# Layer-Wise Adaptive Model Aggregation for Scalable Federated Learning

**Sunwoo Lee**[1,2], **Tuo Zhang**[1], **and Salman Avestimehr**[1]

[1]University of Southern California, USA
[2]Inha University, South Korea
{sunwool,tuozhang,avestime}@usc.edu

## Abstract

In Federated Learning (FL), a common approach for aggregating local solutions across clients is periodic full model averaging. It is, however, known that different layers of neural networks can have a different degree of model discrepancy across the clients. The conventional full aggregation scheme does not consider such a difference and synchronizes the whole model parameters at once, resulting in inefficient network bandwidth consumption. Aggregating the parameters that are similar across the clients does not make meaningful training progress while increasing the communication cost. We propose `FedLAMA`, a layer-wise adaptive model aggregation scheme for scalable FL. `FedLAMA` adjusts the aggregation interval in a layer-wise manner, jointly considering the model discrepancy and the communication cost. This fine-grained aggregation strategy enables to reduce the communication cost without significantly harming the model accuracy. Our extensive empirical study shows that, as the aggregation interval increases, `FedLAMA` shows a remarkably smaller accuracy drop than the periodic full aggregation, while achieving comparable communication efficiency.

## Introduction

In Federated Learning (FL), periodic full model averaging is the most common approach for aggregating local solutions. As the aggregation interval increases, however, the periodic full aggregations cause a high degree of model discrepancy across clients, and it results in a slower convergence. It has recently been observed that the magnitude of gradients can be largely different across neural network layers (You et al. 2019). That is, all the layers can diverge across the clients in a difference pace. The conventional periodic full aggregation scheme does not consider such a difference and always synchronizes the full model at once. Averaging the parameters that are similar across all the clients does not make meaningful training progress while consuming the network bandwidth. Considering the limited network bandwidth in usual FL environments, such inefficient model aggregations can crucially harm the scalability of FL applications.

Most of the FL strategies assume the underlying periodic full model averaging. A variety of federated optimizers, such as FedAvg (McMahan et al. 2017), FedProx (Li et al. 2018), FedNova (Wang et al. 2020), and SCAFFOLD (Karimireddy et al. 2020), periodically average the full local solutions across all the clients. Many communication-efficient FL strategies, such as gradient (model) sparsification (Wangni et al. 2017; Wang et al. 2018; Alistarh et al. 2018), low-rank approximation (Vogels, Karimireddy, and Jaggi 2020; Wang, Agarwal, and Papailiopoulos 2021), and quantization (Alistarh et al. 2017; Wen et al. 2017; Albasyoni et al. 2020; Reisizadeh et al. 2020) techniques, also periodically aggregate the compressed form of the full local solutions. Adaptive model aggregation techniques (Wang and Joshi 2018a; Haddadpour et al. 2019) adjust the aggregation interval at run-time to reduce the total number of communications, however, they still aggregate the full local models at once. Because all the above FL strategies are based on the periodic full model aggregation scheme which ignores the layer-wise data characteristics, they can rapidly lose the accuracy as the aggregation interval increases.

In this paper, we propose `FedLAMA`, a Layer-wise Adaptive Model Aggregation scheme for FL. Our study breaks the convention of periodic full aggregation and introduces a novel and efficient model aggregation framework for scalable FL. `FedLAMA` first prioritizes all the layers based on their contributions to the total model discrepancy. We present a useful metric for estimating the layer-wise degree of model discrepancy at run-time. Then, the aggregation intervals are adjusted based on the assigned priorities such that the low priority layers are less frequently aggregated than the other layers. The layers are prioritized again after all the layers are aggregated at least once. Figure 1 shows schematic illustrations of the periodic full aggregation and `FedLAMA`. As the degree of model discrepancy at each layer increases, its color turns to red. We see that all the layers have a difference pace of the color change. Rather than having a uniform aggregation interval at all the layers, `FedLAMA` assigns a longer interval to the slower layers than the other layers, and it results in suppressing the total model discrepancy at the minimal communication cost.

Our study provides essential insights into how to better utilize the network bandwidth in FL. `FedLAMA` finds the layer-wise model aggregation interval settings, jointly considering the communication efficiency and the federated optimization efficiency. The fine-grained model aggregation strategy allows to spend more network bandwidth for the

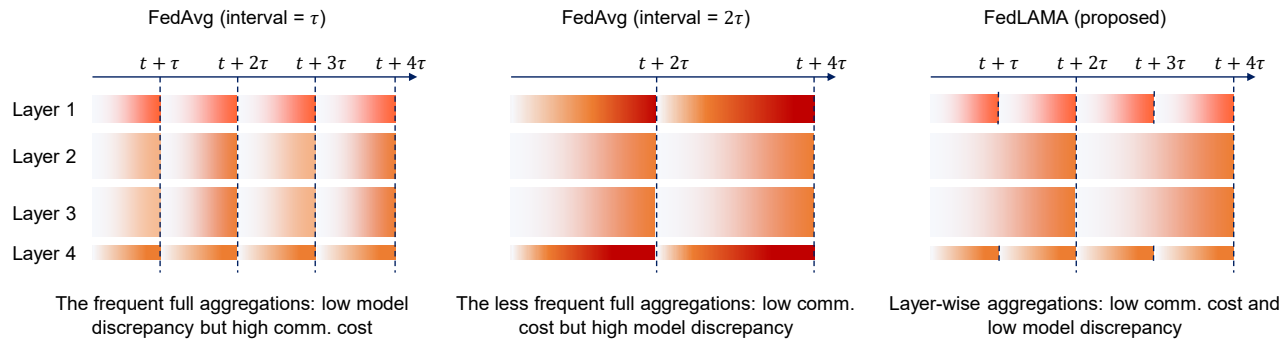


Figure 1: The comparison of FedAvg and FedLAMA. The dotted vertical lines indicate the model aggregation. The gradient color of each layer indicates the degree of model discrepancy across clients. The longer the aggregation interval, the higher the degree of model discrepancy (the slower convergence). FedLAMA adaptively controls the layer-wise model discrepancy.

critical layers that strongly contribute to the model discrepancy while relaxing the aggregation frequency at the other less critical layers. Thus, when consuming a similar amount of network bandwidth as the periodic full aggregation, FedLAMA yields a remarkably reduced degree of model discrepancy, making the global loss converge faster. We also show that FedLAMA provides a solid convergence guarantee for non-convex smooth problems under non-IID data settings.

We evaluate the performance of FedLAMA using three representative benchmark datasets: CIFAR-10 (Krizhevsky 2009), CIFAR-100, and Federated Extended MNIST (Cohen et al. 2017). We first compare FedLAMA to the conventional periodic full model aggregation scheme to demonstrate its superior convergence properties. We also compare FedLAMA to the state-of-the-art communication-efficient FL strategies, FedPAQ (Reisizadeh et al. 2020) and FedCOM (Haddadpour et al. 2021), to demonstrate how efficiently it consumes the network bandwidth. Finally, we also show that FedLAMA can be applied to FL together with other communication-efficient FL strategies. This result shows that FedLAMA is rather complementary to other FL strategies.

## Related Work

**Communication-Efficient FL Algorithms** – Many recent works proposed sparsification and quantization methods specifically designed for FL (Alistarh et al. 2017, 2018; Albasyoni et al. 2020; Wangni et al. 2017; Wang et al. 2018; Wang, Agarwal, and Papailiopoulos 2021; Alistarh et al. 2017; Wen et al. 2017; Reisizadeh et al. 2020). These methods are also called *sketched* approach (Konečnỳ et al. 2016). Another set of communication-efficient FL strategies are low-rank model approximation (Phan et al. 2020; Vogels, Karimireddy, and Jaggi 2020; Yao et al. 2021; Hyeon-Woo, Ye-Bin, and Oh 2021). These are called *structured* approach.

Although top-k sparsification is widely used in distributed learning (Wangni et al. 2017; Alistarh et al. 2018; Bibikar et al. 2021), it is not directly applicable to cross-edge FL. Since residuals, the small accumulated updates, should remain at the client-side until they sufficiently grow up, they become out-of-date if the device is not activated at a communication round. Due to the limitation, only cross-silo settings are considered in (Bibikar et al. 2021). FedPAQ (Reisizadeh et al. 2020) and FedCOM (Haddadpour et al. 2021) are the state-of-the-art quantization-based FL strategies. This local update compression approach still assumes the periodic full model aggregation scheme, and thus they sharply lose the accuracy as the aggregation interval increases.

**Layer-wise Model Freezing** – Layer freezing (dropping) is a representative layer-wise technique for neural network training (Brock et al. 2017; Kumar et al. 2019; Zhang and He 2020; Goutam et al. 2020). All these methods commonly stop updating the network layers in a bottom-up direction. These techniques are supported by the analysis presented in (Raghu et al. 2017). Since the layers converge from the input-side to the output-side sequentially, the layer-wise freezing can reduce the training time without strongly affecting the accuracy. These previous works clearly show the advantages of processing individual layers separately.

## Background

**Federated Optimization** – We consider federated optimization problems as follows.

$$\min_{\mathbf{x}\in\mathbb{R}^d} \left[ F(\mathbf{x}) := \frac{1}{m}\sum_{i=1}^{m} F_i(\mathbf{x}) \right], \tag{1}$$

where $m$ is the number of local models and $F_i(\mathbf{x}) = \mathbb{E}_{\xi_i\sim D_i}[F_i(\mathbf{x},\xi_i)]$ is the local objective function associated with local data distribution $D_i$.

FedAvg (McMahan et al. 2017) is a basic algorithm that solves the above minimization problem. As the degree of data heterogeneity increases, FedAvg converges more slowly. Several variants, such as FedProx (Li et al. 2018), FedNova (Wang et al. 2020), and SCAFFOLD (Karimireddy et al. 2020), tackle the data heterogeneity issue.

**Model Discrepancy** – All local SGD-based FL algorithms allow the clients to independently run SGD steps within each communication round. Thus, the variance of stochastic gradients and heterogeneous data lead the local models to different directions on the parameter space. We formally define

such a model discrepancy as follows.

$$\frac{1}{m}\sum_{i=1}^{m}\|\mathbf{u}-\mathbf{x}^i\|^2,$$

where $m$ is the number of local models (clients), $\mathbf{u}$ is the synchronized model, and $\mathbf{x}^i$ is client $i$'s local model. This quantity bounds the difference between the local gradients and the global gradients under a smoothness assumption on objective functions. Even when the data is independent and identically distributed (IID), there always exist a certain degree of model discrepancy due to the gradient variance.

## Layer-wise Adaptive Model Aggregation

In this section, we first discuss how to prioritize the network layers jointly considering the communication cost and the model discrepancy. Then, we describe a layer-wise adaptive model aggregation algorithm built upon the proposed layer prioritization strategy. Finally, we present a general FL framework built upon our model aggregation method.

### Layer Prioritization

In theoretical convergence analysis, the distance between the local models and the global model is commonly used to bound the difference between their gradients. Specifically, it is assumed that the difference between the local gradient and the global gradient is bounded by the distance between the corresponding sets of model parameters under an assumption on smoothness of the objective function. Motivated by this convention, we define 'layer-wise unit model discrepancy', a useful metric for prioritizing the layers as follows.

$$d_l = \frac{\frac{1}{m}\sum_{i=1}^{m}\|\mathbf{u}_l-\mathbf{x}_l^i\|^2}{\tau_l * \dim(\mathbf{u}_l)}, \qquad l \in \{1,\cdots,L\} \tag{2}$$

where $L$ is the number of layers, $\mathbf{u}$ is the global model, $\mathbf{x}^i$ is the client $i$'s local model, $\tau$ is the aggregation interval.

The communication cost is proportional to the number of parameters. Thus, $\frac{1}{m}\sum_{i=1}^{m}\|\mathbf{u}_l-\mathbf{x}_l^i\|^2/\dim(\mathbf{u}_l)$ shows how much model discrepancy can be eliminated by synchronizing the layer at a unit communication cost. We assign a high priority to the layers with a large $d_l$ value.

### Layer-Wise Aggregation Interval Adjustment

We consider selecting a subset of network layers and relaxing their synchronization frequency to reduce the communication cost in Federated Learning. As increasing the aggregation interval at more layers, the communication cost is proportionally reduced while having a higher degree of model discrepancy. Thus, we focus on finding the layers that *least* increase the model discrepancy while *most* reducing the communication cost.

To quantify such criteria, we define accumulative model discrepancy ratio $\delta_l$ and model partition ratio $\lambda_l$. First, the accumulative model discrepancy ratio is defined based on the unit model discrepancy (Eq. 2) as follows.

$$\delta_l = \frac{\sum_{k=1}^{l} d_{\mathbf{I}[k]} * \dim(\mathbf{u}_{\mathbf{I}[k]})}{\sum_{k=1}^{L} d_{\mathbf{I}[k]} * \dim(\mathbf{u}_{\mathbf{I}[k]})}, \tag{3}$$

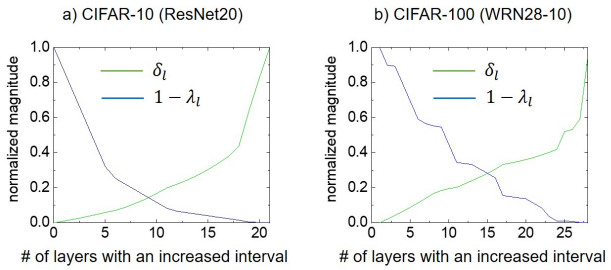


Figure 2: The comparison between the model discrepancy ratio $\delta_l$ and the model partition ratio $1-\lambda_l$ for a) CIFAR-10 and b) CIFAR-100 training.

where $\mathbf{I}$ is a list of layer indices sorted based on $d_l$. Given $l$ layers with the smallest $d_l$ values, $\delta_l$ quantifies their contribution to the total model discrepancy. Second, the model partition ratio is defined as follows.

$$\lambda_l = \frac{\sum_{k=1}^{l} \dim(\mathbf{u}_{\mathbf{I}[k]})}{\sum_{k=1}^{L} \dim(\mathbf{u}_{\mathbf{I}[k]})}. \tag{4}$$

Because both $\delta_l$ and $\lambda_l$ are a ratio which lies between 0 and 1, we now can directly compare these two independent criteria and find a sweat spot where the model discrepancy is *least* increased while *most* reducing the communication cost. Specifically, we find the $l$ value which makes $\delta_l$ and $1-\lambda_l$ as close as possible. This condition guarantees that, when the aggregation interval is increased at the chosen $l$ layers, the relative impact on the model discrepancy and the communication cost are almost the same. Because we calculate $\delta_l$ and $\lambda_l$ using $\mathbf{I}$, the layer indices sorted based on the unit model discrepancy, we can expect the minimal increase of the model discrepancy and the maximal decrease of the communication cost.

Figure 2 shows the actual $\delta_l$ and $\lambda_l$ values measured from CIFAR-10 (ResNet20) and CIFAR-100 (Wide-ResNet28-10) training. The x-axis in the charts is the $l$ value that appears in Eq. 3 and 4. We see that the y-axis value of the cross point is much smaller than 0.5 in both charts. This means that, when increasing the aggregation interval at the $l$ layers with the smallest $d_l$ values, we can expect the minimal increase of the model discrepancy and the maximal decrease of the communication cost.

Algorithm 1 shows the described aggregation interval adjustment algorithm. The algorithm first obtains $\mathbf{I}$, a list of layer indices that would sort $\mathbf{d}$ in an increasing order (*argsort*). Then, looping over the elements in $\mathbf{d}$ following the order of $\mathbf{I}$, the algorithm calculates $\lambda_l$ and $\delta_l$. If $\delta_l < (1-\lambda_l)$, the algorithm sets the aggregation interval at layer $\mathbf{I}[l]$ to $\phi\tau'$. Otherwise, the aggregation interval is set to $\tau'$. $\phi$ is the interval increase factor, a user-tunable hyper-parameter.

### Framework

Now we build a general FL framework `FedLAMA` upon the proposed layer-wise adaptive model aggregation scheme. Algorithm 2 shows the framework. The algorithm receives two hyper-parameters, the base interval $\tau'$ and the interval

Algorithm 1: Layer-wise Aggregation Interval Adjustment.

**Input**: $\mathbf{d}$: the observed model discrepancy at all $L$ layers,
$\tau'$: the base aggregation interval,
$\phi$: the interval increase factor
$\mathbf{I} \leftarrow$ argsort ($\mathbf{d}$)
**for** $l = 1$ to $L$ **do**
  The layer index $j \leftarrow \mathbf{I}[l]$
  $\delta_l \leftarrow$ Equation 3
  $\lambda_l \leftarrow$ Equation 4
  **if** $\delta_l < (1 - \lambda_l)$ **then**
    $\tau_j \leftarrow \phi\tau'$
  **else**
    $\tau_j \leftarrow \tau'$
  **end if**
**end for**
**Output:** $\tau$: the new aggregation intervals at all $L$ layers

Algorithm 2: FedLAMA: Federated Layer-wise Adaptive Model Aggregation.

**Input**: $\tau'$: base interval, $\phi$: interval increase factor
$\tau_l \leftarrow \tau', \forall l \in \{1, \cdots, L\}$
$\tau \leftarrow \tau'\phi$
**for** $t = 0$ to $T - 1$ **do**
  **for** $j = 0$ to $\tau$ **do**
    Local SGD step: $\mathbf{x}^i_{t,j+1} = \mathbf{x}^i_{t,j} - \eta\nabla f(\mathbf{x}^i_{t,j}, \xi_i)$
    **for** $l = 1$ to $L$ **do**
      **if** $j$ mod $\tau_l$ is 0 **then**
        Synchronize layer $l$: $\mathbf{u}_t \leftarrow \frac{1}{m}\sum_{i=1}^m \mathbf{x}^i_{t,j}$
        $d_l \leftarrow$ Equation 2
        Update layer $l$ of the local model: $\mathbf{x}^i_{t,j} = \mathbf{u}_t$
      **end if**
    **end for**
  **end for**
  Adjust the aggregation interval at all $L$ layers (Alg. 1)
**end for**
**Output:** $\mathbf{u}_T$

increase factor $\phi$. The network layers are assigned with an aggregation interval of either $\tau'$ or $\tau'\phi$ depending on their contribution to the model discrepancy. Whenever a layer is synchronized, the unit model discrepancy $d_l$ is updated for the next adjustment. After the whole model is synchronized once, the algorithm re-adjusts the layer-wise aggregation intervals using the observed unit model discrepancy values.

**Impact of Interval Increase Factor $\phi$** – The communication latency cost is usually not negligible in FL, and the total number of communications strongly affects the scalability. Algorithm 1 increases the aggregation interval at a few selected layers by multiplying $\phi$ to their base interval $\tau'$. This approach ensures that the latency cost is not increased while the bandwidth consumption is reduced when increasing $\phi$.

FedAvg can be considered as a special case of FedLAMA where $\phi$ is set to 1. When $\phi > 1$, FedLAMA less frequently synchronizes the chosen layers, having a lower total communication cost. When increasing the aggregation interval, FedLAMA multiplies $\phi$ to the base interval $\tau'$. So, it is guaranteed that the whole model parameters are fully synchronized after every $\phi\tau'$ iterations. Because of the layers with the base aggregation interval $\tau'$, the total model discrepancy of FedLAMA after $\phi\tau'$ iterations is always smaller than that of FedAvg with the interval of $\phi\tau'$.

## Convergence Analysis

To demonstrate that FedLAMA is a robust model aggregation scheme for FL, we provide a convergence guarantee and analyze its properties. The proofs can be found in Appendix.

**Notations** – $\mathbf{x}^i_{t,j} \in \mathbb{R}^d$ denotes the client $i$'s local model at $j^{th}$ local step in $t^{th}$ communication round. The stochastic gradient computed from a single training data point $\xi_i$ is denoted by $\nabla F_i(\mathbf{x}^i_{t,j}, \xi_i)$. For convenience, we use $\mathbf{g}^i_{t,j}$ instead. The local full-batch gradient is denoted by $\nabla F_i(\cdot)$. We use $\|\cdot\|$ to denote $\ell_2$ norm.

**Assumptions** – Our analysis assumes the followings.

1. (Smoothness). Each local objective function is L-smooth, that is, $\|\nabla F_i(\mathbf{x}) - \nabla F_i(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|, \forall i \in \{1, \cdots, m\}$.
2. (Unbiased Gradient). The stochastic gradient at each client is an unbiased estimator of the local full-batch gradient: $\mathbb{E}_{\xi_i}\left[\mathbf{g}^i_{t,j}\right] = \nabla F_i(\mathbf{x}^i_{t,j})$.
3. (Bounded Variance). The gradient variance is bounded: $\mathbb{E}_{\xi_i}\left[\|\mathbf{g}^i_{t,j} - \nabla F_i(\mathbf{x}^i_{t,j})\|^2\right] \leq \sigma^2, \forall i \in \{1, \cdots, m\}$.
4. (Bounded Dissimilarity). There exist constants $\beta^2 \geq 1$ and $\kappa^2 \geq 0$ such that $\frac{1}{m}\sum_{i=1}^m \|\nabla F_i(\mathbf{x})\|^2 \leq \beta^2\|\frac{1}{m}\sum_{i=1}^m \nabla F_i(\mathbf{x})\|^2 + \kappa^2$. If local objective functions are identical to each other, $\beta^2 = 1$ and $\kappa^2 = 0$.

The assumption 4 considers the variance as well as the bias of the local gradients across the clients (Wang et al. 2020).

**Lemma 1.** *(framework) Under Assumption* $1 \sim 3$*, if the learning rate* $\eta \leq \frac{1}{L\tau}$*, Algorithm 1 ensures*

$$\sum_{t=0}^{T-1} \mathbb{E}\left[\|\nabla F(\mathbf{u}_t)\|^2\right] \leq \frac{2}{\eta\tau}\left(F(\mathbf{u}_0) - F(\mathbf{u}_{T-1})\right) + \frac{L\eta T}{m}\sigma^2 + \frac{L^2}{m\tau}\sum_{t=0}^{T-1}\sum_{j=1}^{\tau}\sum_{i=1}^{m}\mathbb{E}\left[\left\|\mathbf{x}^i_{t,j} - \mathbf{u}_t\right\|^2\right].$$

**Lemma 2.** *(model discrepancy) Under Assumption* $1 \sim 4$*, Algorithm 1 ensures*

$$\frac{1}{m}\sum_{j=0}^{\tau-1}\sum_{i=1}^{m}\mathbb{E}\left[\left\|\mathbf{x}^i_{t,j} - \mathbf{u}_t\right\|^2\right] \leq \frac{\eta^2\tau(\tau-1)}{1-A}\sigma^2 + \frac{\tau A\beta^2}{(1-A)L^2}\mathbb{E}\left[\|\nabla F(\mathbf{u}_t)\|^2\right] + \frac{\tau A\kappa^2}{(1-A)L^2}, \quad (5)$$

*where* $A \coloneqq 2\eta^2L^2\tau(\tau-1) < 1$.

Based on Lemma 1 and 2, we analyze the convergence rate of FedLAMA as follows.

**Theorem 1.** *Suppose all $m$ local models are initialized to the same point $\mathbf{u}_0$. Under Assumption $1 \sim 4$, if Algorithm 1 runs for $T$ communication rounds and the learning rate satisfies $\eta \leq \min\left\{\frac{1}{\tau L}, \frac{1}{L\sqrt{2\tau(\tau-1)(2\beta^2+1)}}\right\}$, the average-squared gradient norm of $\mathbf{u}_t$ is bounded as follows*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\left\|\nabla F(\mathbf{u}_t)\right\|^2\right] \leq \frac{4}{\eta\tau T}\left(F(\mathbf{u}_0) - F(\mathbf{u}_*)\right) + \frac{2L\eta}{m}\sigma^2 + 3L^2\eta^2(\tau-1)\sigma^2 + 6\eta^2L^2\tau(\tau-1)\kappa^2 \quad (6)$$

*where $\mathbf{u}_*$ indicates a local minimum and $\tau$ is the largest averaging interval across all the layers ($\tau'\phi$).*

**Remark 1.** *(Finite Horizon Result) With a sufficiently small diminishing learning rate and a large number of training iterations, FedLAMA achieves linear speedup. If the learning rate is $\eta = \frac{\sqrt{m}}{\sqrt{T}}$, we have*

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1}\left\|\nabla F(\mathbf{u}_t)\right\|^2\right] \leq \mathcal{O}\left(\frac{1}{\sqrt{mT}}\right) + \mathcal{O}\left(\frac{m}{T}\right). \quad (7)$$

*If $T > m^3$, the first term on the right-hand side becomes dominant and it achieves linear speedup. That is, FedLAMA has the same complexity of the convergence rate as FedAvg (Wang and Joshi 2018b; Yang, Fang, and Liu 2021).*

**Remark 2.** *(Impact of layer-wise aggregation) Theorem 1 considers the worst-case where only the low-priority layers contribute to the total model discrepancy. That is, if any other layers contribute to the model discrepancy, FedLAMA is guaranteed to converge faster than FedAvg with an aggregation interval of $\phi\tau'$. In practice, the local gradients most likely do not become zero at any layers, and thus every layer more or less contribute to the total model discrepancy.*

## Experiments

**Experimental Settings** – We evaluate FedLAMA using three representative benchmarks: CIFAR-10 (ResNet20 (He et al. 2016)), CIFAR-100 (WideResNet28-10 (Zagoruyko and Komodakis 2016)), and Federated Extended MNIST (CNN (Caldas et al. 2018)). We use TensorFlow 2.4.3 for local training and MPI for model aggregation. All our experiments are conducted on 8 NVIDIA A1000 GPUs.

Due to the limited compute resources, we simulate large-scale FL such that each process sequentially trains multiple models and then the models are globally averaged at once. Thus, instead of wall-clock time, we consider the total communication cost calculated as follows (Reisizadeh et al. 2020; Bibikar et al. 2021).

$$\mathcal{C} = \sum_{l=1}^{L}\mathcal{C}_l = \sum_{l=1}^{L}\dim(\mathbf{u}_l) * c_l, \quad (8)$$

where $c_l$ is the total number of communications at layer $l$.

**Non-IID Datasets** – We split CIFAR datasets to 128 subsets using Dirichlet distributions. The concentration coefficient is $0.1$ which represents a high-degree of non-IIDness. The distribution is generated with respect to the labels, and the local dataset sizes are all different (unbalanced non-IID). As for FEMNIST, we randomly select 256 writers' samples ($\sim 7.3\%$ of the whole samples) and assign 2 writers' samples to each client. In all the experiments, random $25\%$ of 128 clients run each communication round.

### Comparison to Periodic Full Aggregation Scheme

We compare FedLAMA's classification accuracy to that of the periodic full aggregation. Table 1, 2, and 3 show the results of CIFAR-10, CIFAR-100, and FEMNIST, respectively. When increasing the aggregation interval, FedAvg increases the base interval $\tau'$ while FedLAMA increases the increase factor $\phi$. FedLAMA has two training results, early stopping and full training. In the early stopping, FedLAMA stops the training once it achieves a higher accuracy than FedAvg. In the full training, FedLAMA runs the maximum training step budget and choose the best achieved accuracy. The 'Comm. cost' column shows the relative communication cost as increasing the aggregation interval.

First, the early stopping results show that FedLAMA achieves a similar accuracy to FedAvg in significantly fewer training steps. This result proves that FedLAMA enables to increase the aggregation interval having a minimal impact on the model discrepancy. Second, the full training results show that, after the same number of training steps, FedLAMA achieves remarkably higher accuracy than FedAvg. While spending a comparable network bandwidth to FedAvg, FedLAMA converges faster, and it results in having higher accuracy.

### Comparison to Other FL Strategies

We also compare FedLAMA to the following FL strategies.

1. FedAvg (McMahan et al. 2017)
2. FedProx (Wang et al. 2020)
3. FedPAQ (Reisizadeh et al. 2020)
4. FedCOM (Haddadpour et al. 2021)

FedProx is a federated optimization algorithm that tackles the data heterogeneity issue by adding a proximal term to the objective function. FedPAQ is the state-of-the-art quantization framework designed for communication-efficient FL. FedCOM is another quantization framework that employs the server-side learning rate. Although these are not directly related to the aggregation settings, we believe that this comparison will deliver useful insights to the community.

We do not compare FedLAMA to model pruning methods such as FedDST (Bibikar et al. 2021) and PruneFL (Jiang et al. 2022) or model decomposition methods such as FedHM (Yao et al. 2021) and FedPara (Hyeon-Woo, Ye-Bin, and Oh 2021), because they significantly change the local model architecture affecting the computational cost. We also do not compare FedLAMA to the sparsification methods. Although top-k sparsification is used in many distributed learning studies (Wangni et al. 2017; Alistarh et al. 2018; Stich,

| FedAvg (Periodic Full Avg.) | | | | | FedLAMA | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Full training | | | | | Early stopping | | | | | | Full training | |
| LR | $\tau'$ | $\phi$ | Validation acc. | $C$ ratio | LR | $\tau'$ | $\phi$ | Validation acc. | # of steps | $C$ ratio | Validation acc. | $C$ ratio |
| 0.4 | 10 | 1 | $81.66 \pm 0.3\%$ | 100% | 0.4 | 10 | 1 | $81.66 \pm 0.3\%$ | 9,860 | 100% | $81.66 \pm 0.3\%$ | 100% |
| 0.3 | 20 | 1 | $72.99 \pm 0.5\%$ | 50% | 0.2 | 10 | 2 | $\mathbf{77.33} \pm 0.3\%$ | **5,160** | **32.01%** | $\mathbf{81.46} \pm 0.3\%$ | **61.55%** |
| 0.3 | 40 | 1 | $66.64 \pm 0.5\%$ | 25% | 0.2 | 10 | 4 | $\mathbf{68.32} \pm 0.4\%$ | **4,120** | **18.65%** | $\mathbf{80.60} \pm 0.4\%$ | **44.36%** |

Table 1: The CIFAR-10 (ResNet20) classification results. The total number of local steps is 10,000 and the local batch size is 32. The dataset is split based on a Dirichlet distribution ($\alpha = 0.1$) w.r.t the labels.

| FedAvg (Periodic Full Avg.) | | | | | FedLAMA | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Full training | | | | | Early stopping | | | | | | Full training | |
| LR | $\tau'$ | $\phi$ | Validation acc. | $C$ ratio | LR | $\tau'$ | $\phi$ | Validation acc. | # of steps | $C$ ratio | Validation acc. | $C$ ratio |
| 0.4 | 10 | 1 | $76.14 \pm 0.5\%$ | 100% | 0.4 | 10 | 1 | $76.14 \pm 0.5\%$ | 9,920 | 100% | $76.14 \pm 0.5\%$ | 100% |
| 0.2 | 20 | 1 | $70.64 \pm 0.4\%$ | 50% | 0.3 | 10 | 2 | $\mathbf{70.77} \pm 0.3\%$ | **6,280** | **39.13%** | $\mathbf{75.34} \pm 0.3\%$ | **62.32%** |
| 0.2 | 40 | 1 | $61.15 \pm 0.5\%$ | 25% | 0.2 | 10 | 4 | $\mathbf{61.86} \pm 0.4\%$ | **5,240** | **22.67%** | $\mathbf{71.93} \pm 0.3\%$ | **43.26%** |

Table 2: The CIFAR-100 (Wide-ResNet28-10) classification results. The total number of local steps is 10,000 and the local batch size is 32. The dataset is split based on a Dirichlet distribution ($\alpha = 0.1$) w.r.t. the labels.

| FedAvg (Periodic Full Avg.) | | | | | FedLAMA | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Full training | | | | | Early stopping | | | | | | Full training | |
| LR | $\tau'$ | $\phi$ | Validation acc. | $C$ ratio | LR | $\tau'$ | $\phi$ | Validation acc. | # of steps | $C$ ratio | Validation acc. | $C$ ratio |
| 0.02 | 20 | 1 | $81.57 \pm 0.3\%$ | 100% | 0.02 | 20 | 1 | $81.57 \pm 0.3\%$ | 3,980 | 100% | $81.57 \pm 0.3\%$ | 100% |
| 0.02 | 40 | 1 | $80.72 \pm 0.6\%$ | 50% | 0.06 | 20 | 2 | $\mathbf{80.74} \pm 0.3\%$ | **1,600** | **20.74%** | $\mathbf{82.33} \pm 0.2\%$ | **51.86%** |
| 0.02 | 80 | 1 | $80.48 \pm 0.3\%$ | 25% | 0.04 | 20 | 4 | $\mathbf{80.56} \pm 0.6\%$ | **1,360** | **9.76%** | $\mathbf{81.37} \pm 0.4\%$ | **28.51%** |
| 0.02 | 160 | 1 | $78.85 \pm 0.4\%$ | 12.5% | 0.04 | 20 | 8 | $\mathbf{79.18} \pm 0.3\%$ | **2,400** | **10.68%** | $\mathbf{80.62} \pm 0.5\%$ | **17.80%** |

Table 3: The FEMNIST (CNN) classification results. The total number of local steps is 4,000 and the local batch size is 32. Each client is assigned with two different writers' samples.

Cordonnier, and Jaggi 2018; Wang et al. 2018; Sattler et al. 2019), it potentially harms the FL accuracy. In cross-device settings, the residuals, the small accumulated local updates, can be significantly out-of-date if the client is not selected for many communication rounds, and it can make the global loss converge more slowly. Due to these limitations, only cross-silo settings are considered in (Bibikar et al. 2021).

We compare the learning curves among various FL strategies under the aggregation interval settings that provide a similar communication cost $C$ (Eq. 8). The settings are provided in Appendix. Figure 3 shows the FEMNIST (left) and CIFAR-10 (right) learning curves. For both benchmarks, FedLAMA achieves a higher accuracy than all the other FL strategies. This result shows that FedLAMA consumes the network bandwidth most efficiently. FedLAMA maintains a lower degree of model discrepancy during the training than the other FL strategies, and it results in making the best training progress within the same number of training steps.

## Harmonization with Other FL Strategies

Because FedLAMA is a standalone model aggregation scheme, it is complementary to other federated optimizers and compression methods. Table 4 shows the performance evaluation of FedProx + FedLAMA and FedPAQ + FedLAMA. Likely to the results in Table 3, FedLAMA achieves a similar accuracy to the periodic full model averaging in significantly fewer training steps. This result demonstrates that FedLAMA is indeed complementary to the state-of-the-art FL strategies. Especially, FedPAQ + FedLAMA results show that the communication cost can be extremely reduced by employing the compression method together with the proposed model aggregation framework.

## Communication Cost Analysis

We quantify the communication cost using the number of communications and the transferred data size. Figure 4 shows the total number of communications at the individual layers. For FedLAMA, $\tau'$ is 20 and $\phi$ is 2. We find that FedLAMA increases the aggregation interval mostly at the output-side large layers in all the benchmarks. This indicates that the observed $d_l$ value (Eq. 2) at the selected layers are smaller than that of the other layers. Figure 5 shows the layer-wise transferred data size (Eq. 8). Although the number of communications is reduced at few selected layers only, the total transferred data size is remarkably reduced because the selected layers take up most of the model size.

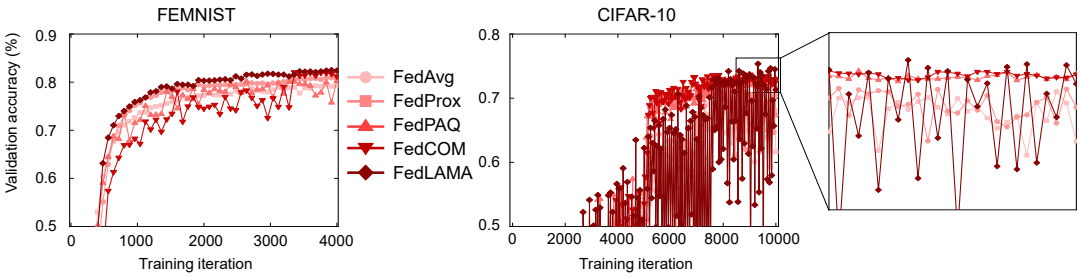


Figure 3: The FEMNIST (left) and CIFAR-10 (right) classification performance comparison among a variety of FL strategies.

| FedProx (Periodic Full Avg.) | | | | | FedProx + FedLAMA | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Full training | | | | | Early stopping | | | | | | Full training | |
| LR | $\tau'$ | $\phi$ | Validation acc. | $C$ ratio | LR | $\tau'$ | $\phi$ | Validation acc. | # of steps | $C$ ratio | Validation acc. | $C$ ratio |
| 0.04 | 20 | 1 | $82.24 \pm 0.2\%$ | 100% | 0.04 | 20 | 1 | $82.24 \pm 0.2\%$ | 3,920 | 100% | $82.24 \pm 0.2\%$ | 100% |
| 0.04 | 40 | 1 | $81.08 \pm 0.2\%$ | 50% | 0.04 | 20 | 2 | $\mathbf{81.55} \pm 0.1\%$ | **1,920** | **25.59%** | $\mathbf{82.02} \pm 0.2\%$ | **53.32%** |
| 0.04 | 80 | 1 | $80.62 \pm 0.2\%$ | 25% | 0.02 | 20 | 4 | $\mathbf{80.65} \pm 0.2\%$ | **1,840** | **13.79%** | $\mathbf{81.44} \pm 0.2\%$ | **29.97%** |
| 0.02 | 160 | 1 | $78.84 \pm 0.2\%$ | 12.5% | 0.02 | 20 | 8 | $\mathbf{78.94} \pm 0.5\%$ | **1,920** | **8.56%** | $\mathbf{80.21} \pm 0.2\%$ | **17.83%** |
| FedPAQ (Periodic Full Avg.) | | | | | FedPAQ + FedLAMA | | | | | | | |
| Full training | | | | | Early stopping | | | | | | Full training | |
| LR | $\tau'$ | $\phi$ | Validation acc. | $C$ ratio | LR | $\tau'$ | $\phi$ | Validation acc. | # of steps | $C$ ratio | Validation acc. | $C$ ratio |
| 0.02 | 20 | 1 | $80.22 \pm 0.4\%$ | 100% | 0.04 | 20 | 1 | $80.22 \pm 0.4\%$ | 3,900 | 100% | $80.22 \pm 0.4\%$ | 100% |
| 0.02 | 40 | 1 | $79.59 \pm 0.6\%$ | 50% | 0.03 | 20 | 2 | $\mathbf{79.60} \pm 0.2\%$ | **3,040** | **40.52%** | $\mathbf{80.07} \pm 0.2\%$ | **55.32%** |
| 0.02 | 80 | 1 | $75.47 \pm 0.5\%$ | 25% | 0.03 | 20 | 4 | $\mathbf{75.62} \pm 0.2\%$ | **1,440** | **10.61%** | $\mathbf{78.21} \pm 0.3\%$ | **29.48%** |
| 0.02 | 160 | 1 | $73.37 \pm 0.6\%$ | 12.5% | 0.03 | 20 | 8 | $\mathbf{74.19} \pm 0.2\%$ | **1,120** | **5.06%** | $\mathbf{77.95} \pm 0.3\%$ | **18.08%** |

Table 4: FEMNIST classification performance with FedLAMA and other FL techniques. The scaling factor of the proximal term for FedProx, $\mu$, is 0.001. The compression level for FedPAQ is 16.

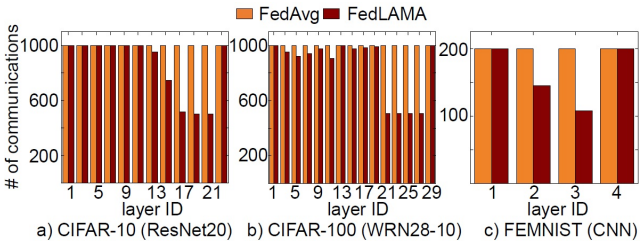


Figure 4: The number of communications at the individual layers, counted during the whole training.

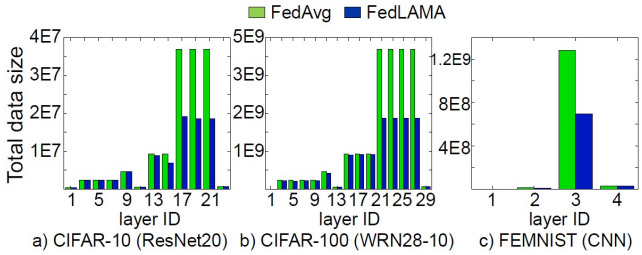


Figure 5: The total data size at all individual layers, that corresponds to Figure 4.

## Conclusion

We proposed FedLAMA, a layer-wise adaptive model aggregation scheme for scalable FL. FedLAMA saves the network bandwidth consumption by adaptively increasing the aggregation interval at less critical layers. Our study demonstrates that such a novel layer-wise model aggregation scheme achieves much higher accuracy within the same fixed epoch budget as compared to the periodic full model averaging scheme. We believe this result will introduce many unprecedented research directions on partial model aggregation. Harmonizing FedLAMA with other advanced optimizers, gradient compression, and low-rank approximation methods can be promising and important future work.

## Acknowledgments

This material is based upon work supported by Defense Advanced Research Projects Agency (DARPA) under Contract No. FASTNICS HR001120C0088 and gifts from Intel and Qualcomm. The views, opinions, and/or findings expressed are those of the author(s) and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. This work was supported by INHA UNIVERSITY Research Grant.

## References


Albasyoni, A.; Safaryan, M.; Condat, L.; and Richtárik, P. 2020. Optimal gradient compression for distributed and federated learning. *arXiv preprint arXiv:2010.03246*.

Alistarh, D.; Grubic, D.; Li, J.; Tomioka, R.; and Vojnovic, M. 2017. QSGD: Communication-efficient SGD via gradient quantization and encoding. *Advances in Neural Information Processing Systems*, 30.

Alistarh, D.; Hoefler, T.; Johansson, M.; Khirirat, S.; Konstantinov, N.; and Renggli, C. 2018. The convergence of sparsified gradient methods. *arXiv preprint arXiv:1809.10505*.

Bibikar, S.; Vikalo, H.; Wang, Z.; and Chen, X. 2021. Federated dynamic sparse training: Computing less, communicating less, yet learning better. *arXiv preprint arXiv:2112.09824*.

Brock, A.; Lim, T.; Ritchie, J. M.; and Weston, N. 2017. Freezeout: Accelerate training by progressively freezing layers. *arXiv preprint arXiv:1706.04983*.

Caldas, S.; Duddu, S. M. K.; Wu, P.; Li, T.; Konečnỳ, J.; McMahan, H. B.; Smith, V.; and Talwalkar, A. 2018. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*.

Cohen, G.; Afshar, S.; Tapson, J.; and Van Schaik, A. 2017. EMNIST: Extending MNIST to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, 2921–2926. IEEE.

Goutam, K.; Balasubramanian, S.; Gera, D.; and Sarma, R. R. 2020. LayerOut: Freezing Layers in Deep Neural Networks. *SN Computer Science*, 1(5): 1–9.

Haddadpour, F.; Kamani, M. M.; Mahdavi, M.; and Cadambe, V. R. 2019. Local sgd with periodic averaging: Tighter analysis and adaptive synchronization. *arXiv preprint arXiv:1910.13598*.

Haddadpour, F.; Kamani, M. M.; Mokhtari, A.; and Mahdavi, M. 2021. Federated learning with compression: Unified analysis and sharp guarantees. In *International Conference on Artificial Intelligence and Statistics*, 2350–2358. PMLR.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hyeon-Woo, N.; Ye-Bin, M.; and Oh, T.-H. 2021. FedPara: Low-Rank Hadamard Product for Communication-Efficient Federated Learning. *arXiv preprint arXiv:2108.06098*.

Jiang, Y.; Wang, S.; Valls, V.; Ko, B. J.; Lee, W.-H.; Leung, K. K.; and Tassiulas, L. 2022. Model pruning enables efficient federated learning on edge devices. *IEEE Transactions on Neural Networks and Learning Systems*.

Karimireddy, S. P.; Kale, S.; Mohri, M.; Reddi, S.; Stich, S.; and Suresh, A. T. 2020. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, 5132–5143. PMLR.

Konečnỳ, J.; McMahan, H. B.; Yu, F. X.; Richtárik, P.; Suresh, A. T.; and Bacon, D. 2016. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*.

Krizhevsky, A. 2009. Learning multiple layers of features from tiny images. Technical report.

Kumar, A.; Balasubramanian, A.; Venkataraman, S.; and Akella, A. 2019. Accelerating deep learning inference via freezing. In *11th {USENIX} Workshop on Hot Topics in Cloud Computing (HotCloud 19)*.

Li, T.; Sahu, A. K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; and Smith, V. 2018. Federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, 1273–1282. PMLR.

Phan, A.-H.; Sobolev, K.; Sozykin, K.; Ermilov, D.; Gusak, J.; Tichavskỳ, P.; Glukhov, V.; Oseledets, I.; and Cichocki, A. 2020. Stable low-rank tensor decomposition for compression of convolutional neural network. In *European Conference on Computer Vision*, 522–539. Springer.

Raghu, M.; Gilmer, J.; Yosinski, J.; and Sohl-Dickstein, J. 2017. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. *arXiv preprint arXiv:1706.05806*.

Reisizadeh, A.; Mokhtari, A.; Hassani, H.; Jadbabaie, A.; and Pedarsani, R. 2020. Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization. In *International Conference on Artificial Intelligence and Statistics*, 2021–2031. PMLR.

Sattler, F.; Wiedemann, S.; Müller, K.-R.; and Samek, W. 2019. Robust and communication-efficient federated learning from non-iid data. *IEEE transactions on neural networks and learning systems*, 31(9): 3400–3413.

Stich, S. U.; Cordonnier, J.-B.; and Jaggi, M. 2018. Sparsified SGD with memory. *Advances in Neural Information Processing Systems*, 31.

Vogels, T.; Karimireddy, S. P.; and Jaggi, M. 2020. Practical Low-Rank Communication Compression in Decentralized Deep Learning. In *NeurIPS*.

Wang, H.; Agarwal, S.; and Papailiopoulos, D. 2021. Pufferfish: Communication-efficient Models At No Extra Cost. *arXiv preprint arXiv:2103.03936*.

Wang, H.; Sievert, S.; Charles, Z.; Liu, S.; Wright, S.; and Papailiopoulos, D. 2018. Atomo: Communication-efficient learning via atomic sparsification. *arXiv preprint arXiv:1806.04090*.

Wang, J.; and Joshi, G. 2018a. Adaptive communication strategies to achieve the best error-runtime trade-off in local-update SGD. *arXiv preprint arXiv:1810.08313*.

Wang, J.; and Joshi, G. 2018b. Cooperative SGD: A unified framework for the design and analysis of communication-efficient SGD algorithms. *arXiv preprint arXiv:1808.07576*.

Wang, J.; Liu, Q.; Liang, H.; Joshi, G.; and Poor, H. V. 2020. Tackling the objective inconsistency problem in

heterogeneous federated optimization. *arXiv preprint arXiv:2007.07481*.

Wangni, J.; Wang, J.; Liu, J.; and Zhang, T. 2017. Gradient sparsification for communication-efficient distributed optimization. *arXiv preprint arXiv:1710.09854*.

Wen, W.; Xu, C.; Yan, F.; Wu, C.; Wang, Y.; Chen, Y.; and Li, H. 2017. Terngrad: Ternary gradients to reduce communication in distributed deep learning. *arXiv preprint arXiv:1705.07878*.

Yang, H.; Fang, M.; and Liu, J. 2021. Achieving linear speedup with partial worker participation in non-iid federated learning. *arXiv preprint arXiv:2101.11203*.

Yao, D.; Pan, W.; Wan, Y.; Jin, H.; and Sun, L. 2021. FedHM: Efficient Federated Learning for Heterogeneous Models via Low-rank Factorization. *arXiv preprint arXiv:2111.14655*.

You, Y.; Li, J.; Reddi, S.; Hseu, J.; Kumar, S.; Bhojanapalli, S.; Song, X.; Demmel, J.; Keutzer, K.; and Hsieh, C.-J. 2019. Large batch optimization for deep learning: Training bert in 76 minutes. *arXiv preprint arXiv:1904.00962*.

Zagoruyko, S.; and Komodakis, N. 2016. Wide residual networks. *arXiv preprint arXiv:1605.07146*.

Zhang, M.; and He, Y. 2020. Accelerating training of transformer-based language models with progressive layer dropping. *arXiv preprint arXiv:2010.13369*.