Parameter-Averaging Laws for Multitask Language Models

Woojin Chung
Graduate School of AI, KAIST
gartland@kaist.ac.kr

Hyowon Cho
Graduate School of AI, KAIST
hyyoka@kaist.ac.kr

James Thorne
Graduate School of AI, KAIST
thorne@kaist.ac.kr

SeYoung Yun
Graduate School of AI, KAIST
yunseyoung@kaist.ac.kr

Abstract

Parameter-averaging, a method for combining multiple models into a single one, has emerged as a promising approach to enhance performance without requiring additional space or retraining. Nonetheless, the conditions for successful parameter-averaging remain undefined, calling for further research to characterize them. In this study, we empirically investigate the influential factors for successful parameter-averaging and reveal *positive correlations between representation power and the performance gain of parameter-averaging*. Specifically, we evaluate how computational budget, data diversity and vocabulary size contribute to representation power, and their influence on the success of parameter-averaging. Our results demonstrate that parameter-averaging improves the generalization ability for both in-domain and out-of-domain data. Additionally, to reduce the computational cost of parameter-averaging, we introduce *partial averaging*, which assumes arbitrary participation of a subset of contributors. We observe that partial averaging outperforms fine-tuning for models with sufficient representation power. Furthermore, we find that the impact of data heterogeneity, which arises from different data distributions of contributors, reduces as the representation power of the model increases. These findings provide valuable insights into the principles governing parameter-averaging and its potential for enhancing model performance.

1 Introduction

The advantages of multitask learning in the field of Natural Language Processing (NLP) have been consistently observed in numerous studies [4, 6, 34, 35, 36, 44, 45]. Multitask learning refers to a training approach where machine learning models are trained using data from multiple tasks simultaneously. By utilizing shared representations, these models learn common concepts and patterns across a set of related tasks. Combining knowledge from multiple datasets to build a unified model can enhance the generalization ability of the model on in-domain data [35] and out-of-domain data [44].

However, updating models with new knowledge often requires retraining, which can be computationally inefficient. To address this issue, researchers have recently explored strategies that involve combining multiple models. One common technique is model ensembling, where the outputs of individual models are aggregated to generate the final prediction [33, 40]. Model ensembling shows promising results outperforming standard fine-tuning [8, 14, 25]. Nevertheless, this technique requires additional space and computational resources to accommodate multiple models and generate the final prediction. Another approach is *parameter-averaging* (i.e. model merging), which combines

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

multiple models into a single one in parameter space [29]. Parameter-averaging allows the knowledge of multiple models to be consolidated into a single model without the need for additional space.

In addition to computational and parameter efficiency, parameter-averaging has shown potential for improving performance. Recent research has demonstrated that simple averaging of parameters can enhance the performance of a specific model when fine-tuned with the same dataset but different hyperparameters [46]. Other works have focused on improving performance by merging models using various weighted averaging algorithms on multiple domains or tasks [8, 19, 29]. Researchers have also investigated merging models on the same task but trained on different datasets and or tasks form can enhance the performance [13, 37].

Parameter-averaging methods can fail under certain circumstances, such as when including failed fine-tuned models in the merging process [20, 46]. However, while previous works have demonstrated the effectiveness and potential of parameter-averaging, none of them have specified *the conditions under which it is successful*, requiring further research to identify the factors that influence its success. In this work, we first investigate the underlying laws governing parameter-averaging and show *a strong correlation between the representation power of a model and the performance gains achieved through parameter-averaging*. Specifically, we evaluate how the effect of computational budget, data diversity and vocabulary size contribute to representation power, and their influence on the success of parameter-averaging. We also show that parameter-averaging improves the generalization ability for both in-domain and out-of-domain data, which is an evidence that it can truly combine knowledge from multiple models. To support our findings, we performed an empirical study, comparing 13 fine-tuned language models with inherently controlled variables.

We further explore a way to maximize computational efficiency. To reduce the computational cost of parameter-averaging, we introduce *partial averaging*, which assumes arbitrary participation of a subset of contributors. We show that the performance can be maintained for models with sufficient representation power. Also, we observe that the effect of data heterogeneity [21] decreases as the representation power of the model increases, suggesting that parameter-averaging models with sufficient representation power are robust to various contributors with different data distribution.

Our key findings are as follows:

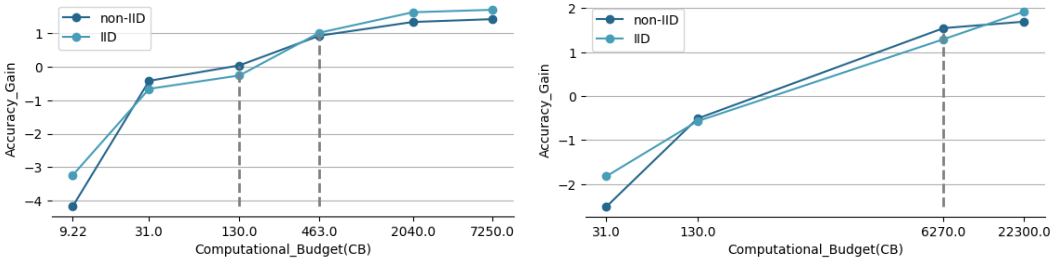
- We find that the performance gain of parameter-averaging has a positive correlation with the representation power of a model (Section 4.)
- We observed that parameter-averaging can improve the generalization ability of both in-domain and out-of-domain data (Section 5).
- We first show that partial averaging, assuming arbitrary participation of a subset of contributors, outperforms full fine-tuning for models with sufficient representation power (Section 6).
- We find that the impact of data heterogeneity, which arises from different data distributions of contributors, reduces as the representation power of the model increases (Section 7).

2 Related Works

2.1 Representation Power of Language Models

Computational Budget. Previous study first demonstrates that the performance of a model, which reflects its representation power, follows a power-law relationship with three scaling factors, including the number of model parameters(excluding embeddings), the dataset size, and the compute used for training [22, 39]. Furthermore, one research found that the model size and dataset size should be scaled in equal proportions [15].

Vocabulary Size. Vocabulary size is another attribute that affects the representation power of a language model. It is widely known that a larger vocabulary size generally improves the accuracy of tokenization, especially for handling Out-Of-Vocabulary words, as it captures a wider range of distinct words or subword units. Recent studies show that vocabulary size directly affects the performance of language models, suggesting that the proper vocabulary size per language for accurate tokenization is about 30k vocabulary [12]. Several multilingual NLP works demonstrate that increasing the vocabulary size improves the representation ability of multilingual language models [5, 26, 48].



(a) Monolingual case

(b) Multilingual case

Figure 1: We conducted accuracy tests on NLI datasets to evaluate the performance of both monolingual and multilingual models. The Computational Budget (CB) was set at N*10^{15}. We used a dotted vertical line to indicate the point at which parameter-averaging surpasses standard fine-tuning. The graph illustrates the relationship between the performance gain achieved through parameter-averaging and the increase in computational budget. Additionally, the graph shows the distinction between the IID and Non-IID settings. Although the exact point of exceeding differs, both the monolingual and multilingual cases exhibit a clear linear correlation between the performance gain and computational budget.

Dataset Diversity. Recent works have consistently shown that the diversity of pre-training corpus leads to better downstream generalization capability [11, 42, 23]. By enabling a model to see various domains and knowledge, the general cross-domain knowledge and downstream generalization capabilities of the model improve compared to models trained on only a handful of data sources [11].

2.2 Theoretical explanation for Parameter-averaging

Previous studies on the geometry of loss landscapes have consistently revealed that models trained on the same dataset generally converge into a single, connected linear basin of the loss surface [9, 10, 31]. This observation has led to the widely accepted notion that neural networks trained on the same dataset will maintain loss when interpolated in the weight space. Models on the same task but trained on different datasets have local minima which are interconnected [13, 37]. Specifically, language models that have been fine-tuned on the same dataset form a compact cluster in the weight space and those fine-tuned on different datasets or tasks form a more dispersed cluster. Furthermore, merging two models inside a specific region leads to comparable or even better performance than models found via fine-tuning, even on tasks that the original models were not fine-tuned on [13].

2.3 Merging of Language Models

With theoretical guarantees of the effectiveness of parameter-averaging trained models [1, 10], researchers have been exploring the potential of combining language models through two approaches: model ensembling and parameter-averaging. Model ensembling has achieved impressive results, surpassing standard fine-tuning [14, 17, 25]. However, it requires additional space to accommodate all the expert language models' outputs for the final prediction.

On the other hand, parameter-averaging algorithms combine multiple models into a single model in parameter space. This approach is highly efficient, as it maintains constant inference cost and space[29], allowing it to be extended to a larger number of tasks. Also, parameter-averaging has demonstrated great potential for success in addition to its efficiency. Recent research indicates that simple parameter-averaging can enhance the performance of a specific model when fine-tuned with different hyperparameters using the same dataset [46]. Other studies have focused on improving performance by merging models using various weighted averaging algorithms to construct a single model applicable to multiple domains or tasks, capable of generalizing to new domains [8, 19, 29].

Federated learning is another line of work that naturally uses merging. In federated learning, multiple contributors collaborate to train a centralized model by merging their locally trained parameters to form a new set of parameters. Works on federated learning have also focused on a partial client participation scenario where only a small fraction of the total number of contributors is merged, as it

Table 1: Summary of the representation power of models. The representation power of a model is determined by the combination of its computational budget, vocabulary size and the diversity of training datasets (i.e. number of languages). The computational budget of a model is determined by the combination of its parameter size (**N**), dataset size (**B**), and the compute used for training (**S**). More precisely, the computational budget is denoted as 6*N*B*S. The models are presented in ascending order.

Name	Lang. Count	Vocab. Size	Model Size	Dataset Size	Training Steps	Batch Size	Computational Budget
TINY-BERT	1	30,522	6M	16GB (3.3B)	1M	256	$9.22 * 10^{15}$
DISTILBERT [41]	1	30,522	43M	16GB (3.3B)	15K	8000	$3.1 * 10^{16}$
DISTILMBERT [41]	104	119,547	43M	67.4GB	15K	8000	$3.1 * 10^{16}$
BERT-BASE [7]	1	30,522	85M	16GB (3.3B)	1M	256	$1.3 * 10^{17}$
MBERT [34]	104	119,547	85M	67.4GB	1M	256	$1.3 * 10^{17}$
BERT-LARGE [7]	1	30,522	302M	16GB (3.3B)	1M	256	$4.63 * 10^{17}$
ROBERTA-BASE [27]	1	50,265	85M	160GB (30B)	0.5M	8000	$2.04 * 10^{18}$
BART-BASE [24]	1	50,265	99M	160GB (30B)	0.5M	8000	$2.38 * 10^{18}$
XLMR-BASE [5]	100	250,002	85M	2.5TB (296.5B)	1.5M	8192	$6.27 * 10^{18}$
XLMV-BASE [26]	100	901,629	85M	2.5TB (296.5B)	1.5M	8192	$6.27 * 10^{18}$
ROBERTA-LARGE [27]	1	50,265	302M	160GB (30B)	0.5M	8000	$7.25 * 10^{18}$
BART-LARGE [24]	1	50,265	353M	160GB (30B)	0.5M	8000	$8.47 * 10^{18}$
XLMR-LARGE [5]	100	250,002	302M	2.5TB (296.5B)	1.5M	8192	$2.23 * 10^{19}$

is unrealistic to anticipate all contributors to participate in every single round of federated learning training [3, 18, 28, 47].

3 Experiment Setting

3.1 Data Partitioning

We consider three different training settings: centralized, IID and Non-IID. In the centralized setting, the model performs standard training using all available data. IID and Non-IID perform distributed learning with multiple clients, assigning a portion of the data to each client, which sees data for multiple classes sampled from all data (IID) or sees data for one class (Non-IID) . For example, each client under IID setting sees various languages in multilingual datasets, while each client under non-IID setting sees only one language.

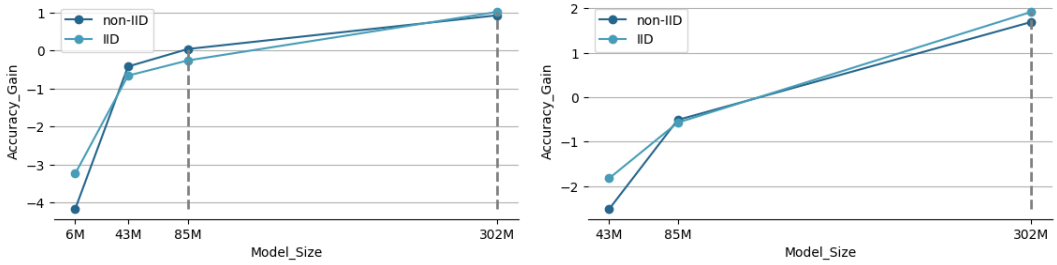
3.2 Algorithms

To analyze the tendency of parameter-averaging, we use the classic FL algorithm called Federated Average (FedAvg) [30], which performs uniform averaging between multiple clients. In the given communication round t, K active clients among N clients run stochastic gradient descent (SGD) on their local data. The central server distributes global model parameters w^t to these clients, and after a certain number of steps, the clients send their updated parameters to the central server. The server then averages these updates into a single centralized set of parameters, $w^{t+1} = \sum_i p_i w_i^{t+1}$, using the weighted sum of client weights p_i, which are proportional to the amount of training data stored on each client i, i.e. $p_i = \frac{n_i}{\sum_i n_i}$. The centralized parameters are then broadcast to each client, and the process is repeated for the next round.

The application of Federated Learning (FL) scenario allows for the comprehensive examination of the interdependent impacts that arise from the core aspects of parameter-averaging. These factors include the number of clients, training steps for each client, the number of averaging, the nature and size of data assigned to individual clients, and the overall data volume. To demonstrate this, we conducted several experiments by modifying the local iteration of each model and the number of averaging. This allowed us to validate the impact of these features on the overall process. Results are reported in Appendix B.

3.3 Models

We provide a summary of the configuration for each model used in our study, as shown in Table 1. We carefully selected 13 language models that exhibit high relevance while also being distinguishable based on the factors below.



(a) Monolingual case (b) Multilingual case

Figure 2: Results on MNLI and XNLI dataset. The results indicate that accuracy gain improves with model size. We selected five models for comparison, including TINY-BERT(6M), BERT-BASE(85M), and BERT-LARGE(302M), which share the same architecture, training objective, and pre-training corpus but differ in size for monolingual models. Additionally, we evaluated two models for the multilingual case: DISTILMBERT(43M) and MBERT(85M). Furthermore, we also evaluated two models, namely DISTILMBERT(43M) and MBERT(85M), for the multilingual case. The results from this evaluation reveal a similar trend to that observed in the monolingual case, further affirming the relationship between model size and accuracy gain.

Model Size To isolate the effect of model size, we compare the performance of TINY-BERT, BERT-BASE, and BERT-LARGE. Since there is no publicly available pre-trained weight for TINY-BERT, we trained it from scratch (refer to Appendix A.1 for details).

Dataset Size To directly observe the impact of dataset size, we utilized two pairs of models: BERT-BASE and ROBERTA-BASE for the monolingual case, and MBERT and XLMR-BASE for the multilingual case. Each pair consists of models with the same size and maximum input length.

Vocabulary Size We selected two models, XLMR-BASE and XLMV-BASE, to investigate the influence of vocabulary size. The XLMV-BASE model uses a larger vocabulary compared to the XLMR-BASE model.

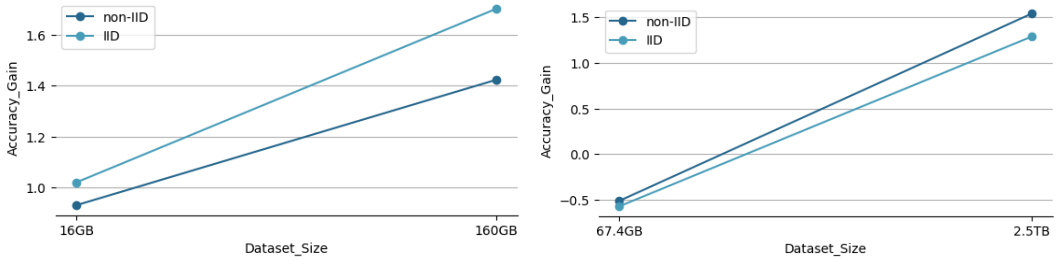
Dataset Diversity To assess the impact of pre-training corpus diversity, we compare the performance of BERT, MBERT, DISTILBERT, and DISTILMBERT. These models differ only in terms of the number of languages included in their pre-training corpus.

3.4 Training and Evaluation Strategy

We fine-tuned models with a learning rate of 2e-5 and batch size of 32. Each contributors in parameter-averaging run 2 epochs with a fixed batch size per task, and we perform 5 rounds of parameter-averaging as it was the best setting in both full parameter averaging and partial averaging (Appendix B). Each contributors were fine-tuned on train dataset with 77K examples and evaluated on test dataset with 5K examples in monolingual and multilingual setting. We run each experiment with three different random seeds and report the average accuracy. For evaluation, we defined *accuracy gain* as the gap between parameter-averaged models and centralized model.

4 Representation Power and Parameter-Averaging

What drives the success of merging? This question remained unanswered in studies focusing on parameter-averaging [8, 25]. We revealed that the representation power of the model and the accuracy gain of parameter-averaging has a positive correlation. In this section, we explore the impact of various aspects related to 'the representation power of the model. This correlation offers a definitive guide on choosing models that maximize the benefits of parameter-averaging. We also show this correlation appears to be model and task agnostic.



(a) Monolingual case

(b) Multilingual case

Figure 3: Results on MNLI and XNLI datasets. We paired two models with different pre-training datasets. Specifically, we compared BERT-BASE and ROBERTA-BASE for the monolingual case and MBERT and XLMR-BASE for the multilingual case. The accuracy gain shows a positive correlation with the size of the dataset in both cases.

4.1 Effect of Model Sizes

The size of a model has a significant impact on its performance, and this impact also extends to the effectiveness of parameter-averaging. In our study, we discovered that the accuracy gain and the model size has a positive correlation, as illustrated in Figure 2.

To investigate this correlation, we compared the performance of three models: TINY-BERT, BERT-BASE, and BERT-LARGE. For the largest model, 302M (BERT-LARGE), the average accuracy for both IID and Non-IID settings was higher than that of centralized setting. In contrast, for the smallest model, 6M (TINY-BERT), the average accuracy was significantly lower than that of centralized setting. For the 85M (BERT-BASE) model, we observed no accuracy gain, one getting slight improvement while another showing slight degradation. As the only difference between these models is their size, it is clear that the size of the model and the impact of parameter-averaging have positive correlation,

Similar trends were observed in a multilingual setting. We found that while parameter-averaging had a negative impact on the performance of the 85M (MBERT) and 43M (DISTILMBERT) models, the negative impact of parameter-averaging was smaller for the larger 85M (MBERT) model. This indicates that parameter-averaging large models enhances the performance, while parameter-averaging with small models such as TINY-BERT and DISTILBERT will have negative impact only.

4.2 Effect of Dataset Sizes

The representation power of a single language model relies on several factors, including batch size **B** and training step **S**, which can be directly expressed by dataset size **(D)**. The size of the dataset used for pre-training can be decomposed into the multiplication of **B**, **S**, and maximum input length **L** divided by the number of epochs **E**, i.e. $D = \frac{L*B*S}{E}$. Therefore, we decided to simultaneously evaluate the impact of batch size and training step on parameter-averaging by varying dataset size.

To exclude the involvement of other variables, we selected BERT-BASE and ROBERTA-BASE for the monolingual case and MBERT and XLMR-BASE for the multilingual case. Figure 3 illustrates that increasing dataset size boosts the positive impact of parameter-averaging. While BERT-BASE did not achieve any significant performance improvement through parameter averaging, ROBERTA-BASE achieved accuracy gains of 1.63 on IID and 1.39 on Non-IID settings when compared to the central fine-tuning method. Similarly, XLMR-BASE benefited more from parameter-averaging than MBERT.

As previous studies have demonstrated that the dataset size directly affects the representation power of a language model [15, 22] and ROBERTA-BASE and XLMR-BASE is trained with larger datasets, we concluded that they possess stronger representation power than their counterparts.

4.3 Effect of Vocabulary Sizes

We have analyzed the degradation caused by parameter-averaging, which arises from their limited representation power resulting from a smaller vocabulary size.

Previous research has indicated that a model must have a minimum vocabulary size of 30K per language to achieve high-quality language representation [12, 48]. However, recent multilingual models only have 250K vocabulary size to represent 100 languages. It is 2.5K vocabulary per language on average, which is not sufficient to encode individual language precisely. Therefore, to explore the impact of vocabulary size on the accuracy gain of parameter averaging, we compared the XLMR-BASE and XLMV-BASE models. XLMV-BASE shares same configuration with XLMR-BASE except for vocabulary size, which is 900K. The accuracy gain of parameter averaging XLMR-BASE model in IID partition was 1.29 whereas accuracy gain in XLMV-BASE model was 1.41, as reported in Table 2. It clearly shows that increasing vocabulary size leads to the higher accuracy gain of parameter averaging.

Table 2: The size of the vocabulary is related to the accuracy gain by parameter-averaging.

	XLMV-BASE	XLMR-BASE
Model Size	85M	85M
Vocab. Size	900k	250K
Acc. Gains	+1.41	+1.29

4.4 Effect of the Dataset Diversity

Previous works show that using diverse pre-training corpus enhances the representation power of language models [11]. One clear example is that a multilingual model trained on 7 languages outperforms its monolingual counterpart [5]. We conducted experiments to verify whether a similar correlation exists between the diversity of the pre-training corpus and the model's parameter-averaging capability. To this end, we compared multilingual and monolingual models with the same configuration except for the number of languages they trained with. These models were then fine-tuned on the MNLI dataset. Table 3 shows a correlation between the diversity of the dataset and the accuracy gained through parameter-averaging. While both the DISTILBERT model, trained exclusively on English, and the DISTILMBERT model, trained on multiple languages, did not surpass the standard fine-tuned model, we observed that parameter-averaging yielded better results for DISTILMBERT compared to DISTILBERT. Similarly, parameter-averaging proved beneficial for MBERT, while BERT-BASE did not exhibit the same advantage.

Table 3: Results with MNLI dataset using monolingual and multilingual models. Language models trained with diverse pre-training datasets have better scores after parameter-averaging.

Model	Method	Acc. Gains
DISTILBERT	IID	-0.66
	Non-IID	-0.42
DISTILMBERT	IID	-0.32
	Non-IID	-0.19
BERT-BASE	IID	-0.26
	Non-IID	+0.04
MBERT	IID	+0.11
	Non-IID	+0.37

5 Generalization Ability of Parameter-Averaging

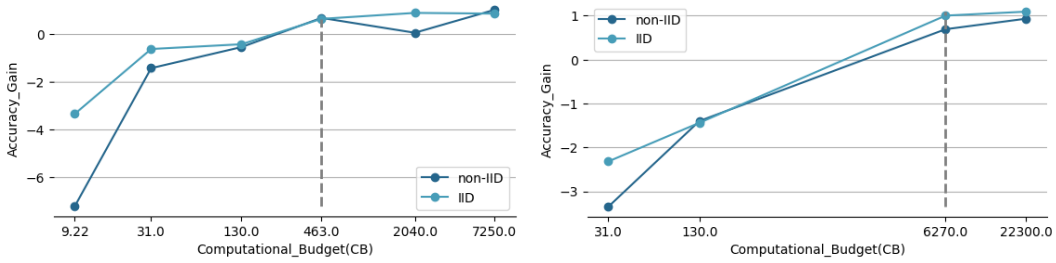
In this section, we show that parameter-averaging models with sufficient representation power enhances generalization ability of the model on both in-domain and out-of-domain data. This finding suggests that parameter-averaging has the potential to be a better alternative for fine-tuning large language models without requiring extra computational budget. In Table 4, we present the accuracy scores for each language in the XNLI dataset using a Non-IID setting with XLMR-BASE.

In the *full* scenario, all languages are included in the fine-tuning process, whereas in the *partial* scenario, only randomly selected languages from the XNLI dataset are used for training. Languages which are not selected during fine-tuning stage are considered as unseen, allowing us to evaluate the model's generalization ability on out-of-domain data. Experiment are run as follows: 3 contributors out of 15 contributors are randomly sampled at every parameter-averaging round. We run for 5 parameter-averaging round as it is minimum round to select and fine-tune every contributors. In this experiment, only 9 contributors were selected during fine-tuning. Further details of training procedure can be found in Appendix A.2.

Central(*partial*) vs Non-IID(*partial*). In this experiment, we compare standard fine-tuned model and parameter-averaged model trained on partial scenario. In seen languages, parameter-averaged model and standard fine-tuned model show similar accuracy whereas in unseen languages, parameter-averaged model outperforms its baseline counterpart by 1.45 point on average. This outcome reveals

Table 4: Results of parameter-averaging models trained on Non-IID partitions using XNLI dataset. We compared the performance of two types of models: merged models trained with 9 seen languages, and fully merged models trained with all 15 languages. Additionally, we conducted a comparison between the performance of the merged models and that of the central fine-tuned models. **Bold** numbers indicate the best-performing results and underline indicates the second best performance.

Method	Unseen							Seen										Total Avg.
	es	el	bg	ar	vi	zh	Avg.	en	fr	de	ru	tr	th	hi	sw	ur	Avg.	
Central (*full*)	78.08	75.94	77.68	71.94	76.48	**75.66**	75.96	81.94	77.04	75.96	75.44	73.40	72.36	71.30	68.71	67.65	73.75	74.63
Central (*partial*)	77.14	74.95	76.86	71.49	75.50	74.97	75.15	82.45	78.64	77.62	76.66	75.38	**73.90**	**73.14**	68.93	69.27	75.11	75.13
Non-IID (*full*)	**79.86**	**78.00**	**79.3**	**74.74**	**77.64**	75.62	**77.53**	**84.30**	**79.06**	**78.53**	**77.44**	**75.36**	72.96	72.70	68.85	69.77	**75.44**	**76.27**
Non-IID (*partial*)	78.79	77.52	78.55	72.82	77.36	74.56	76.60	83.40	78.47	78.49	76.20	75.06	72.78	72.34	**69.23**	**70.19**	75.13	75.71



(a) Monolingual case (b) Multilingual case

Figure 4: The graph shows positive correlation between computational budget and the performance gain of partial averaging. When the representation power of the model is sufficiently strong, partial average outperforms standard fine-tuning.

the fact that parameter-averaging effectively enhances contributor's generalization capability to unseen dataset.

Central(*full*) vs Non-IID(*partial*). In this experiment, we compare standard fine-tuned model trained on full scenario and parameter-averaged model trained on partial scenario. In both seen and unseen languages, parameter-averaged model outperforms standard fine-tuned model by 1.38 point and 0.64 point on average. This result highlights the effectiveness of parameter-averaging in incorporating new knowledge.

Non-IID(*full*) vs Non-IID(*partial*). In this experiment, we compare parameter-averaged model trained on full and partial scenario. In both seen and unseen languages, parameter-averaged model with full scenario outperforms its partial counterpart by 0.31 point and 0.93 point on average. Given the fact that full scenario requires 5 times more computational power than partial scenario, performance gap can be considered relatively small. This finding motivates us to explore ways to reduce computational costs while preserving performance in parameter-averaging.

6 Partial Averaging

While it is possible in theory to merge a large number of models, it is less practical when it comes to deployment. The computational cost of training and merging all clients can be prohibitively expensive, especially in scenarios where there are a large number of contributors. Moreover, averaging all contributors can lead to delayed convergence [8].

Therefore, we further explore a way to maximize the computational efficiency. To reduce the computational cost of parameter-averaging, we introduce *partial averaging* (or partial client participation scenario in federated learning). Partial averaging involves selecting a subset of contributors to participate in the parameter-averaging process, rather than averaging all clients. Note that the we randomly select contributors. This technique is realistic but often reported as more challenging scenario [3, 18, 28, 47].

To assess the effectiveness of partial averaging, we conducted experiments with XNLI and MNLI datasets. The process of parameter-averaging requires a minimum of two contributors. To enable this, the proportion of contributors participating in the merging process varied. In the case of MNLI, 40%

of the total contributors were involved in averaging, while in XNLI, the proportion was 20%. This difference also implies a corresponding reduction in computational costs.

Interestingly, as depicted in Figure 6, we observed that if the language model possesses sufficient representation power, partial averaging can outperform full fine-tuning. Additionally, Table 5 presents the results that demonstrate that although full parameter-averaging leads to better performance compared to partial averaging, the difference between the two approaches is relatively small despite a significant gap in computational costs. This suggests that partial averaging can be considered an efficient method for fine-tuning.

7 Data Heterogeneity and Parameter-Averaging

Despite the practical reality of Non-IID data samples being distributed across various clients [16], handling this issue remains a significant challenge in the field of Federated Learning (FL) [43]. When models trained on heterogeneous data are averaged, it leads to a phenomenon known as client drift, where gradients become orthogonal to each other. This, in turn, causes the global server optimizer to diverge, as it is guided by the loss of the averaged models [32].

However, recent studies have shown that utilizing a pre-trained model can significantly reduce the performance gap between IID and Non-IID [32]. In addition, it has been found that the performance gap can be further reduced when using a pre-trained transformer model with stronger representation power in image classification task [38].

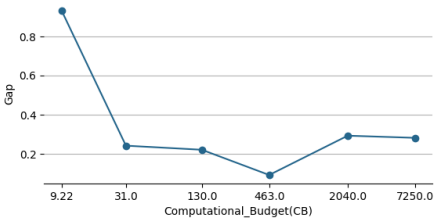


Figure 5: The performance gap between IID and Non-IID was negligible when the model has sufficient representation power.

In order to investigate whether this trend also holds true in the field of natural language processing (NLP), we conducted experiments using larger pre-trained language models. The result in Figure 5 indicates that using stronger representation power models can further reduce the performance gap between IID and Non-IID. Therefore, it can be concluded that and stronger representation power by pre-training in scale can almost fully mitigate the negative effects of data heterogeneity and therefore, data heterogeneity is negligible when merging sufficiently strong models.

8 Explaining Previous results

Our findings further support and offer preliminary explanations in line with recent studies on parameter-averaging. Previous research has indicated that a single parameter-averaged model surpasses the performance of the ROBERTA-BASE model [8], and our results align with this observation, indicating that the ROBERTA-BASE model possesses sufficient representation power to benefit from parameter-averaging. Another recent study demonstrates that the perplexity gap between base language models without parameter averaging and a single uniform parameter-average decreases as the number of model parameters increases, both during pre-training and inference stages [25]. Similarly, this perspective suggests that merging adapters may not yield the same positive impact as merging the full model, as adapters generally have less representation power than the full model itself [17].

9 Discussions and Conclusions

Broader Impacts Parameter-averaging is an important paradigm that offers several benefits, such as expanding new knowledge, efficient fine-tuning and privacy preservation. Nonetheless, this technique requires additional measures to prevent the involvement of malicious contributors, which can harm the overall performance.

Conclusions and Limitations Our result conveys a single, consistent message: There is a clear positive correlation between the representation power of a language model and the accuracy gain of parameter-averaging. In Section 4, we conclude that computational budget(4.1, 4.2), vocabulary size(4.3) and dataset diversity(4.4) affect accuracy gain of parameter-averaging. We further observe

that merging can improve the generalization ability on out-of-domain data on Section 5. In addition, Section 6 shows that partial averaging, more realistic deployment setting, can also defeat standard fine-tuned model. We also show in Section 7 that the effect of data heterogeneity has negligible impact on parameter-averaging as contributors have sufficient representation power. We expect for future works to further explore merging on (i) large-scale language models and (ii) broader region such as parameter-averaging models trained with different tasks.

Acknowledgements

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by Korea government (MSIT) [No. 2021-0-00907, Development of Adaptive and Lightweight Edge-Collaborative Analysis Technology for Enabling Proactively Immediate Response and Rapid Learning, 90%] and [No. 2019-0-00075, Artificial Intelligence Graduate School Program (KAIST), 10%].

References

[1] Samuel K Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. "Git re-basin: Merging models modulo permutation symmetries". In: *arXiv preprint arXiv:2209.04836* (2022).

[2] Daniel J Beutel et al. "Flower: A Friendly Federated Learning Research Framework". In: *arXiv preprint arXiv:2007.14390* (2020).

[3] Jieming Bian et al. *Accelerating Hybrid Federated Learning Convergence under Partial Participation*. 2023. arXiv: 2304.05397 [cs.DC].

[4] Rich Caruana. *Multitask Learning*. July 1997. DOI: https://doi.org/10.1023/A:1007379606734. URL: https://doi.org/10.1023/A:1007379606734.

[5] Alexis Conneau et al. *Unsupervised Cross-lingual Representation Learning at Scale*. 2019. DOI: 10.48550/ARXIV.1911.02116. URL: https://arxiv.org/abs/1911.02116.

[6] Michael Crawshaw. *Multi-Task Learning with Deep Neural Networks: A Survey*. 2020. arXiv: 2009.09796 [cs.LG].

[7] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://aclanthology.org/N19-1423.

[8] Shachar Don-Yehiya et al. "ColD Fusion: Collaborative Descent for Distributed Multitask Finetuning". In: *arXiv preprint arXiv:2212.01378* (2022).

[9] Rahim Entezari et al. *The Role of Permutation Invariance in Linear Mode Connectivity of Neural Networks*. 2021. DOI: 10.48550/ARXIV.2110.06296. URL: https://arxiv.org/abs/2110.06296.

[10] Jonathan Frankle et al. "Linear mode connectivity and the lottery ticket hypothesis". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 3259–3269.

[11] Leo Gao et al. *The Pile: An 800GB Dataset of Diverse Text for Language Modeling*. 2020. arXiv: 2101.00027 [cs.CL].

[12] Jonas Geiping and Tom Goldstein. *Cramming: Training a Language Model on a Single GPU in One Day*. 2022. arXiv: 2212.14034 [cs.CL].

[13] Almog Gueta et al. *Knowledge is a Region in Weight Space for Fine-tuned Language Models*. 2023. arXiv: 2302.04863 [cs.LG].

[14] Suchin Gururangan et al. "Scaling Expert Language Models with Unsupervised Domain Discovery". In: *arXiv preprint arXiv:2303.14177* (2023).

[15] Jordan Hoffmann et al. *Training Compute-Optimal Large Language Models*. 2022. arXiv: 2203.15556 [cs.CL].

[16] Wei Huang et al. *Fairness and Accuracy in Federated Learning*. 2020. arXiv: 2012.10069 [cs.LG].

[17] Joel Jang et al. *Exploring the Benefits of Training Expert Language Models over Instruction Tuning*. 2023. DOI: 10.48550/ARXIV.2302.03202. URL: https://arxiv.org/abs/2302.03202.

[18] Divyansh Jhunjhunwala et al. *FedVARP: Tackling the Variance Due to Partial Client Participation in Federated Learning*. 2022. arXiv: 2207.14130 [cs.LG].

[19] Xisen Jin et al. *Dataless Knowledge Fusion by Merging Weights of Language Models*. 2023. arXiv: 2212.09849 [cs.CL].

[20] Jeevesh Juneja et al. *Linear Connectivity Reveals Generalization Strategies*. 2022. DOI: 10.48550/ARXIV.2205.12411. URL: https://arxiv.org/abs/2205.12411.

[21] Peter Kairouz et al. *Advances and Open Problems in Federated Learning*. 2019. DOI: 10.48550/ARXIV.1912.04977. URL: https://arxiv.org/abs/1912.04977.

[22] Jared Kaplan et al. *Scaling Laws for Neural Language Models*. 2020. arXiv: 2001.08361 [cs.LG].

[23] Katherine Lee et al. *Deduplicating Training Data Makes Language Models Better*. 2022. arXiv: 2107.06499 [cs.CL].

[24] Mike Lewis et al. *BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension*. 2019. arXiv: 1910.13461 [cs.CL].

[25] Margaret Li et al. "Branch-train-merge: Embarrassingly parallel training of expert language models". In: *arXiv preprint arXiv:2208.03306* (2022).

[26] Davis Liang et al. *XLM-V: Overcoming the Vocabulary Bottleneck in Multilingual Masked Language Models*. 2023. arXiv: 2301.10472 [cs.CL].

[27] Yinhan Liu et al. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. 2019. arXiv: 1907.11692 [cs.CL].

[28] Grigory Malinovsky et al. *Federated Learning with Regularized Client Participation*. 2023. arXiv: 2302.03662 [cs.LG].

[29] Michael Matena and Colin Raffel. *Merging Models with Fisher-Weighted Averaging*. 2022. arXiv: 2111.09832 [cs.LG].

[30] H. Brendan McMahan et al. "Communication-Efficient Learning of Deep Networks from Decentralized Data". In: (2016). DOI: 10.48550/ARXIV.1602.05629. URL: https://arxiv.org/abs/1602.05629.

[31] Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. "What is being transferred in transfer learning?" In: (2020). DOI: 10.48550/ARXIV.2008.11687. URL: https://arxiv.org/abs/2008.11687.

[32] John Nguyen et al. *Where to Begin? Exploring the Impact of Pre-Training and Initialization in Federated Learning*. 2022. DOI: 10.48550/ARXIV.2206.15387. URL: https://arxiv.org/abs/2206.15387.

[33] D. Opitz and R. Maclin. "Popular Ensemble Methods: An Empirical Study". In: *Journal of Artificial Intelligence Research* 11 (Aug. 1999), pp. 169–198. DOI: 10.1613/jair.614. URL: https://doi.org/10.1613%2Fjair.614.

[34] Telmo J. P. Pires, Eva Schlinger, and Dan Garrette. "How Multilingual is Multilingual BERT?" In: *Annual Meeting of the Association for Computational Linguistics*. 2019.

[35] Clifton Poth et al. *What to Pre-Train on? Efficient Intermediate Task Selection*. 2021. arXiv: 2104.08247 [cs.CL].

[36] Yada Pruksachatkun et al. "Intermediate-Task Transfer Learning with Pretrained Language Models: When and Why Does It Work?" In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 5231–5247. DOI: 10.18653/v1/2020.acl-main.467. URL: https://aclanthology.org/2020.acl-main.467.

[37] Yujia Qin et al. *Exploring Mode Connectivity for Pre-trained Language Models*. 2022. arXiv: 2210.14102 [cs.CL].

[38] Liangqiong Qu et al. *Rethinking Architecture Design for Tackling Data Heterogeneity in Federated Learning*. 2022. arXiv: 2106.06047 [cs.LG].

[39] Jack W. Rae et al. *Scaling Language Models: Methods, Analysis Insights from Training Gopher*. 2022. arXiv: 2112.11446 [cs.CL].

[40] Lior Rokach. "Ensemble-based classifiers". In: *Artificial intelligence review* 33 (2010), pp. 1–39.

[41] Victor Sanh et al. *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*. 2020. arXiv: 1910.01108 [cs.CL].

[42] Shaden Smith et al. *Using DeepSpeed and Megatron to Train Megatron-Turing NLG 530B, A Large-Scale Generative Language Model*. 2022. arXiv: 2201.11990 [cs.CL].

[43] Haoyu Wang et al. "FedKC: Federated Knowledge Composition for Multilingual Natural Language Understanding". In: *Proceedings of the ACM Web Conference 2022*. WWW '22. Virtual Event, Lyon, France: Association for Computing Machinery, 2022, pp. 1839–1850. ISBN: 9781450390965. DOI: 10.1145/3485447.3511988. URL: https://doi.org/10.1145/3485447.3511988.

[44] Jing Wang, Mayank Kulkarni, and Daniel Preotiuc-Pietro. "Multi-Domain Named Entity Recognition with Genre-Aware and Agnostic Inference". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 8476–8488. DOI: 10.18653/v1/2020.acl-main.750. URL: https://aclanthology.org/2020.acl-main.750.

[45] Orion Weller, Kevin Seppi, and Matt Gardner. *When to Use Multi-Task Learning vs Intermediate Fine-Tuning for Pre-Trained Encoder Transfer Learning*. 2022. arXiv: 2205.08124 [cs.CL].

[46] Mitchell Wortsman et al. "Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time". In: *International Conference on Machine Learning*. PMLR. 2022, pp. 23965–23998.

[47] Haibo Yang, Minghong Fang, and Jia Liu. *Achieving Linear Speedup with Partial Worker Participation in Non-IID Federated Learning*. 2021. arXiv: 2101.11203 [cs.LG].

[48] Bo Zheng et al. *Allocating Large Vocabulary Capacity for Cross-lingual Language Model Pre-training*. 2021. arXiv: 2109.07306 [cs.CL].

A Experimental Setups

Here we provide additional details of our experimental setups. The code is implemented by PyTorch and the overall code structure is based on Flower [2] library with some modifications. We report the final round score for parameter-averaged model (both iid and non-iid setting) and the highest score for centralized model and calculate accuracy gain based on this score.

A.1 Tiny BERT

To investigate how parameter-averaging affects performance of model with notably limited representation power, we trained extremely small version of BERT, TINY-BERT, which consists of 2 layers, a hidden dimension size of 128, 2 attention heads and an intermediate dimension size of 512. We follow the hyperparameters proposed by crammed BERT with the exception of training a batch size of 256 and using post layer normalization with learning rate 1e-4[12]. Pretraining TINY-BERT takes around 1 day on 4 A5000(24GB) GPUs.

A.2 Experiment Configuration on Section 5

During partial averaging, some contributors may not be participated in parameter-averaging process since contributors are chosen randomly in each communication round. In our experiment, only 9 languages participate in parameter-averaging process and we count the number of access to each language. The result is as follows:

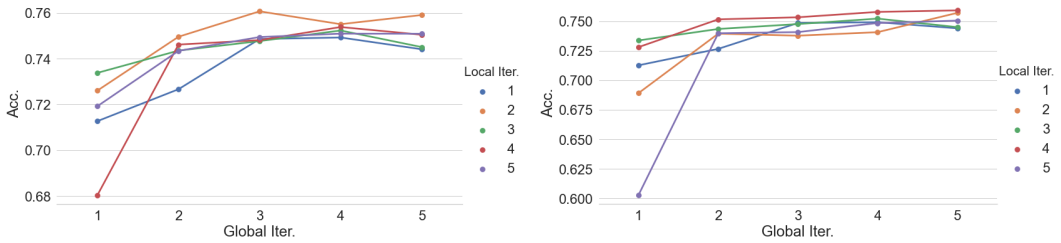
Table 5: The number of each contributors participate in parameter-averaging process

Language	en	fr	de	ru	tr	th	hi	sw	ur
Count	1	2	1	2	3	1	2	1	2
Total iter.	2	4	2	4	6	2	4	2	4

For fair comparison, we decided to fine-tuned the Central(*partial*) model which trained on 9 seen languages for 4 epochs to make sure that total number of iteration is larger than Non-IID(*partial*). We reported the best scores among 4 epochs.

B Hyperparameter Details

We offer an explanation for conducting experiments with contributors trained for 2 epochs and 5 rounds of parameter-averaging. We conducted 50 experiments where we varied the number of training epoch of contributors and parameter-averaging rounds from 1 to 5 on iid and non-iid setting.



(a) IID (b) Non-IID

Figure 6: Accuracy graph by different training epochs and parameter averaging round

Parameter-averaging round As depicted in Figure 6, we reach a conclusion that single parameter-averaging is not enough. There is a noticeable enhancement in performance as the number of parameter-averaging rounds increases, particularly when comparing parameter-averaging round 1 and 2. Furthermore, we only observed marginal improvements beyond a parameter-averaging round of 5. Table 6 reports specific score for each parameter-averaging round.

Training epochs of contributors Among all combinations of the training epochs and parameter averaging rounds, we found out that the combination of local 2 and global 5 is the most robust and efficient choice.

Table 6: Accuracy for different combinations of training epochs of contributors and parameter-averaging round. **Bold** scores indicate the best-performing results and underline indicate the second and third best result.

Traning epochs	Parameter-Avg round	Total iter.	IID Acc.	Non-IID Acc.
1	1	1	71.27	56.27
1	2	2	72.66	72.07
1	3	3	74.85	72.41
1	4	4	74.92	75.21
1	5	5	74.41	74.36
2	1	2	72.6	68.92
2	2	4	74.95	73.95
2	3	6	**76.06**	73.78
2	4	8	75.5	74.07
2	5	10	75.9	75.71
3	1	3	73.38	72.45
3	2	6	74.35	73.67
3	3	9	74.76	74.23
3	4	12	75.22	74.96
3	5	15	74.5	74.76
4	1	4	68.04	72.81
4	2	8	74.61	75.16
4	3	12	74.8	75.34
4	4	16	75.38	75.79
4	5	20	75.04	**75.92**
5	1	5	71.92	60.29
5	2	10	74.34	73.98
5	3	15	74.94	74.08
5	4	20	75.09	74.85
5	5	25	75.08	75.05

C MNLI

C.1 Full Parameter-Averaging Result

Table 7: Accuracy on MNLI dataset. All available contributors participate in parameter-averaging process. We use 3 different seeds and report the average. The value in the parenthesis is accuracy gain score.

Model	Type	Method	MNLI					Total Avg.
			Fiction	Government	Slate	Telephone	Travel	
TINY-BERT	Monolingual	Central	62.7	63.85	56.12	62.79	56.07	60.30
		IID(100%)	58.03	59.2	54.05	60.01	54.0	57.05 (-3.24)
		Non-IID(100%)	57.86	59.88	52.96	57.06	52.89	56.12 (-4.17)
DISTILBERT	Monolingual	Central	78.38	80.62	75.24	79.5	75.25	77.79
		IID(100%)	77.34	80.87	74.45	78.55	74.46	77.13 (-0.66)
		Non-IID(100%)	76.97	81.23	75.4	77.76	75.5	77.37 (-0.42)
BERT-BASE	Monolingual	Central	80.71	82.54	78.44	81.0	78.43	80.22
		IID(100%)	80.43	83.26	77.74	80.65	77.74	79.96 (-0.26)
		Non-IID(100%)	80.2	83.5	78.53	80.57	78.53	80.26 (+0.04)
BERT-LARGE	Monolingual	Central	84.52	84.0	80.82	82.42	80.82	82.52
		IID(100%)	84.6	84.95	82.0	84.15	82.0	83.53 (+1.02)
		Non-IID(100%)	84.25	85.02	82.3	83.38	82.3	83.45 (+0.93)
ROBERTA-BASE	Monolingual	Central	84.25	85.15	82.3	86.05	82.3	84.01
		IID(100%)	86.3	86.35	84.15	87.25	84.15	85.64 (+1.63)
		Non-IID(100%)	86.05	86.6	83.95	86.2	83.95	85.35 (+1.34)
ROBERTA-LARGE	Monolingual	Central	87.92	87.3	85.88	88.57	85.88	87.11
		IID(100%)	89.48	88.7	88.1	89.7	88.1	88.81 (+1.70)
		Non-IID(100%)	89.35	88.38	87.75	89.45	87.75	88.53 (+1.42)
DISTILMBERT	Multilingual	Central	72.62	77.52	71.17	76.5	71.17	73.79
		IID(100%)	73.63	77.25	70.4	75.67	70.4	73.47 (-0.32)
		Non-IID(100%)	72.15	77.6	71.58	75.08	71.58	73.6 (-0.19)
MBERT	Multilingual	Central	76.38	81.12	75.38	78.85	75.38	77.42
		IID(100%)	77.42	80.1	75.28	79.58	75.28	77.53 (+0.11)
		Non-IID(100%)	77.03	80.62	75.97	79.35	75.97	77.79 (+0.37)

C.2 Partial Averaging Result

Table 8: Accuracy on MNLI dataset. Only 40% of contributors participate in parameter-averaging process. We use 3 different seeds and report the average. The value in the parenthesis is accuracy gain score.

Model	Method	MNLI					Total Avg.
		Fiction	**Government**	**Slate**	**Telephone**	**Travel**	
TINY-BERT	Central	62.70	63.85	56.12	62.79	56.07	60.30
	IID(40%)	58.5	59.51	53.97	58.85	53.93	56.95 (-3.35)
	Non-IID(40%)	54.37	55.21	50.34	55.19	50.35	53.09 (-7.21)
DISTILBERT	Central	78.38	80.62	75.24	79.5	75.25	77.79
	IID(40%)	77.5	80.43	74.43	79.07	74.37	77.15(-0.63)
	Non-IID(40%)	76.51	80.19	74.08	77.02	73.97	76.35(-1.44)
BERT-BASE	Central	80.71	82.54	78.44	81.0	78.43	80.22
	IID(40%)	80.52	82.88	77.72	80.1	77.72	79.78 (-0.43)
	Non-IID(40%)	79.48	82.65	77.95	80.32	77.95	79.66 (-0.55)
BERT-LARGE	Central	84.52	84.00	80.82	82.42	80.82	82.52
	IID(40%)	83.78	85.10	81.70	83.45	81.70	83.14 (+0.62)
	Non-IID(40%)	84.60	84.12	81.78	83.65	81.78	83.18 (+0.66)
ROBERTA-BASE	Central	84.25	85.15	82.30	86.05	82.30	84.01
	IID(40%)	84.75	86.70	83.60	85.80	83.60	84.88 (+0.88)
	Non-IID(40%)	83.50	85.65	82.75	85.60	82.75	84.05 (+0.04)
ROBERTA-LARGE	Central	87.92	87.30	85.88	88.57	85.88	87.11
	IID(40%)	88.22	88.55	86.55	89.9	86.55	87.95 (+0.85)
	Non-IID(40%)	88.95	88.5	87.00	89.12	87.00	88.11 (+1.00)

D XNLI

D.1 Full parameter-Averaging Result

Table 9: Accuracy on XNLI dataset. All available contributors participate in parameter-averaging process. We use 3 different seeds and report the average. The value in the parenthesis is accuracy gain score.

Model	Method	XNLI															Total Avg.
		en	fr	es	de	el	bg	ru	tr	ar	vi	th	zh	hi	sw	ur	
DISTILMBERT	Central	75.44	70.73	71.85	69.5	67.55	70.21	68.01	66.01	65.19	68.56	54.21	69.79	62.15	60.42	60.18	66.65
	IID(100%)	73.56	69.41	69.39	67.72	65.71	67.69	66.52	64.18	62.75	66.59	53.17	68.1	60.38	58.07	59.27	64.83 (-1.82)
	Non-IID(100%)	73.83	70.46	71.06	68.55	64.59	67.48	66.56	62.4	62.18	67.22	48.44	67.27	59.49	53.71	58.85	64.13 (-2.51)
MBERT	Central	78.22	73.98	75.02	72.89	70.17	72.71	71.43	68.85	67.87	72.28	56.48	72.66	65.6	63.15	62.41	69.58
	IID(100%)	77.84	73.36	74.58	72.19	69.89	72.61	71.25	68.49	67.18	71.13	56.95	71.71	65.06	61.38	61.51	69.00 (-0.57)
	Non-IID(100%)	79.24	74.72	75.95	73.22	70.51	72.66	72.05	67.54	67.11	72.3	55.09	71.28	64.83	57.66	61.77	69.06 (-0.51)
XLMR-BASE	Central	81.95	77.61	78.18	76.13	76.05	77.67	75.83	72.99	72.91	76.39	72.69	75.92	71.45	68.44	67.34	74.76
	IID(100%)	83.47	78.48	79.32	77.61	77.83	78.51	76.92	74.51	74.33	77.14	74.76	76.61	72.44	69.63	69.38	76.06 (+1.29)
	Non-IID(100%)	83.95	79.15	80.14	78.62	78.2	79.15	77.49	75.14	74.69	77.63	73.5	75.93	72.55	68.91	69.67	76.31 (+1.54)
XLMR-LARGE	Central	87.13	82.38	83.88	82.77	82.18	83.14	80.58	79.3	78.98	80.46	77.79	80.39	77.63	73.16	73.39	80.21
	IID(100%)	88.55	84.21	85.62	84.78	84.03	84.86	81.92	81.27	81.25	82.17	79.12	81.9	79.66	76.54	76.06	82.12 (+1.91)
	Non-IID(100%)	89.15	84.57	85.64	84.94	84.04	84.77	81.92	81.24	81.44	81.87	77.83	80.94	78.96	75.56	75.67	81.90 (+1.69)
XLMV-BASE	Central	82.48	77.97	79.27	78.06	77.69	78.65	76.16	75.08	73.56	76.34	72.74	76.58	72.08	70.25	69.61	75.76
	IID(100%)	84.03	79.34	79.96	79.35	78.62	79.76	77.9	76.11	74.22	77.33	75.11	77.72	74.52	71.89	71.82	77.17 (+1.41)
	Non-IID(100%)	84.3	79.72	80.51	79.59	78.67	80.04	78.43	76.35	74.49	77.61	73.67	76.82	73.79	71.54	71.25	77.11 (+1.35)

D.2 Partial Averaging Result

Table 10: Accuracy on XNLI dataset. Only 20% of contributors participate in parameter-averaging process. We use 3 different seeds and report the average. The value in the parenthesis is accuracy gain score.

Model	Method	XNLI															Total Avg.
		en	fr	es	de	el	bg	ru	tr	ar	vi	th	zh	hi	sw	ur	
DISTILMBERT	Central	75.44	70.73	71.85	69.50	67.55	70.21	68.01	66.01	65.19	68.56	54.21	69.79	62.15	60.42	60.18	66.65
	IID(20%)	72.77	68.74	69.4	67.13	65.18	67.95	65.46	63.79	62.2	65.92	52.48	67.02	60.22	57.66	59.16	64.33 (-2.31)
	NON-IID(20%)	73.11	69.7	69.48	68.33	64.36	65.53	66.02	63.37	60.87	67.69	49.07	66.5	59.88	51.56	58.53	63.60 (-3.05)
MBERT	Central	78.22	73.98	75.02	72.89	70.17	72.71	71.43	68.85	67.87	72.28	56.48	72.66	65.6	63.15	62.41	69.58
	IID(20%)	77.22	72.52	73.97	71.35	69.08	71.80	69.97	67.63	66.04	70.33	55.36	71.06	63.6	60.8	61.49	68.14 (-1.43)
	NON-IID(20%)	78.19	73.96	73.66	72.34	67.10	70.53	71.63	68.28	64.34	71.21	54.92	69.62	65.02	60.82	61.10	68.18 (-1.39)
XLMR-BASE	Central	81.95	77.61	78.18	76.13	76.05	77.67	75.83	72.99	72.91	76.39	72.69	75.92	71.45	68.44	67.34	74.76
	IID(20%)	83.16	78.33	79.11	77.56	76.96	78.55	76.84	73.92	74.02	76.74	73.96	76.66	72.28	68.8	68.72	75.70 (+0.94)
	NON-IID(20%)	83.07	78.25	78.87	77.99	77.27	78.29	76.33	74.50	73.22	76.74	72.49	74.81	72.1	67.69	69.34	75.39 (+0.63)
XLMR-LARGE	Central	87.13	82.38	83.88	82.77	82.18	83.14	80.58	79.3	78.98	80.46	77.79	80.39	77.63	73.16	73.39	80.21
	IID(20%)	88.24	83.3	84.73	83.84	83.24	84.08	81.33	80.62	80.44	81.42	78.42	81.18	78.68	75.27	74.8	81.30 (+1.10)
	NON-IID(20%)	88.46	83.94	84.81	84.25	83.54	83.91	81.48	80.93	80.27	81.18	77.55	80.14	78.37	73.80	74.56	81.14 (+0.93)