance pruning and model expressiveness. Structured pruning of hidden dimensions, as shown by (Tao et al., 2023), extends to embeddings and attention heads. ZipLM (Kurtic et al., 2023) optimizes structured compression for accuracy and hardware efficiency.

Federated Learning with LLMs. FL on LLMs has primarily focused on LLM *fine-tuning* and has gained attention for enabling private and efficient model updates. FedPrompt (Zhao et al., 2023) introduces prompt-tuning in FL, reducing communication costs by updating only soft prompts. Fang et al. (2024) and Li et al. (2024b) leverage low-rank adapters for parameter-efficient fine-tuning, improving training speed and accuracy. FedNLP (Lin et al., 2022) provides a benchmarking framework for evaluating FL methods on NLP tasks. FedAdapter (Cai et al., 2023) uses adapters to accelerate model convergence in federated settings. FeDeRA (Yan et al., 2024) employs singular value decomposition to further improve LoRA-based fine-tuning efficiency. C2A (Kim et al., 2023) introduces a hypernetwork-based framework for generating client-specific adapters to handle client heterogeneity. FedBPT (Sun et al., 2024) enables efficient prompt-tuning with a gradient-free approach, reducing memory and communication costs. PrE-Text (Hou et al., 2024) generates differentially private synthetic data to enable central training, reducing on-device computation.

Federated Pruning on DNNs. Model pruning in FL improves efficiency by reducing communication and computation costs. FedP3 (Yi et al., 2024) and HeteroFL (Diao et al., 2021) address client model heterogeneity, enabling smaller, personalized models. FedTiny (Huang et al., 2023) and PruneFL (Jiang et al., 2022) implement progressive pruning to fit models within resource constraints. FedPrune (Munir et al., 2021) improves performance by pruning global models based on client capabilities, while Complement Sparsification (Jiang and Borcea, 2023) reduces communication overhead using sparsity techniques. However, all the work above only applies to smaller DNNs, and cannot trivially scale to massive LLMs.

Federated Distillation. In addition to model pruning, knowledge distillation (KD) has been explored as a resource-efficient approach to training NNs in edge environments (Li et al., 2024a). In this paradigm, a smaller proxy model is used to facilitate learning, either by mimicking the behavior of a larger model or by aggregating knowledge from multiple clients. For instance, FedBiOT (Wu et al., 2024) introduces a method for locally fine-tuning LLMs without requiring full-model transmission, allowing clients to distill and learn from a proxy model. Similarly, DaFKD (Wang et al., 2023), a domain-aware FL knowledge distillation framework, incorporates domain-specific adaptations to improve the generalization of models across heterogeneous clients.

Despite the merits of federated distillation, our proposed approach offers several advantages. Proxy models and FL distillation methods require training additional models, increasing computational overhead. Additionally, they assume an accessible public dataset for generating the soft labels, which usually does not hold in practice. Our framework directly prunes the global LLM without auxiliary models and any additional public data. It achieves target sparsity across heterogeneous clients while maintaining privacy through local pruning and aggregation.

3 Preliminaries

3.1 Pruning of LLMs

In this work, we focus on *unstructured* pruning, (Frantar and Alistarh, 2023; Sun et al., 2023; Bai et al., 2024b) which typically utilizes local pruning. Local pruning circumvents the memory issue mentioned above by dividing the full model compression into sub-problems for each layer and constructing a *local loss* to measure the ℓ_2-error between the outputs of the uncompressed and compressed layers. Hence, the local pruning can be formulated by:

$$\min_{\mathbf{M}_\ell, \widehat{\mathbf{W}}_\ell} \|\mathbf{W}_\ell \cdot \mathbf{X}_\ell - (\mathbf{M}_\ell \odot \widehat{\mathbf{W}}_\ell) \cdot \mathbf{X}_\ell\|_2^2, \quad (1)$$

where $\odot$ denotes element-wise multiplication. Although smaller than the global pruning, the local pruning still needs to optimize both the mask $\mathbf{M}_\ell$ and the remaining weights $\widehat{\mathbf{W}}_\ell$ and thus remains NP-hard. Therefore, exactly solving it for larger layers is unrealistic, leading all existing methods to resort to approximations.

Mask Selection & Weight Reconstruction. A particularly popular approach is to separate the problem into *mask selection* and *weight reconstruction* (Hubara et al., 2021; Kwon et al., 2022). Concretely, this means first choosing a pruning mask $\mathbf{M}$ according to some salient criterion, like the weight magnitude (Zhu and Gupta, 2017), and then optimizing the remaining unpruned weights while keeping the mask unchanged. Importantly, once the mask is fixed, Eq. 1 turns into a *linear regression* problem that can be easily optimized.

3.2 Background of Federated Learning

FL is a distributed machine learning paradigm designed to enable collaborative training of a single global model without exchanging private data between clients (Gu et al., 2023). In this setup, multiple clients, each with their own local dataset, train their local models independently. The central server coordinates the process by sending the global model to a selected group of clients. These clients then perform local optimization using their respective datasets. After local training, the updated models are sent back to the server, where they are aggregated to update the global model.

In each communication round, the server aggregates the updates from multiple clients to refine the global model. Mathematically, let $\mathcal{D}_i$ denote the local dataset of the i-th client, and θ represent the global model parameters. The process of updating the global model is given by:

$$\tilde{\theta} \in \arg\min \sum_{i=1}^{K} \alpha_i \cdot \mathcal{L}(\mathcal{D}_i; \theta), \quad (2)$$

where $\mathcal{L}(\mathcal{D}_i; \theta)$ is the local objective function computed over the dataset $\mathcal{D}_i$ with the model parameters θ. The factor α_i is typically proportional to the size of client i's dataset, i.e., $\alpha_i = \frac{|\mathcal{D}_i|}{\sum_i |\mathcal{D}_i|}$, and K denotes the total number of participating clients.

There exist various methods for aggregating client updates, with FedAvg (McMahan et al., 2017) being the most widely adopted due to its simplicity and effectiveness in a wide range of FL applications. This method aggregates the client updates weighted by their respective dataset sizes, providing a robust way to train models in data-sensitive environments.

4 Proposed Method

In this section, we present FedSpaLLM, our novel framework for federated pruning of large language models (LLMs). The proposed method is designed to address the challenges of pruning in federated learning settings, specifically targeting communication efficiency and system heterogeneity across clients. We first formulate the problem of pruning LLMs in an FL setup and introduce a specialized aggregation function based on the ℓ_0-norm to handle sparse model updates. Next, we propose an adaptive mask expansion technique to ensure that the global model meets the target sparsity, even when clients generate diverse pruning masks. Finally, we introduce a layer sampling strategy that allows clients to prune subsets of model layers based on their computational resources, enabling personalized pruning and reducing communication costs.

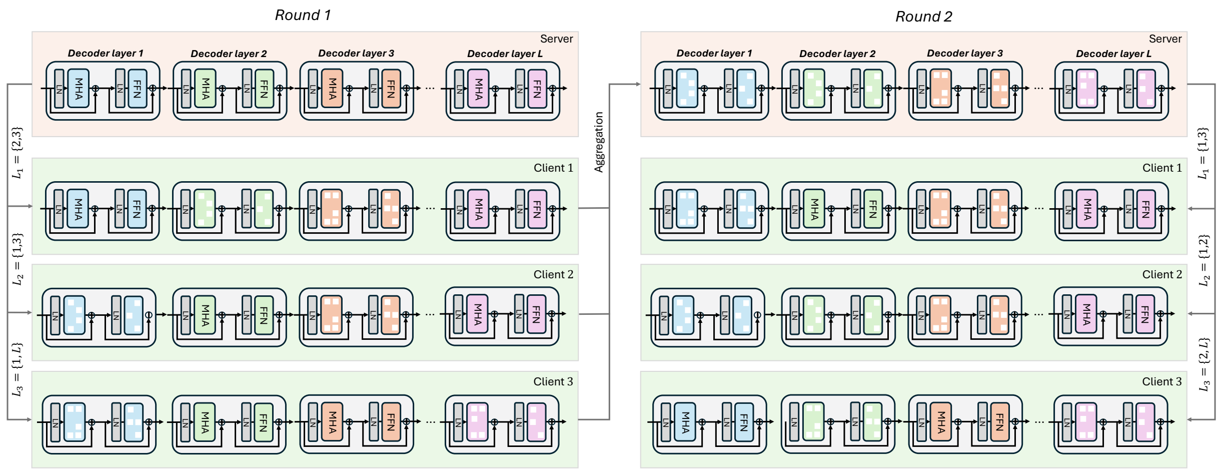


Figure 1: Visualization of the proposed **FedSpaLLM** framework. Instead of transmitting the full model at each communication round, the server samples a subset of layers based on each client's computational resources. Clients prune only the sampled layers and retain the rest from their cached pre-trained dense model. After local pruning, clients only send their pruned layers to the server, which aggregates the pruned layers using a novel ℓ_0-norm aggregation function that averages only the non-zero parameters. This approach ensures that important weights are preserved while reducing communication overhead. The layer sampling strategy enables personalized pruning tailored to client heterogeneity, reducing resource usage without compromising overall model performance.

Figure 1 provides a visual overview of the FedSpaLLM framework, illustrating how clients and the central server interact during the pruning process.

4.1 Problem Formulation

We present the first formulation of pruning LLMs under the FL setting. In this scenario, multiple clients collaboratively prune a global LLM while ensuring that their local data remains private. Let W_g denote the global model parameters, where $W_g \in \mathbb{R}^d$, and each client i holds its own local dataset $\mathcal{D}_i = \{X_i, Y_i\}$ for training and pruning purposes.

During pruning, each client applies a binary pruning mask, $M_i \in \{0,1\}^d$, to its local model W_i. This mask determines which weights are retained ($M_i = 1$) and which are pruned ($M_i = 0$). The objective is to prune the global model while ensuring that the pruned models on each client still perform effectively on their respective local datasets. Importantly, our formulation allows for model heterogeneity or personalization, meaning that each client can have its own set of model parameters, W_i, which differs from the global model, W_g. This flexibility contrasts with the traditional FL setting, where all clients share the same global model.

The overall objective of federated pruning is to minimize the weighted sum of local losses across clients. Each client i minimizes its local loss function $\mathcal{L}_i$ based on the pruned model $M_i \odot W_i$. Formally, this can be written as:

$$\min_{W_1,\ldots,W_N,M_1,\ldots,M_N} \sum\nolimits_{i=1}^{N} \alpha_i \cdot \mathcal{L}_i(M_i \odot W_i), \quad (3)$$

subject to the global model W_g being an aggregation of the locally pruned models:

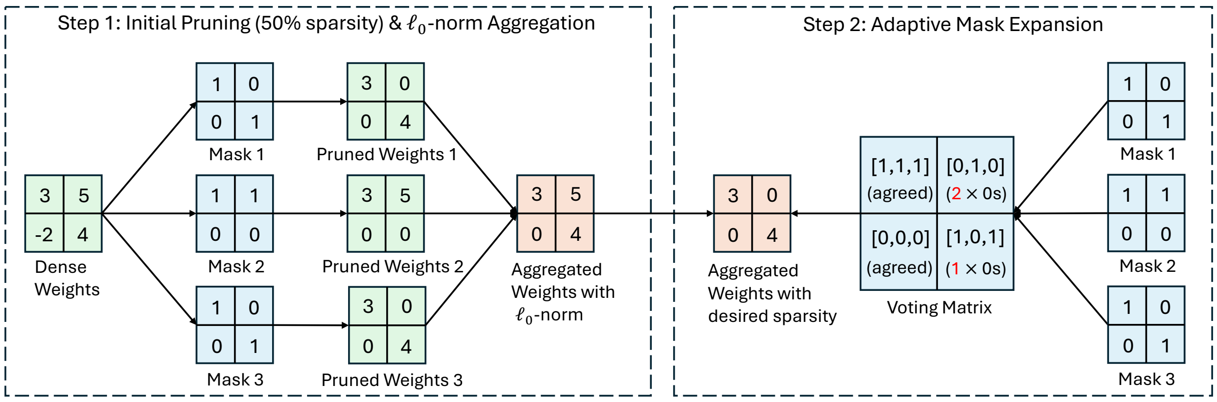
$$W_g = \Omega\Big(M_1 \odot W_1, \ldots, M_N \odot W_N\Big), \quad (4)$$

where Ω denotes the aggregation function that combines the pruned models from all clients to update the global model.

Our formulation is general and supports model heterogeneity, where each client may have its own model parameters, W_i, as opposed to the vanilla FL setting sharing a common global model. This flexibility is crucial for accommodating diverse client environments, making our method applicable to a wide range of real-world federated pruning scenarios.

4.2 Aggregation Function

ℓ_0-norm Aggregator. In traditional FL, the model weights from different clients are aggregated using a simple mean average or a similar function after each round of local updates. However, in our case, the parameters are sparsified through pruning, resulting in binary pruning masks. Due to the discrete nature of these masks, using a vanilla average function is not suitable. Specifically, when aggregating binary masks, dividing by the total count of clients



Step 1: Initial Pruning (50% sparsity) & ℓ_0-norm Aggregation
Dense Weights
Mask 1
Mask 2
Mask 3
Pruned Weights 1
Pruned Weights 2
Pruned Weights 3
Aggregated Weights with ℓ_0-norm
Step 2: Adaptive Mask Expansion
Aggregated Weights with desired sparsity
[1,1,1] (agreed)
[0,1,0] (2 × 0s)
[0,0,0] (agreed)
[1,0,1] (1 × 0s)
Voting Matrix

Figure 2: Visualization of the proposed **Aggregation Function** of FedSpaLLM to handle heterogeneous sparsified parameters. After clients prune their local models, the server aggregates the pruned layers by using the ℓ_0-norm aggregation function. This method avoids diluting the effect of unpruned weights by excluding zeros from the averaging process, thus preserving important parameters. To achieve the target global sparsity, an adaptive mask expansion is applied: the server counts the number of times each weight has been pruned across clients and uses this information to expand the pruning mask. The mask expansion prioritizes pruning weights that are most commonly pruned across clients, balancing individual client pruning decisions with the global sparsity goal.

would incorrectly include zeros (pruned positions) in the calculation, causing the averaged value to approach zero. This leads to an undesirable convergence, where the model parameters tend to vanish as the denominator grows larger than it should be.

To address this, we propose an aggregation function based on the ℓ_0-norm. Instead of averaging over all elements, we compute the average only over the non-zero elements (i.e., weights that are retained across clients). Mathematically, given the pruning masks $M_1, M_2, \ldots, M_N$ from N clients and the corresponding local weights $W_1, W_2, \ldots, W_N$, the aggregation is defined as:

$$W_g = \frac{\sum_{i=1}^{N} M_i \odot W_i}{\left\| \sum_{i=1}^{N} M_i \right\|_{\ell_0}}. \tag{5}$$

In this way, the aggregation only considers the non-zero elements, ensuring that the resulting global weights do not tend toward zero unless they are truly pruned across all clients. This method ensures the correct balance between sparsity and preserving important weights in the global model.

Adaptive Aggregation w/. Mask Expansion. Another challenge arises from the heterogeneity of data across clients. Even with the same target sparsity, different clients may generate different pruning masks based on their local data. If we simply averaged these masks, the result would be equivalent to taking the intersection of all local masks. In other words, only the weights pruned by all clients would be pruned in the global model. This leads to a situation where the global model's sparsity is always smaller than the target sparsity, as fewer weights are pruned overall.

To address this, we propose an *mask expansion* technique. The idea is to first apply the ℓ_0-norm aggregation described above and then expand the global pruning mask to achieve the target sparsity. Specifically, after the initial aggregation, we count the number of zeros (pruned positions) for each weight across clients. If the count is below the target sparsity, we sort the remaining weights and select the ones with the most agreement (i.e., those that are most commonly pruned) to prune further, ensuring the final sparsity matches the desired level.

Let C_j represent the number of zeros for the j-th weight across clients, where:

$$C_j = \sum_{i=1}^{N} \mathbb{I}(M_i^{(j)} = 0), \tag{6}$$

where $\mathbb{I}(\cdot)$ is the indicator function that returns 1 if $M_i^{(j)} = 0$ (the j-th weight is pruned by client i) and 0 otherwise. After counting, we sort the weights based on their C_j values and expand the pruning mask for weights that are not fully agreed upon by all clients. To achieve the target sparsity s, we select the top-k weights from the sorted list, where:

$$k = \text{ceil}(s \cdot d) - \sum_{j=1}^{d} \mathbb{I}(C_j = N), \tag{7}$$

where d is the total number of weights and s is the target sparsity. The first term ensures that the final mask achieves the desired sparsity, and the

Algorithm 1 FedSpaLLM: Federated Pruning of Large Language Models

```
Require: Global LLM model $W_g$, Number of communication rounds $T$, Number of clients $N$, Sparsity
    target $s$, Client computational resources $\{r_i\}_{i=1}^N$
Ensure: Pruned global model $W_g'$
 1: Initialize global model $W_g$
 2: for each communication round $t = 1$ to $T$ do
 3:     Server: Sample a subset of layers for each client based on computational resources
 4:     for each client $i = 1$ to $N$ do
 5:         Sample $k_i$ layers: $L_i \leftarrow \text{Sample}(L, k_i)$, where $k_i \propto r_i$
 6:         Send only the sampled layers $L_i$ to client $i$
 7:     end for
 8:     for each client $i = 1$ to $N$ in parallel do
 9:         Client: Prune the received layers $L_i$ with local data
10:         Use pre-cached dense layers for unsampled layers
11:         Generate pruning mask $M_i$ for layers $L_i$: $M_i \leftarrow \text{GeneratePruningMask}(W_g[L_i], \text{LocalData}_i)$
12:         Prune and update local model: $W_i' \leftarrow M_i \odot W_g[L_i]$
13:         Send pruned weights $W_i'[L_i]$ and mask $M_i$ back to the server
14:     end for
15:     Server: Aggregate pruned layers using $\ell_0$-norm aggregation
16:     $W_g' \leftarrow \ell_0\text{-NormAggregation}(\{W_i'[L_i], M_i\}_{i=1}^N)$
17:     Apply adaptive mask expansion to achieve target sparsity $s$
18:     Adjust global pruning mask to meet desired sparsity level
19: end for
20: return Final pruned global model $W_g'$
```

second term subtracts the number of weights that are already pruned by all clients (i.e., those where $C_j = N$).

By employing this sorting and expansion method, we ensure that the global pruning mask adheres to the desired sparsity while reflecting the commonly agreed-upon pruned positions, balancing individual client decisions and the global sparsity requirement.

4.3 Personalization with Layer Sampling

In FL for LLM pruning, communication overhead, and system heterogeneity are key challenges. The pruning of LLMs is typically done in a local manner, where the pruning of each decoder layer is independent, allowing for efficient and flexible layer sampling. This independence enables us to design a novel sampling strategy that maintains both efficiency and pruning accuracy while addressing the diverse computational capacities of clients.

Remark: Unlike traditional FL, where models are typically fine-tuned holistically, pruning in LLMs allows each decoder layer to be pruned independently. We exploit this property by sampling layers, significantly reducing communication costs without compromising the overall pruning outcome.

Layer Sampling Strategy. In each communication round, the server randomly samples a subset of layers from the LLM for each client to prune. Let $L = \{l_1, l_2, \ldots, l_m\}$ denote the set of all layers in the LLM, where m is the total number of layers. For each client i, the server selects a subset of layers $L_i \subseteq L$ to be pruned, where the number of layers sampled, $|L_i| = k_i$, is proportional to the computational capacity of client i, denoted as r_i. Formally,

$$L_i = \text{Sample}(L, k_i), \quad k_i \propto r_i. \tag{8}$$

Once client i receives the subset L_i, it performs pruning on the sampled layers while retaining the original weights of the unsampled layers. Let $W_g = \{W_1, W_2, \ldots, W_m\}$ represent the global model weights, and $M_i = \{M_1, M_2, \ldots, M_m\}$ represent the pruning masks for client i. The locally pruned model W_i' is updated as:

$$W_i' = \begin{cases} M_j \odot W_j, & \text{if } j \in L_i, \\ W_j^{\text{dense}}, & \text{if } j \notin L_i. \end{cases} \tag{9}$$

For unsampled layers $j \notin L_i$ we retain their original unpruned dense weights W_j^{dense}. This process ensures that communication costs are minimized, as we only need to communicate the sampled layers L_i, only a portion of the model, in each round.

Corollary 4.1 (Sparsity Guarantee)**.** *Let $\mathcal{S}_{global}$ denote the target global sparsity, and let $\mathcal{S}_i$ be the sparsity achieved by client i on its local model. If the layer sampling strategy ensures that all layers $\mathcal{L}$ are sampled at least once across all clients in each communication round, the aggregated global pruned model $\hat{W}_g$ will maintain the same sparsity as the target sparsity $\mathcal{S}_{global}$. Since all clients share the same target sparsity, and our ℓ_0-norm aggregation guarantees that the aggregated layers have the exact sparsity as the locally pruned layers, the sparsity of the global model will always equal the desired target. Formally:*

$$\mathcal{S}_{global} = \mathcal{S}_i, \quad \forall i, \tag{10}$$

where N is the total number of clients. Thus, the sparsity of the global model is consistent with the target across all communication rounds.

This corollary ensures that, with our sampling strategy, the global model will meet the desired global sparsity after aggregation.

Theorem 4.2 (Unbiased Estimator)**.** *Let $\hat{W}_g$ denote the global model obtained by aggregating pruned models after layer sampling, and W_g^* be the global model obtained if all layers were pruned at every client (without sampling). The model $\hat{W}_g$ is an unbiased estimator of W_g^* under the layer sampling strategy. Formally:*

$$\mathbb{E}[\hat{W}_g] = W_g^*. \tag{11}$$

This holds as long as each layer has an equal probability of being selected across clients during each communication round, ensuring that all layers are eventually represented in the final aggregated model.

Theorem 1 ensures that the global model obtained by layer sampling converges to the same result as if every client had pruned all layers. The key here is the randomness of the layer selection process: over multiple communication rounds, the expected contribution of each layer is preserved, resulting in an unbiased estimate of the fully pruned model. This guarantees that our sampling strategy does not introduce systematic errors and that, over time, the sampled model will mirror the performance of the fully pruned model.

5 Experiments

Experiments Setup. We implement our FedSpaLLM in PyTorch (Paszke et al., 2019) and use the HuggingFace Transformers library (Wolf et al., 2019) for handling models and datasets. All pruning experiments are conducted on NVIDIA A100 GPUs. For each client, we utilize SparseGPT (Frantar and Alistarh, 2023) to perform pruning. For the calibration data, we follow (Frantar and Alistarh, 2023) and use 2048-token segments, randomly chosen from the first shard of the C4 (Raffel et al., 2020) dataset. This represents generic text data crawled from the internet and ensures that our experiments remain zero-shot since no task-specific data is seen during pruning. In addition, we consider a random pruning baseline in which the model is randomly pruned to the target sparsity.

In particular, we perform experiments on the OPT model family (Zhang et al., 2022) and LlaMA-2 model family (Touvron et al., 2023) with 4, 8, and 16 clients. We consider OPT-125m, OPT-1.3b, and LlaMA-2 7b with unstructured sparsity of 50% to 80%. In each communication round, each of the clients receives a copy of the same global model from the server and each client is assumed to utilize 32 calibration samples to perform its own pruning. By evaluating models of varying sizes alongside different number of clients, we can gain a more comprehensive understanding of FedSpaLLM's performances in its scalability and robustness. In terms of metrics, we mainly focus on perplexity, which is known to be a challenging and stable metric that is well-suited for evaluating the accuracy of compression methods (Yao et al., 2022; Dettmers and Zettlemoyer, 2023), and thus measuring the performances of the compressed models. We consider the test sets of raw-WikiText2 (Merity et al., 2016) (WT-2) and PTB (Marcus et al., 1994) as well as a subset of the C4 validation data, which are all popular benchmarks in LLM compression literature (Yao et al., 2022; Park et al., 2022; Frantar and Alistarh, 2023; Sun et al., 2023).

5.1 Results and analyses

We present the perplexity results in Tables 1 to 3. In the tables, we report the random pruning baseline, denoted by “Random", the average perplexity of the client models, denoted by “standalone", and the global model, denoted by “FedSpaLLM". From the results, we see that random pruning results in

OPT-125m												
Sparsity	50%			60%			70%			80%		
Dataset	WT-2	PTB	C4	WT-2	PTB	C4	WT-2	PTB	C4	WT-2	PTB	C4
Random	1.45e4	1.36e4	1.17e4	1.86e4	1.74e4	1.62e4	2.16e4	2.09e4	1.96e4	3.22e4	3.22e4	3.03e4
Standalone	37.87	57.38	33.78	62.09	93.14	49.38	237.07	296.61	158.54	1699.62	1907.27	759.18
FedSpaLLM	37.65	57.07	33.75	61.28	92.32	48.96	226.44	287.72	152.96	1414.77	1699.16	739.97

OPT-1.3b												
Sparsity	50%			60%			70%			80%		
Dataset	WT-2	PTB	C4	WT-2	PTB	C4	WT-2	PTB	C4	WT-2	PTB	C4
Random	1.90e4	1.92e4	1.75e4	1.77e4	1.73e4	1.67e4	2.14e4	2.23e4	2.01e4	3.00e4	3.10e4	2.97e4
Standalone	18.16	26.78	20.31	23.55	35.93	24.15	62.08	97.46	52.13	1814.44	1756.18	636.56
FedSpaLLM	18.09	26.52	20.13	22.70	34.05	23.72	54.87	82.99	48.05	1653.65	1564.86	598.83

LlaMA-2 7b												
Sparsity	50%			60%			70%			80%		
Dataset	WT-2	PTB	C4	WT-2	PTB	C4	WT-2	PTB	C4	WT-2	PTB	C4
Random	8.83e4	1.20e5	1.41e5	1.98e5	2.99e5	1.92e5	5.06e4	4.98e4	4.93e4	4.99e4	4.91e4	4.97e4
Standalone	7.13	185.18	9.47	10.71	1255.83	13.52	31.60	6582.79	34.17	124.59	8398.51	110.45
FedSpaLLM	6.71	88.15	9.03	9.02	300.35	11.94	20.14	1371.44	23.28	119.21	3382.99	98.87

Table 1: Average perplexity of the client models and perplexity of the global model with 4 clients; the lower the perplexity, the better.

OPT-125m												
Sparsity	50%			60%			70%			80%		
Dataset	WT-2	PTB	C4	WT-2	PTB	C4	WT-2	PTB	C4	WT-2	PTB	C4
Random	1.24e4	1.12e4	9.93e3	1.65e4	1.52e4	1.40e4	2.11e4	2.04e4	1.86e4	3.29e4	3.31e4	3.15e4
Standalone	38.40	57.51	33.95	64.52	94.19	50.08	244.90	308.59	158.49	1655.23	1893.9	732.29
FedSpaLLM	38.01	56.66	33.65	63.00	91.50	49.10	217.54	274.06	147.58	1415.78	1611.45	665.36

OPT-1.3b												
Sparsity	50%			60%			70%			80%		
Dataset	WT-2	PTB	C4	WT-2	PTB	C4	WT-2	PTB	C4	WT-2	PTB	C4
Random	1.59e4	1.44e4	1.40e4	2.15e4	2.17e4	1.84e4	2.26e4	2.27e4	2.06e4	3.25e4	3.27e4	3.22e4
Standalone	18.51	27.18	20.29	23.99	36.53	24.54	68.70	108.93	56.28	2109.46	2134.22	733.80
FedSpaLLM	18.38	27.89	20.03	23.20	35.60	23.75	63.31	105.37	54.85	1979.77	1989.21	690.86

LlaMA-2 7b												
Sparsity	50%			60%			70%			80%		
Dataset	WT-2	PTB	C4	WT-2	PTB	C4	WT-2	PTB	C4	WT-2	PTB	C4
Random	2.62e5	1.14e5	2.92e5	6.57e4	6.11e4	6.58e4	6.15e4	6.25e4	6.17e4	4.57e4	4.76e4	4.78e4
Standalone	7.17	193.89	9.43	10.76	1126.63	13.45	30.77	8349.21	33.31	134.08	1.3e4	108.87
FedSpaLLM	6.70	73.67	8.97	9.09	175.98	12.03	21.18	1242.25	24.98	117.73	2819.17	96.44

Table 2: Average perplexity of the client models and perplexity of the global model with 8 clients; the lower the perplexity, the better.

perplexity that are orders of magnitude higher than both standalone and FedSpaLLM. As the model size increases, the performances of random pruning becomes even worse. This suggests that random pruning may have pruned weights that are crucial for maintaining model quality. Comparing standalone and FedSpaLLM, we see that across the models, datasets, and sparsity levels, FedSpaLLM consistently outperforms standalone in achieving lower perplexity. In general, as the target sparsity increases, we see more noticeable improvements in the perplexity of the global model over the client models. This is expected because as the sparsity increases, more information is required to accurately prune the model weights to maintain the model performances. In essence, each of the client models is pruned with the private calibration samples of each client while the global model benefits from the collaborative information from the communicated weights from the clients. As a result, the global model is effectively utilizing a signicantly larger calibration sample set, even though such a centralized calibration sample set is prohibited in FL setting as the client's calibration samples are private. Notably, FedSpaLLM achieves substantially lower perplexity compared to standalone in higher sparsity levels, highlighting the benefits of FedSpaLLM where the clients collaboratively contribute to the global model with much better performances while the privacy of their own data is well

OPT-125m												
Sparsity	50%			60%			70%			80%		
Dataset	WT-2	PTB	C4	WT-2	PTB	C4	WT-2	PTB	C4	WT-2	PTB	C4
Random	1.22e4	1.14e4	9.96e3	1.66e4	1.56e4	1.45e4	2.47e4	2.35e4	2.24e4	3.20e4	3.23e4	2.95e4
Standalone	38.24	57.37	33.91	67.42	98.29	51.92	264.88	326.87	171.83	1624.46	1764.94	777.79
FedSpaLLM	38.00	56.90	33.62	66.41	97.92	51.84	240.18	304.17	167.30	1389.22	1454.49	700.84

OPT-1.3b												
Sparsity	50%			60%			70%			80%		
Dataset	WT-2	PTB	C4	WT-2	PTB	C4	WT-2	PTB	C4	WT-2	PTB	C4
Random	1.80e4	1.77e4	1.53e4	1.88e4	1.87e4	1.75e4	2.32e4	2.22e4	2.14e4	2.99e4	3.02e4	2.94e4
Standalone	18.99	27.48	20.76	25.01	37.87	25.49	82.24	138.91	65.53	2472.21	2475.89	873.45
FedSpaLLM	18.20	27.30	20.52	23.93	36.36	22.72	77.51	129.40	63.62	2327.93	2361.86	838.53

LlaMA-2 7b												
Sparsity	50%			60%			70%			80%		
Dataset	WT-2	PTB	C4	WT-2	PTB	C4	WT-2	PTB	C4	WT-2	PTB	C4
Random	1.76e5	2.79e5	2.19e5	8.30e4	7.96e4	7.92e4	9.73e4	2.54e5	1.13e5	5.20e4	4.89e4	4.89e4
Standalone	7.17	188.35	9.41	10.80	969.84	13.44	31.81	6671.71	33.20	137.21	7757.43	106.87
FedSpaLLM	6.75	70.79	9.04	9.33	178.15	12.45	25.09	702.03	27.27	115.91	1791.83	94.87

Table 3: Average perplexity of the client models and perplexity of the global model with 16 clients; the lower the perplexity, the better.

maintained. Furthermore, we can see the improvements in the perplexity of the global model over the client models are particularly significant for the LlaMA-2 model family and the PTB dataset. We observe that, in the case of LlaMA-2 7b, the client models generally struggle with the PTB dataset from sparsity 60% and beyond, regardless of the number of clients. In many of those cases, the global model achieves up to 5 times better perplexity values. This demonstrates the effectiveness of FL in the pruning tasks. In addition, we do not observe there is noticeable trend in perplexity values with varying number of clients and the perplexity values of the global models are comparable regardless of the number of clients participating in FL, when the sparsity level is small.

6 Conclusion

In this paper, we introduced FedSpaLLM, a novel framework for pruning LLMs in federated learning settings. By addressing key challenges such as communication efficiency and system heterogeneity, FedSpaLLM enables collaborative pruning without sharing private data. Our method introduces several innovations, including the use of ℓ_0-norm aggregation for handling sparse model updates, an adaptive mask expansion technique to ensure target global sparsity and a layer sampling strategy that personalizes the pruning process based on client resources.

Through extensive experiments, we demonstrated that FedSpaLLM significantly reduces the communication and computational costs of LLM pruning while maintaining model performance across diverse, real-world federated environments. Our results showed that FedSpaLLM outperforms existing approaches in global model perplexity, making it a promising solution for resource-constrained applications where privacy is critical.

In future work, we aim to extend FedSpaLLM to other model compression techniques, such as quantization and distillation, to further improve efficiency in federated learning scenarios. Additionally, exploring more advanced techniques for client selection and resource-aware scheduling could enhance the adaptability and scalability of the framework to even larger LLMs and more heterogeneous environments.

7 Limitations

While FedSpaLLM introduces a highly efficient framework for pruning large language models in federated settings, there are a few limitations to consider. First, the effectiveness of the layer sampling strategy depends on the accurate estimation of client computational resources, which may vary dynamically in real-world deployments. This could lead to suboptimal layer sampling in highly fluctuating environments. Second, while our adaptive mask expansion ensures global sparsity, further refinements could improve its handling of extreme heterogeneity in client data distributions. Finally, our current experiments focus on moderate-scale LLMs, and additional work is required to assess scalability for larger models beyond several billion parameters.

8 Acknowledgement

This work was supported by the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research, under Contract DE-AC02-06CH11357. This research used resources of the Argonne Leadership Computing Facility at Argonne National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under contract DE-AC02-06CH11357.

References

Guangji Bai, Zheng Chai, Chen Ling, Shiyu Wang, Jiaying Lu, Nan Zhang, Tingwei Shi, Ziyang Yu, Mengdan Zhu, Yifei Zhang, et al. 2024a. Beyond efficiency: A systematic survey of resource-efficient large language models. *arXiv preprint arXiv:2401.00625*.

Guangji Bai, Yijiang Li, Chen Ling, Kibaek Kim, and Liang Zhao. 2024b. Sparsellm: Towards global pruning of pre-trained language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Dongqi Cai, Yaozong Wu, Shangguang Wang, Felix Xiaozhu Lin, and Mengwei Xu. 2023. Fedadapter: Efficient federated learning for modern nlp. In *MobiCom*.

Tim Dettmers and Luke Zettlemoyer. 2023. The case for 4-bit precision: k-bit inference scaling laws. In *International Conference on Machine Learning*, pages 7750–7774. PMLR.

Enmao Diao, Jie Ding, and Vahid Tarokh. 2021. Heterofl: Computation and communication efficient federated learning for heterogeneous clients. In *ICLR*.

Zihan Fang, Zheng Lin, Zhe Chen, Xianhao Chen, Yue Gao, and Yuguang Fang. 2024. Automated federated pipeline for parameter-efficient fine-tuning of large language models. *Arxiv*.

Elias Frantar and Dan Alistarh. 2023. Massive language models can be accurately pruned in one-shot. *arXiv preprint arXiv:2301.00774*.

Zishan Gu, Ke Zhang, Guangji Bai, Liang Chen, Liang Zhao, and Carl Yang. 2023. Dynamic activation of clients and parameters for federated learning over heterogeneous graphs. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 1597–1610. IEEE.

Song Han, Huizi Mao, and William J Dally. 2015. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*.

Charlie Hou, Akshat Shrivastava, Hongyuan Zhan, Rylan Conway, Trang Le, Adithya Sagar, Giulia Fanti, and Daniel Lazar. 2024. Pre-text: Training language models on private federated data in the age of llms. *arXiv preprint arXiv:2406.02958*.

Hong Huang, Lan Zhang, Chaoyue Sun, Ruogu Fang, Xiaoyong Yuan, and Dapeng Wu. 2023. Distributed pruning towards tiny neural networks in federated learning. *ICDCS*.

Itay Hubara, Brian Chmiel, Moshe Island, Ron Banner, Joseph Naor, and Daniel Soudry. 2021. Accelerated sparse neural training: A provable and efficient method to find n: m transposable masks. *Advances in neural information processing systems*, 34:21099–21111.

Xiaopeng Jiang and Cristian Borcea. 2023. Complement sparsification: Low-overhead model pruning for federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 8087–8095.

Yuang Jiang, Shiqiang Wang, Victor Valls, Bong Jun Ko, Wei-Han Lee, Kin K. Leung, and Leandros Tassiulas. 2022. Prunefl: Model pruning enables efficient federated learning on edge devices. *TNNLS*.

Yeachan Kim, Junho Kim, Wing-Lam Mok, Jun-Hyung Park, and SangKeun Lee. 2023. Client-customized adaptation for parameter-efficient federated learning. In *ACL Findings*.

Eldar Kurtic, Elias Frantar, and Dan Alistarh. 2023. Ziplm: Hardware-aware structured pruning of language models. *arXiv preprint arXiv:2302.04089*.

Woosuk Kwon, Sehoon Kim, Michael W Mahoney, Joseph Hassoun, Kurt Keutzer, and Amir Gholami. 2022. A fast post-training pruning framework for transformers. *Advances in Neural Information Processing Systems*, 35:24101–24116.

Lin Li, Jianping Gou, Baosheng Yu, Lan Du, and Zhang Yiand Dacheng Tao. 2024a. Federated distillation: A survey. *arXiv preprint arXiv:2404.08564*.

Yixiao Li, Yifan Yu, Qingru Zhang, Chen Liang, Pengcheng He, Weizhu Chen, and Tuo Zhao. 2023. Losparse: Structured compression of large language models based on low-rank and sparse approximation. *arXiv preprint arXiv:2306.11222*.

Zilinghan Li, Shilan He, Pranshu Chaturvedi, Volodymyr Kindratenko, Eliu A Huerta, Kibaek Kim, and Ravi Madduri. 2024b. Secure federated learning across heterogeneous cloud and high-performance computing resources-a case study on federated fine-tuning of llama 2. *Computing in Science & Engineering*.

Bill Yuchen Lin, Chaoyang He, Zihang Zeng, Hulin Wang, Yufen Huang, Christophe Dupuy, Rahul Gupta, Mahdi Soltanolkotabi, Xiang Ren, and Salman Avestimehr. 2022. Fednlp: Benchmarking federated learning methods for natural language processing tasks. In *NAACL Findings*.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. Llm-pruner: On the structural pruning of large language models. *arXiv preprint arXiv:2305.11627*.

Mitch Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. 1994. The penn treebank: Annotating predicate argument structure. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

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

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

Muhammad Tahir Munir, Muhammad Mustansar Saeed, Mahad Ali, Zafar Ayyub Qazi, and Ihsan Ayyub Qazi. 2021. Fedprune: Towards inclusive federated learning. *Arxiv*.

R OpenAI. 2023. Gpt-4 technical report. arxiv 2303.08774. *View in Article*, 2:13.

Gunho Park, Baeseong Park, Se Jung Kwon, Byeongwook Kim, Youngjoo Lee, and Dongsoo Lee. 2022. nuqmm: Quantized matmul for efficient inference of large-scale generative language models. *arXiv preprint arXiv:2206.09557*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Chao Ren, Han Yu, Hongyi Peng, Xiaoli Tang, Anran Li, Yulan Gao, Alysa Ziying Tan, Bo Zhao, Xiaoxiao Li, Zengxiang Li, et al. 2024. Advances and open challenges in federated learning with foundation models. *arXiv preprint arXiv:2404.15381*.

Jingwei Sun, Ziyue Xu, Hongxu Yin, Dong Yang, Daguang Xu, Yiran Chen, and Holger R. Roth. 2024. Fedbpt: Efficient federated black-box prompt tuning for large language models. *Arxiv*.

Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. 2023. A simple and effective pruning approach for large language models. *arXiv preprint arXiv:2306.11695*.

Chaofan Tao, Lu Hou, Haoli Bai, Jiansheng Wei, Xin Jiang, Qun Liu, Ping Luo, and Ngai Wong. 2023. Structured pruning for efficient generative pre-trained language models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10880–10895.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Shikhar Tuli and Niraj K Jha. 2023. Acceltran: A sparsity-aware accelerator for dynamic inference with transformers. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*.

Haozhao Wang, Yichen Li, Wenchao Xu, Ruixuan Li, Yufeng Zhan, and Zhigang Zeng. 2023. Dafkd: Domain-aware federated knowledge distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20412–20421.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Feijie Wu, Zitao Li, Yaliang Li, Bolin Ding, and Jing Gao. 2024. Fedbiot: Llm local fine-tuning in federated learning without full model. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3345–3355.

Yuxuan Yan, Shunpu Tang, Zhiguo Shi, and Qianqian Yang. 2024. Federa: Efficient fine-tuning of language models in federated learning leveraging weight decomposition. *Arxiv*.

Zhewei Yao, Reza Yazdani Aminabadi, Minjia Zhang, Xiaoxia Wu, Conglong Li, and Yuxiong He. 2022. Zeroquant: Efficient and affordable post-training quantization for large-scale transformers. *Advances in Neural Information Processing Systems*, 35:27168–27183.

Kai Yi, Nidham Gazagnadou, Peter Richtarik, and Lingjuan Lyu. 2024. Fedp3: Federated personalized and privacy-friendly network pruning under model heterogeneity. *ICLR*.

Chen Zhang, Yu Xie, Hang Bai, Bin Yu, Weihong Li, and Yuan Gao. 2021. A survey on federated learning. *Knowledge-Based Systems*, 216:106775.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Haodong Zhao, Wei Du, Fangqi Li, Peixuan Li, and Gongshen Liu. 2023. Fedprompt: Communication-efficient and privacy preserving prompt tuning in federated learning. In *ICASSP*.

Michael Zhu and Suyog Gupta. 2017. To prune, or not to prune: exploring the efficacy of pruning for model compression. *arXiv preprint arXiv:1710.01878*.

Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. 2023. A survey on model compression for large language models. *arXiv preprint arXiv:2308.07633*.

A Theoretical Proofs

A.1 Proof of Corollary 1 (Sparsity Guarantee)

Proof. 1. **Client Sparsity Consistency:** Each client enforces the same target sparsity $\mathcal{S}_{global}$. This implies that for each client i, the sparsity of the pruned layers $\mathcal{L}_i$ matches the global sparsity target $\mathcal{S}_{global}$. Formally, we have:

$$\mathcal{S}_i = \mathcal{S}_{global}, \quad \forall i = 1, 2, \ldots, N.$$

Since all clients prune their local models independently but according to the same target sparsity, each pruned local model achieves the same sparsity.

2. **Layer Sampling Strategy:** The layer sampling strategy ensures that all layers of the model $\mathcal{L}$ are eventually sampled across all clients during each communication round. Therefore, every layer in the global model $\hat{W}_g$ has been pruned by each client according to the same sparsity criterion $\mathcal{S}_{global}$.

3. **Aggregation with ℓ_0-norm:** The aggregation function using ℓ_0-norm averages only the non-zero elements (i.e., the pruned weights) from the client models. Since all clients enforce the same sparsity, and the aggregation only involves the non-zero weights from these pruned models, the sparsity of the aggregated global model will match that of the local models. Specifically:

$$\mathcal{S}_{global} = \mathcal{S}_i, \quad \forall i.$$

Thus, the sparsity of the global model after aggregation is equivalent to the sparsity of each client model.

4. **Conclusion:** Therefore, the global model $\hat{W}_g$ will maintain the target sparsity $\mathcal{S}_{global}$ after aggregation in each communication round. The aggregation process ensures that the global sparsity is consistent with the target sparsity across rounds.

$$\boxed{\mathcal{S}_{global} = \mathcal{S}_i, \quad \forall i.}$$

□

A.2 Proof of Theorem 1 (Unbiased Estimator)

Proof. 1. **Layer Sampling Strategy:** Let $\mathcal{L} = \{L_1, L_2, \ldots, L_m\}$ denote the set of all layers in the model, where m is the total number of layers. In each communication round, the server randomly samples a subset of layers $\mathcal{L}_i \subseteq \mathcal{L}$ for each client i. Each layer $L_j \in \mathcal{L}$ has an equal probability p_j of being selected across clients.

The expectation of the sampled weights for layer L_j across all clients can be expressed as:

$$\mathbb{E}[W_i[L_j]] = p_j W_g^*[L_j],$$

where $W_g^*[L_j]$ is the weight of layer L_j in the fully pruned global model W_g^*.

2. **Unbiasedness of Layer Sampling:** Since each layer has an equal probability of being sampled across clients, the expected contribution of each layer is proportional to its selection probability. Over multiple communication rounds, all layers will be sampled enough times to represent the fully pruned model.

Therefore, the expected value of the global model $\hat{W}_g$ is the same as the fully pruned model W_g^*. For any layer L_j, we have:

$$\begin{aligned}\mathbb{E}[\hat{W}_g[L_j]] &= \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N} W_i[L_j]\right] \\ &= \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}[W_i[L_j]] = W_g^*[L_j].\end{aligned} \tag{12}$$

Thus, the global model $\hat{W}_g$ is an unbiased estimator of W_g^*, as the expected value of the pruned weights matches the fully pruned model.

3. **Conclusion:** Therefore, the global model $\hat{W}_g$ obtained by aggregating pruned models after layer sampling is an unbiased estimator of the fully pruned model W_g^*. Formally, we can conclude that:

$$\boxed{\mathbb{E}[\hat{W}_g] = W_g^*}.$$

This unbiased property holds as long as each layer has an equal probability of being selected across clients during each communication round. □