EFFECTIVENESS OF LOCAL STEPS ON HETEROGENEOUS DATA: AN IMPLICIT BIAS VIEW

Anonymous authors
Paper under double-blind review

ABSTRACT

In distributed training of machine learning models, gradient descent with *local iterative steps* is a very popular method to mitigate communication burden, commonly known as Local (Stochastic) Gradient Descent (Local-(S)GD). In the interpolation regime, Local-GD can converge to zero training loss. However, with many potential solutions corresponding to zero training loss, it is not known which solution Local-GD converges to. In this work we answer this question by analyzing implicit bias of Local-GD for classification tasks with *linearly separable data*. In the case of highly heterogeneous data, it has been observed empirically that local models can diverge significantly from each other (also known as "client drift"). However, for the interpolation regime, our analysis shows that the aggregated global model resultant from Local-GD with *arbitrary number* of local steps converges exactly to the model that would result in if all data were in one place (centralized trained model) in direction. Our result gives the exact rate of convergence to the centralized model with respect to the number of local steps. We also obtain this same implicit bias with a learning rate independent of number of local steps with a Modified Local-GD algorithm for the case local problems are exactly solved. Our analysis provides a new view to understand why Local-GD can still work very well with a very large number of local steps even for heterogeneous data. Lastly we also discuss the extension of our results to Local SGD and non-separable data.

1 INTRODUCTION

In this era of large machine learning models, distributed training is an essential part of machine learning pipelines. It can happen in a data center with thousands of connected compute nodes Sergeev & Del Balso (2018); Huang et al. (2019), or across several data centers and millions of mobile devices in federated learning Konečnỳ et al. (2016); Kairouz et al. (2019). In such a network, the communication cost is usually the bottleneck in the whole system. To alleviate the communication burden, and also to preserve privacy to some extent, one common strategy is to perform multiple local updates before sending the information to other nodes, which is called Local Gradient Descent (Local-GD) McMahan et al. (2017); Stich (2019); Lin et al. (2019). In a network with M compute nodes, the goal is to train a global model to fit the distributed datasets:

$$\min_{w\in\mathbb{R}^d} f(w) \qquad \text{with } f(w)\equiv\frac{1}{M}\sum_{i=1}^{M} f_i(w), \tag{1}$$

where $w\in\mathbb{R}^d$ is the single model to be trained and $f_i(w)$ is the local loss function for i^{th} compute node. The local loss $f_i(w)$ is the average of the loss function evaluated at model w for the high-dimensional samples and their corresponding labels, $\{x_s,y_s\}_{s\in S_i}$, where S_i is the local dataset, and $N_i=|S_i|$ is the number of local samples. The samples of the local dataset are obtained iid from the local distribution D_i.

In each round of Local-GD, a central node sends its current model, referred to as the ***global model***, to all compute nodes. Each compute node runs L local gradient descent steps on the global model using its loss f_i on this model to obtain a local model. Each compute node sends its local model back to the central node, where these local models are aggregated, by averaging, to obtain the global model for the next round. The detailed algorithm of Local-GD is described in Algorithms 1.

In modern machine learning, most deep neural networks, where Local-GD has impressive performance, operate in the *overparameterized regime*, where the dimension d of the model is more than the total number of samples MN. In this case, there are multiple solutions corresponding to zero training loss. The main question here is:

Q: Which solution would the aggregated model trained by Local-GD converge to?

Contributions. In this work, we answer this question by analyzing *implicit bias* of Local-GD on classification tasks for linearly separable data. From the implicit bias of Local-GD, we can characterize the dynamics of the global model across rounds. We compare the global model with the ***centralized model*** obtained from running gradient descent (GD) on a dataset consisting of all distributed datasets as if all these datasets were located on the central node. The centralized model is obtained from existing results for the implicit bias of linearly separable data Soudry et al. (2018). But these results cannot be directly applied to Local-GD. For globally linearly separable dataset, we show that the global model converges to the centralized model with any arbitrary number of local steps on heterogeneous data. As a consequence of our result on the implicit bias of Local-GD, we can derive the rate of convergence to centralized model as $O(\frac{1}{\log Lk})$, and the training loss converges at the rate of $O(\frac{1}{Lk})$, where k is number of rounds (see Theorems 2) for a constant learning rate $\eta = \mathcal{O}(\frac{1}{L})$ (this learning rate is common in existing analysis of distributed learning Karimireddy et al. (2020); Koloskova et al. (2020); Crawshaw et al. (2025)). The meaning of this work lies in: 1). providing a theoretical explanation to the phenomenon that Local-GD can work well with a very large number of local steps in practice; 2). showing the local steps can benefit the convergence rate for smooth, convex functions (such as, logistic loss); this could not be derived from previous analysis in vanilla Local-GD.

For a learning rate independent of L, we consider a special case where each local problem with a weakly regularized term is exactly solved, which indicates the behavior of Local-GD with a very large number of local steps. With a Modified Local-GD algorithm (see Section 4.4 we can guarantee that the global model can converge to the centralized model. This result provides the implicit bias of massive local updates without the restrictive learning rate of $O(1/L)$.

Comparisons. Increasing local steps L does not improve worst-case amount of communication for smooth, convex optimization (Woodworth et al., 2020, Theorem 5),(Koloskova et al., 2020, Theorem 6). For the specific problem of distributed logistic regression, (Crawshaw et al., 2025, Corollary 3) show that a two-stage Local-GD algorithm can improve this worst-case bound. However, their first stage still requires $\eta = \mathcal{O}(\frac{1}{L})$, and they can only show that the loss converges, but not the solution the model converges to. In contrast, our Theorem 2 exactly characterizes the global model for Local-GD for any L, and recovers their result as a direct corollary. Another line of work Gu et al. (2023; 2024) approximates Local-Stochastic Gradient Descent (localSGD) by an SDE to obtain an appropriate scaling between L and η. Note that we perform Local-GD with no stochastic noise, and our analysis is exact for finite η. Further, Gu et al. (2023; 2024) do not characterize the exact implicit bias, which we do for linearly separable data. For overparameterized non-linear models, several works Deng et al. (2022b); Song et al. (2023); Maralappanavar et al. (2025) analyze convergence in loss value of Local-GD, but do not provide any guarantees on the global model. Additionally, several works compare the performance of Local-GD and GD on whole dataset Patel et al. (2024); Woodworth et al. (2020) with differences in certain regimes. For overparametrized linear models, we establish that there is no difference between the final model learned by either of these methods.

Practical Implications. In the existing convergence analysis of Local-GD, the number of local steps L should not be very large for heterogeneous data Stich (2019); Li et al. (2020b). In practical implementation of distributed training on large models, the performance of Local-GD is surprisingly good even with heterogeneous data distribution McMahan et al. (2017); Charles et al. (2021). Also, the number of local steps can be very large in Local-GD type algorithms and real-world systems, for example, up to 500 local steps in distributed training of large language models (LLM) Douillard et al. (2023); Jaghouar et al. (2024). Since our results show the Local-GD can converge to centralized model with *arbitrary* number of local steps, it helps explain why Local-GD can still work well with a large number of local steps in practice. In this work we consider linear models as an appropriate starting point to investigate the implicit bias of Local-GD. A popular example of linear models used in practical machine learning pipelines is fine-tuning last layer on pretrained large models or adding linear layers in transfer learning Donahue et al. (2014); Kornblith et al. (2019) and deployment of LLM Devlin (2018); Jiang et al. (2020). Thus we also add an experiment of fine-tuning last layer of neural network to show broader impact of our analysis.

1.1 Related Work

Convergence of Local-GD. When data distribution is homogeneous, many works have been done to establish convergence analysis for Local (Stochastic) GD Stich (2019); Yu et al. (2019); Khaled et al. (2020). With a "properly" small number of local steps, the dominating convergence rate is not affected. Further various assumptions have been made to handle data heterogeneity and develop convergence analysis Li et al.

(2020b); Karimireddy et al. (2020); Khaled et al. (2020); Reddi et al. (2021); Wang et al. (2020); Crawshaw et al. (2023). For strongly convex and smooth loss functions, the number of local steps should not be larger than $O(\sqrt{T})$ for i.i.d data Stich (2019) and non-i.i.d. data Li et al. (2020b). However, in practice Local-GD (FedAvg) works well in many applications McMahan et al. (2017); Charles et al. (2021), even in training large language models Douillard et al. (2023); Jaghouar et al. (2024). In Wang et al. (2024), the authors argue that the previous theoretical assumption does not align with practice and proposed a client consensus hypothesis to explain the effectiveness of FedAvg in heterogeneous data. But they do not consider the impact of overparameterization on distributed training. There are some works incorporating the property of zero training loss of overparameterized neural networks into the conventional convergence analysis of FedAvg Huang et al. (2021); Deng et al. (2022a); Song et al. (2023); Qin et al. (2022). However, they do not guarantee which point FedAvg can converge to, which is especially important for overparameterized models since there are multiple solutions with zero training loss. Our work is different from these works as: 1. We analyze which point the Local-GD can converge to, which is a more elementary problem before obtaining the convergence rate; 2. We use implicit bias as a technical tool to analyze the overparameterized FL.

Implicit Bias. Soudry et al. (2018) is the first work to show the gradient descent converges to a max-margin direction on linearly separable data with a linear model and exponentially-tailed loss function. Ji & Telgarsky (2019a) has provided an alternative analysis and extended this to non-separable data. The theory of implicit bias has been further developed, for example, for wide two-layer neural networks Chizat & Bach (2020), deep linear models Ji & Telgarsky (2019b), linear convolutional networks Gunasekar et al. (2018b), two-layer ReLU networks Kou et al. (2024) etc. Beyond gradient descent, more algorithms have been considered, including gradient descent with momentum Gunasekar et al. (2018a), SGD Nacson et al. (2019), Adam Cattaneo et al. (2023), AdamW Xie & Li (2024). Recently, implicit bias has also been used to characterize the dynamics of continual learning, on linear regression Evron et al. (2022); Goldfarb & Hand (2023); Lin et al. (2023), and linear classification Evron et al. (2023); Jung et al. (2025). In Evron et al. (2023), gradient descent on continually learned tasks is related to Projections onto Convex Sets (POCS) and shown to converge to a *sequential* max-margin scheme. In our work we consider the implicit bias of gradient descent in distributed setting, which is related to a different parallel projection scheme by projecting onto constraint sets *simultaneously*.

Parallel Projection. Parallel projection methods are a family of algorithms to find a common point across multiple constraint sets by projecting onto these sets in parallel. These methods are widely used in feasibility problems in signal processing and image reconstruction Bauschke & Combettes (2011). The straightforward average of multiple projections is known as the simultaneous iterative reconstruction technique (SIRT) in Gilbert (1972). Then de Pierro & Iusem (1984) studied the convergence of PPM for a relaxed version, and Combettes (1994) further generalized the result to inconsistent feasibility problems. In Combettes (1997), an extrapolated parallel projection method was proposed to accelerate the convergence. We note that Jhunjhunwala et al. (2023) used this extrapolation to accelerate FedAvg. However, it was just inspired by the similarity between parallel projection method and FedAvg, while in this work we rigorously prove the relation between PPM and FedAvg using implicit bias of gradient descent.

Algorithm 1 LOCAL-GD.

Input: learning rate η.
Initialize w_0^0
for $k=0$ to $K-1$ **do**
  The aggregator sends global model w_0^k to all compute nodes.
  for $i=1$ to $i=M$ **do**
    compute node i updates local model starting from w_0^k: $w_i^{k,0}=w_0^k$.
    for $l=0$ to $L-1$ **do**
      $w_i^{k,l+1}=w_i^{k,l}-\eta\nabla f_i(w_i^{k,l})$.
    end for
    compute node i sends back the updated local model $w_i^{k+1}=w_i^{k,L}$.
  end for
  The aggregator aggregates all the local models: $w_0^{k+1}=\frac{1}{M}\sum_{i=1}^{M}w_i^{k+1}$.
end for
Output: w_0^K.

2 MOTIVATING OBSERVATION IN LINEAR REGRESSION

In this section we first give some observations in linear regression as a motivating example. The behavior of linear regression is very well-understood in high-dimensional statistics.

Setting: At each compute node i, the dataset S_i consists of N tuples of samples and their corresponding labels, $(x,y) \in \mathbb{R}^d \times \mathbb{R}$. Denote $X_i = [x_{i1}, x_{i2}, ..., x_{iN}]^T \in \mathbb{R}^{N \times d}$ as the data matrix at i-th compute node, and $y_i = [y_{i1}, y_{i2}, ..., y_{iN}] \in \mathbb{R}^N$ as the label vector. Let $X_c = [X_1^T, ..., X_M^T]^T \in \mathbb{R}^{MN \times d}$ be the data matrix consisting of all the local data, and $y_c = [y_1^T, ..., y_M^T]^T \in \mathbb{R}^{MN \times 1}$ be the label vector consisting of the local labels.

We consider a special case of Local-GD in Algorithm 1 where the number of local steps is very large. At each round, the aggregator sends the global model w_0 to all the compute nodes. Each compute node minimizes the squared loss $f_i(w_i) = \frac{1}{2N} \|y_i - X_i w_i\|^2$ by *a large number* of gradient descent steps *until convergence*. Then each compute node sends back the local model and the aggregator aggregates all the local models to get the updated global model.

Underparameterized Regime: When the number of local samples is larger than the dimension d, it is known that local model would converge to the ordinary least square solution $w_i^{k+1} = (X_i^T X_i)^{-1} X_i^T y_i$ regardless of initial point w_i^k. In the meanwhile, the centralized model with all the training samples is $w_c = (X_c^T X_c)^{-1} X_c^T y_c$. However, the average of local models $w_0 = \sum_{i=1}^M (X_i^T X_i)^{-1} X_i^T y_i$ is not identical to the centralized model unless the data is homogeneously distributed and all $X_i^T X_i$ are proportional. So a large number of local steps can hurt the convergence to centralized model with heterogeneous data distribution.

Overparameterized Regime: When the dimension is larger than the number of samples at each compute node ($d > N$), there are multiple solutions corresponding to zero squared loss. However, it is known that gradient descent would converge to the minimum norm solution in the feasible set, which corresponds to a minimum Euclidean distance to the initial point Gunasekar et al. (2018a); Evron et al. (2022), i.e., the solution of the optimization problem

$$\min_{w_i} \quad \|w_i - w_0^k\|^2 \quad \text{s.t.} \quad X_i w_i = y_i. \tag{2}$$

We can obtain the closed form solution as $w_i^{k+1} = (I - P_i) w_0^k + X_i^\dagger y_i$, where $P_i \triangleq X_i^T (X_i X_i^T)^{-1} X_i$ and $X_i^\dagger \triangleq X_i^T (X_i X_i^T)^{-1}$. We observe that P_i is the projection operator to the row space of X_i, and $X_i^\dagger$ is the pseudo inverse of X_i. Meanwhile the centralized model converges to the minimum norm solution $w_c = X_c^T (X_c X_c^T)^{-1} y_c$. Denote $\bar{P} = \frac{1}{M} \sum_{i=1}^M P_i$. In the training process the difference between global model and centralized model is iteratively projected onto the null space of span of row spaces of X_is. It implies that the difference on the span of data matrix gradually decreases until zero. Based on the evolution of the difference, we can prove the following theorem:

Theorem 1. *For the linear regression problem, suppose the initial point w_0^0 is 0 and $d > MN$ and the minimum eigenvalue $\theta_{\min}$ of $\bar{P}$ is larger than 0, then the output of Local-GD, w_0^K, converges to the centralized solution w_c as the number of communication rounds $K \to \infty$ as $\|w_0^K - w_c\| \leq (1 - \theta_{\min})^K \|w_c\|$.*

The proof is deferred in Appendix B. The key step is to show the initial difference is already in the data space, and no residual in the null space of row spaces of X_is. The convergence to the centralized model is at exponential rate. Due to the linearity of the regression problem, we can theoretically show the global model can exactly converge to the centralized model with implicit bias on overparameterized regime. It implies that, even if we use a large number of local steps to exactly solve the local problems on very heterogeneous data, the performance of Local-GD is equivalent to train a model with all the data in one place.

3 IMPLICIT BIAS OF LOCAL-GD FOR CLASSIFICATION

For classification task, we also would like to know whether the global model can converge to the centralized model with any number of local steps. Now we investigate a binary classification task with linear models.

3.1 SETTING

Suppose, for each compute node i, the dataset S_i consists of N_i tuples of samples and their corresponding labels, $(x,y) \in \mathbb{R}^d \times \{+1, -1\}$. We denote $X_i \in \mathbb{R}^{N_i \times d}$ as the data matrix at i-th compute node, and $y_i \in \{+1, -1\}^{N_i}$ as the label vector. The global dataset is the set of M local datasets $S = \cup_{i=1}^M S_i$.

We consider a linear model $w \in \mathbb{R}^d$ for the binary classification task. The local loss at i-th compute node is

$$f_i(w) = \sum_{s \in S_i} g(y_s x_s^T w), \tag{3}$$

where $g(u)$ is a loss function decreasing to zero when $u \to \infty$, such as logistic loss $g(u) = \ln(1+e^{-u})$.

We study LocalGD with an *arbitrary number* of gradient descent steps. To describe our main results, we have the following notations and assumptions. We denote the whole data matrix as $X \in \mathbb{R}^{N \times d}$, where $N = \sum_{i=1}^{M} N_i$. We write $\sigma_{\max} = \sqrt{\theta_{\max}(X^T X)}$ as the maximum singular value of data matrix X, where θ represents eigenvalues of a square matrix. We need an assumption of global separability on whole dataset.

Assumption 1. For all the data samples $(x_s, y_s) \in S$, there exists $w \in \mathbb{R}^d$ such that $y_s x_s^T w > 0$.

Note that linear separability is a common assumption in the analysis of learning in overparameterized regime Nacson et al. (2019); Soudry et al. (2018); Evron et al. (2023). For our distributed case, this implies that all clients share at least 1 minimizer, which imposes an extremely mild condition on the data heterogeneity among clients. In the overparametrized setting, $d \geq mn$, hence, there are likely several such solutions separating the whole dataset. Since there are multiple solutions separating the whole dataset, we define a particular max-margin solution on global dataset:

$$\hat{w} = \arg\min_{w \in \mathbb{R}^d} \|w\| \quad \text{s.t.} \quad y_s x_s^T w \geq 1, \quad \forall s \in S. \tag{4}$$

It has been proven that gradient descent would implicitly lead the linear model to this max-margin solution in direction, i.e., convergence of model direction to $\hat{w}/\|\hat{w}\|$ Soudry et al. (2018). We define the maximum margin as

$$\gamma = \max_{w \in \mathbb{R}^d, \|w\|=1} \min_s y_s x_s^T w \tag{5}$$

which is strictly positive since the global dataset is linearly separable. The data points reaching this margin are support vectors of the global dataset.

To establish convergence, we require additional regularity assumptions on the loss function.

Assumption 2. The loss function $g(u)$ is a positive, differentiable, β-smooth function, monotonically decreasing to zero, and $\limsup_{u \to -\infty} g' < 0$.

Assumption 3. The negative loss derivative $-g'(u)$ has a tight exponential tail. That is, there exists positive constants μ_+, μ_- and $\bar{u}$ such that $\forall u > \bar{u}$:

$$(1 - \exp(-\mu_- u))e^{-u} \leq -g'(u) \leq (1 + \exp(-\mu_+ u))e^{-u}. \tag{6}$$

Note that these assumptions are also used in centralized learning of overparameterized models Soudry et al. (2018); Nacson et al. (2019); Evron et al. (2023), and the logistic loss satisfies all the assumptions. With our setting completely defined, we state our main results.

3.2 Loss Convergence and Implicit Bias of Local-GD

Our main result is on the asymptotic convergence of the model parameter w_0 and loss $f(w)$ for Local-GD.

Theorem 2. *Under assumptions 1, 2, 3, if the learning rate satisfies* $\eta < \min\left(\frac{1}{2L\sigma_{max}^2\beta}, \frac{\gamma^2}{4L\sigma_{\max}^3\beta(\gamma+\sigma_{\max})}\right)$, *then for the process of Local-GD, we have,*

- *Every data point is classified correctly finally:* $\lim_{k\to\infty} x_s^T w_0^k = \infty, \forall s \in S$.
- *The global model obtained from Local-GD will behave as*

$$w_0^k = \log(Lk)\hat{w} + \rho^k, \quad \textit{and,} \quad \left\| \frac{w_0^k}{\|w_0^k\|} - \frac{\hat{w}}{\|\hat{w}\|} \right\| = O\left(\frac{1}{\log Lk}\right) \tag{7}$$

  and $\|\rho^k\| < \infty$ *for all* k. *This implies, the normalized global model converges to the global max-margin solution.*

- *The loss function* $f(w_0^k)$ *decreases to zero as* $f(w_0^k) = O\left(\frac{1}{Lk}\right)$.

The proof is deferred to Appendix C. The technical challenges lie in that we need to control the residual term ρ^k with the *local steps* and *aggregations*, which are handled by a refined analysis in distributed context. This theorem implies the global model can eventually correctly classify all the training samples after many rounds of communication. Given that centralized model also converges to the global max-margin solution from prior results, the global model from Local-GD actually converges to the exact centralized model in direction. Further, this holds for a step size $\eta \propto \frac{1}{L}$, and does not require any additional modifications to the objective, for instance, any regularization on the difference between local and global models during local steps.

Impact of local steps. In this analysis, the number of local steps can be arbitrary. Although the magnitude of model vector would diverge to infinity, the direction of aggregated model still converges to the direction of global max-margin solution. Thus, the number of local steps does not influence the asymptotic convergence to the centralized model, which is very different from underparameterized regime. This result also shows the local steps can be beneficial for convergence to the global max-margin solution as both the loss and the directional error decrease with total number of gradient descent steps(Lk) at rates $\frac{1}{Lk}$ and $\frac{1}{\log(Lk)}$ respectively. Additionally, our convergence rates also match those obtained for GD in centralized learning Soudry et al. (2018) with total number of steps Lk. This demonstrates that our analysis is tight. Further, for constant γ, if we use the same number of local steps, $L = \Theta(\sqrt{\frac{M}{\epsilon}})$ as two-stage Local-GD in (Crawshaw et al., 2025, Corollary 3), then we require the same number of rounds $\mathcal{O}(\sqrt{\frac{M}{\epsilon}})$ of Local-GD to achieve $f(w_0^k) \leq \epsilon$. Note that both Crawshaw et al. (2025) as well as our Theorem 2 require the number of rounds to be larger than some $\bar{k}$ after which asymptotics kick in. Therefore, we assume that ϵ is small enough that the number of rounds to be $\mathcal{O}(\sqrt{\frac{M}{\epsilon}}$ is much larger than this $\bar{k}$.

Learning Rate. Theorems 2 needs the learning rate to be small as $O(1/L)$, which has also been used by existing works Karimireddy et al. (2020); Koloskova et al. (2020); Crawshaw et al. (2025) on Local-GD and Local-SGD. This means the model does not move so far after one round of local iterations. Next, we would see whether the global model still converges to max-margin solution with a learning rate independent of L.

3.3 Discussions

Extension to Local SGD. It is straightforward to extend our analysis of Local-GD to Local SGD that chooses samples without replacement. At each local step of i-th compute node, the update is $w_i^{k,l+1} = w_i^{k,l} - \eta \frac{1}{B} \sum_{s \in S_{i,l}} \nabla g(y_s x_s^T w_i^{k,l})$, where $S_{i,l}$ is the mini-batch of samples at l-th local step and $B = |S_{i,l}|$ is the batch size. We consider the following setting of sampling:

Assumption 4 (Sampling without replacement.)**.** At every communication round, each compute node run stochastic gradient descent with E epochs, where E is an positive integer. Within each epoch, the mini-batches $\{S_{i,0}, S_{i,1}, ..., S_{i,l'}\}$ partition the local dataset S_i, where $l' = N/B$ is the number of local steps for one epoch.

Under this setting, each sample is exactly chosen once inside one epoch of local updates. At each round, the local datasets are passed E times, which is a practically common way. To extend our analysis to Local SGD, we can regard one local dataset as a "batch" in SGD for sampling without replacement. And then we perform multiple gradient steps in the same "batch", not only one step of gradient descent in SGD. In Local SGD, each step is a gradient descent step on a mini batch of local datasets, but we still run the gradient descent steps for E "local steps". Therefore, we can obtain the same asymptotic results as Theorem 2 for Local SGD without any change of the proof framework.

Separability Assumption. In this paper we mainly focus on the linearly separable data, which is a standard assumption in implicit bias analysis and also widely used in recent works Zhang et al. (2024); Crawshaw et al. (2025); Jung et al. (2025). For non-separable case, Ji & Telgarsky (2019a) has shown gradient descent converges to a ray along the direction of max-margin solution of largest linearly separable subset. However, there is still an assumption on the data: in fact, one needs a positive margin on the separable part of data to show both convergence in risk or parameters. Nevertheless, Ji & Telgarsky (2019a) clearly shows strict linear separability is not the main reason for the convergence of gradient descent to a max-margin solution. Since even without this assumption, GD still converges to a variant form of max-margin solution. It is possible to use the same idea in Local-GD. Intuitively, in the case where local datasets are linearly separable but global dataset is non-separable, although local training would guide local

models to local max-margin solutions, the aggregations would force the global model to converge to the max-margin solution of largest linearly separable subset of global dataset, which is the centralized solution.

4 Implicit Bias of Local-GD with Learning Rate Independent of L

4.1 Setting

In this section, we consider Local-GD in a slightly different setting. We aim to solve a local optimization problem with exponential loss and a weakly regularized term for each compute node. The local problem is solved exactly (to reach the local optima) with a large number of local steps.

Algorithm. At each round, the aggregator sends the global model w_0 to all the compute nodes. Each compute node minimizes an *exponential loss* with a *weakly regularized term* by many gradient descent steps *until convergence*. That is, each compute node solves the following problem:

$$\min_{w\in\mathbb{R}^d} f_i(w) \qquad \text{where } f_i(w)\equiv\sum_{s\in S_i}\exp\left(-y_s x_s^T w\right)+\frac{\lambda}{2}\|w-w_0^k\|^2 \tag{8}$$

where λ is a regularization parameter close to 0.

Then each compute node sends back the local model and the aggregator aggregates all the local models to get the updated global model (i.e., they follow Algorithm 1 with $f_i(w_i)$ as specified here).

Regularization methods are very common in distributed learning to force the local models move not too far from global model Li et al. (2020a; 2021); T Dinh et al. (2020). Here we consider the weakly regularized term, $\lambda\to 0$, to give theoretical insights of Local-GD on classification tasks. Experimentally the λ is set to be extremely small that does not affect the minimization of exponential loss. For the local loss functions, we have one assumption on smoothness:

Assumption 5. For each compute node, the local loss function $f_i(w)$ is B-smooth for any round of local steps k.

Learning Rate. In the following analysis of implicit bias, we actually exploit the property of local minimizers. Since local problem (8) is a strongly convex problem for $\lambda>0$, we can run local gradient descent to find the unique minimizer with a learning rate $\eta\leq\frac{2}{B}$ for a large number of local steps L. That's the only requirement of learning rate, which is not dependent of number of local steps L. In other words, the learning rate is only needed to be sufficiently small to ensure local convergence at each round of Local-GD.

4.2 Implicit Bias of Local-GD and Relation to PPM

We consider the whole algorithmic process of Local-GD on classification and use another auxiliary sequence of global models, denoted as $\bar{w}_0^k, k=0,1,2,\ldots$. Starting from an initial point $\bar{w}_0^0$, the central node sends global model $\bar{w}_0^k$ to all the compute nodes at k-th iteration round. Each compute node solves the following *Local Max-Margin* problem to obtain $\bar{w}_i^{k+1}$:

$$\bar{w}_i^{k+1}=\arg\min_{w\in\mathbb{R}^d}\|w-\bar{w}_0^k\| \quad \text{s.t.} \quad y_s x_s^T w\geq 1, \quad \forall s\in S_i. \tag{9}$$

Then the compute node sends the local model back. The central node averages the local models to get $\bar{w}_0^{k+1}=\frac{1}{M}\sum_{i=1}^M \bar{w}_i^{k+1}$. We can show the solution w_0^K obtained in Local-GD converges in direction to the global model from Local Max-Margin problems $\bar{w}_0^K$.

Lemma 1. *For almost all datasets sampled from a continuous distribution satisfying Assumption 1, with initialization $w_0^0=\bar{w}_0^0=0$, we have $w_0^k\to\ln\left(\frac{1}{\lambda}\right)\bar{w}_0^k$, and the residual $\|w_0^k-\ln\left(\frac{1}{\lambda}\right)\bar{w}_0^k\|=O(k\ln\ln\frac{1}{\lambda})$, as $\lambda\to 0$. It implies that at any round $k=o\left(\frac{\ln(1/\lambda)}{\ln\ln(1/\lambda)}\right)$, w_0^k converges in direction to $\bar{w}_0^k$:*

$$\lim_{\lambda\to 0}\frac{w_0^k}{\|w_0^k\|}=\frac{\bar{w}_0^k}{\|\bar{w}_0^k\|}. \tag{10}$$

The proof is deferred in Appendix D. The proof sketch is similar to the continual learning work Evron et al. (2023), but we have to handle the parallel local updates for each dataset from the same initial model

and the aggregation, which is different from the sequential updates where for each dataset the model is trained from the previous model and there is no need to do aggregation.

Based on this equivalence between Local-GD for linear classification and Local Max-Margin scheme, we can further analyze the performance of Local-GD with a large number of local steps. Instead of a closed-form solution for the Local Max-Margin problem (9), we treat it as a projection of the aggregated global model onto a convex set C_i: $\bar{w}_i^{k+1} = P_i(\bar{w}_0^k)$, which is formed by the constraints in (9) and exactly the local feasible set defined in Assumption 1. Here we slightly overload the notation P_i, which was used as the projection matrix in linear regression since the readers can get a sense of the same effect of them in Local-GD. The aggregation is actually to average the local projected points: $\bar{w}_0^{k+1} = \frac{1}{M}\sum_{i=1}^{M} P_i(\bar{w}_0^k)$.

The sequence of Local Max-Margin schemes is therefore projections to local (convex) feasible sets followed by aggregation, which is the Parallel Projection Method (PPM) in literature Gilbert (1972); Combettes (1994). Using Lemma 1, we establish the relation between Local-GD and PPM: the model from Local-GD converges to the model from PPM in direction.

4.3 Convergence to Global Feasible Set

Now we use the properties of PPM to characterize the performance of Local-GD in classification. In Combettes (1994), the convergence of PPM has been provided for a relaxed version. The direct average considered in this work can be seen as a special case of the relaxed version, and the following lemma holds.

Lemma 2 (Theorem 1 and Proposition 8, Combettes (1994))**.** *Suppose all the local feasible sets $C_i, i = 1,2,...$ are closed and convex, and the intersection $\bar{C}$ is not empty. Then for any initial point $\bar{w}_0^0$, the global model $\bar{w}_0$ generated by PPM converges to a point in the global feasible set $\bar{C}$.*

This lemma guarantees that $\bar{w}_0^K$ will converge to the intersection of the convex sets after many rounds of iteration, however we are not sure which exact point it would converge to.

Combining Lemma 1, Lemma 2 and the fact that centralized model would converge to the minimum norm solution in global feasible set, we immediately have:

Theorem 3. *For linear classification problem with exponential loss, suppose initial point is $w_0^0 = 0$. The aggregated global model w_0^K obtained by Local-GD with a large number of local steps converges in direction to one point in the global feasible set $\bar{C}$, while the centralized model converges in direction to the minimum norm point in the same set.*

Here we cannot guarantee the global model obtained by Local-GD with a learning rate independent of L to converge exactly to the centralized model in classification, but show that it converges to the same global feasible set as the centralized solution. To theoretically support that the Local-GD model converges to the centralized model, we propose a slightly Modified Local-GD by just changing the aggregation method, and showing that it converges to the centralized model exactly.

4.4 Modified Local-GD: Convergence to Centralized Model

In Combettes (1996) it was shown that if the aggregation method is modified to incorporate the influence of the initial point $\bar{w}_0^0$ in PPM, then the sequence generated by PPM will converge to a specific point in global feasible set $\bar{C}$ with minimum distance to this initial point. Denote $P_c(\cdot)$ as the projection operator onto the global feasible set $\bar{C}$. Formally we have the following lemma.

Lemma 3 (Theorem 5.3, Combettes (1996))**.** *Suppose $\bar{C}$ is not empty. For any initial point $\bar{w}_0^0$, when the local models are aggregated as*

$$\bar{w}_0^{k+1} = (1-\alpha^{k+1})\bar{w}_0^0 + \alpha^{k+1}\left(\frac{1}{M}\sum_{i=1}^{M} P_i(\bar{w}_0^k)\right), \tag{11}$$

where $\{\alpha^k\}$ satisfy $(i)\lim_{k\to\infty}\alpha^k = 1, (ii)\sum_{k\geq 0}(1-\alpha^k) = \infty, (iii)\sum_{k\geq 0}|\alpha^{k+1}-\alpha^k| < \infty$, then the global model generated by PPM will converge to the point $P_c(\bar{w}_0^0)$.

The sequence generated by PPM would converge to the point in global feasible set, $\bar{C}$, with minimum distance to $\bar{w}_0^0$. The modified aggregation method is a linear combination of initial point and current average of local projected points. One example of the sequence $\{\alpha^k\}$ satisfying the conditions is $\alpha^k = 1 - \frac{1}{k+1}$.

If we start from $\bar{w}_0^0 = 0$, then the point $P_c(\bar{w}_0^0)$ is exactly the minimum norm point in the global feasible set. It shows the PPM can exactly converge to the minimum norm point as the centralized model. Based on this result, we propose a Modified Local-GD algorithm, with the replacement of Line 9 in Algorithm 1 with

$$w_0^{k+1} = (1-\alpha^k) w_0^0 + \alpha^k \left(\frac{1}{M} \sum_{i=1}^{M} w_i^k \right). \tag{12}$$

We still need to prove a lemma analogous to Lemma 1 to establish the equivalence between Modified Local-GD and Modified PPM, which is omitted here due to space limit (Please refer to Appendix D and the proof is very similar to proof in Lemma 1). From the equivalence, Lemma 3, and implicit bias of the centralized model, we can have the following theorem:

Theorem 4. *For linear classification problem with local loss (8), suppose the initial point is $w_0^0 = 0$. Then the global model w_0^K obtained by Modified Local-GD converges in direction to the centralized model obtained from (4).*

Unlike the vanilla Local-GD, which is only guaranteed to converge to the global feasible set, the Modified Local-GD is guaranteed to converge to the centralized model in direction. Note that if we start from $\bar{w}_0^0 = 0$, the aggregation in Modified Local-GD becomes $w_0^{k+1} = \frac{k}{k+1} \left(\frac{1}{M} \sum_{i=1}^{M} w_i^k \right)$, which is just a *scaling* of vanilla aggregation with a parameter less than 1. Thus we can see experimentally they usually converge to the same point and Modified Local-GD converges slightly slower. With Modified Local-GD, we can theoretically show the global model still converges to centralized model in direction with a learning rate independent of L.

5 Experiments

We conducted various experiments on linear classification and neural network fine-tuning. We compared the ***global model***, i.e., the output of Local-GD (Algorithm 1), with the ***centralized model***, i.e., the model obtained from running GD on a dataset consisting of all distributed datasets at one place, in different scenarios.

5.1 Linear Classification

For linear classification, we have 10 compute nodes with 50 training samples at each. The dataset is generated as $y_{ij} = \text{sign}(x_{ij}^T w_i^*)$, where ground truth model is $w_i^* = w^* + z_i$, and w^* is a Gaussian vector randomly chosen, z_i is a Gaussian noise. The data matrix X_i is a Gaussian matrix. This setting makes sure the datasets across compute nodes are different from each other, meanwhile they are not totally different such that there may be a non-empty global feasible set.

We tested four models for linear classification. The global model (G) is trained exactly with Local-GD and logistic loss. The centralized model (C) is trained with gradient descent on the global dataset. The global model from Modified Local-GD (G-Mod) is trained with exponential loss and regularization term as $\lambda = 0.0001$. The centralized SVM model (S) (max-margin solution) is obtained by solving problem (4) via standard scikit-learn package. Note that centralized model and SVM model are the final trained model in the plots. The learning rate of (local) gradient descent is $\eta = 0.01$. Since our theory claimed the convergence is established in direction, the difference here for two models w_1, w_2 is defined after normalization $\|w_1/\|w_1\| - w_2/\|w_2\|\|$.

In Fig. 1(a), we show the difference between global model from Local-GD and centralized model with different number of local steps. The model dimension is chosen as $d = 1500$, ensuring it is globally over-parameterized. The centralized model is trained with 20000 gradient descent steps. It is seen the difference can approach zero for all the L, and larger L can result in faster convergence to the centralized model.

In Figs. 1(b), 1(c), 1(d), the number of local steps is fixed as $L = 150$ for Local-GD and Modified Local-GD, and the number of communication rounds is fixed as $R = 120$ for all the dimensions. Fig. 1(b) shows the difference between these models with respect to the number of rounds R when dimension is $d = 1500$. We can see both global model and modified global model converges to the centralized model in direction, and the centralized model is close to the SVM model but there is small gap. Fig. 1(c) displays the difference with respect to dimension d. It is seen the difference between global model and centralized model gradually decreases with larger dimensions. The modified global model is almost the same as

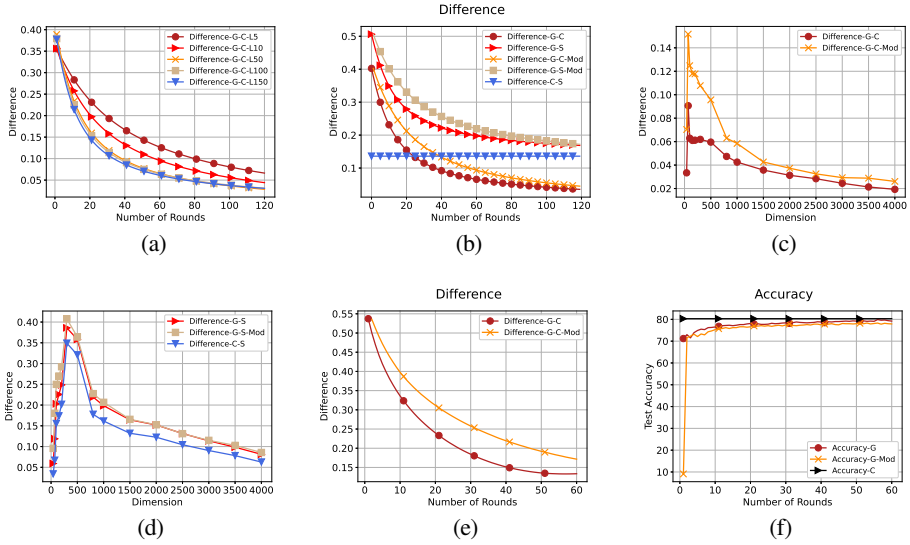


Figure 1: (a) Difference between global model and centralized model with L. (b) Difference between global model and centralized model with R. (c) Difference between global model and centralized model with d. (d) Difference from SVM model with d. (e) Difference between global linear layer and centralized linear layer with R. (f) Test accuracy of neural network fine-tuning.

the centralized model but the gap is slightly larger since it converges slower than vanilla global model with same number of rounds. Fig. 1(d) shows the difference from SVM model with dimension. The gap between the models to SVM model also decreases with larger d.

5.2 Fine-Tuning of Pretrained Neural Network

We further fine-tuned the ResNet50 model pretrained with ImageNet dataset on CIFAR10 dataset. Only the final linear layer is trained during the process, while the rest of model is fixed. The 50000 samples are distributed on 10 compute nodes. For i-th compute node, the half of local dataset belongs to the same class, and the other half consists of rest of 9 classes evenly, which forms a heterogeneous data distribution. The centralized model is trained with the whole CIFAR10 dataset. The models are trained with cross entropy loss and Local SGD. The learning rate is 0.01 and the batch size is 128. The number of local steps is $L=60$ and number of communication rounds is $R=60$. The centralized model is trained with the same learning rate for 3600 steps. We plot the difference between the linear layer and test accuracy with number of rounds in Fig. 1(e) and 1(f). Again the difference is defined in direction. We can see the difference gradually decreases to a small error floor and the accuracy of global models and centralized model is very similar at last.

Due to page limit, we put more experimental results on linear regression, linear classification with Dirichlet distribution in Appendix A.

6 Conclusions

In this work we analyzed the implicit bias of GD in distributed setting, and characterized the dynamics of the global model trained from Local-GD. We showed that Local-GD can converge to a centrally trained model for linearly separable data with a constant learning rate $O(1/L)$, and a Modified Local-GD can have the same convergence for a learning rate independent of L. Our analysis provided a new perspective why Local-GD works well in practice even with a large number of local steps on heterogeneous data.

REPRODUCIBILITY STATEMENT

This paper is mainly a theoretical work. The assumptions 1-5 are clearly explained in the main text. The proofs of Section 2 are included in Appendix B. The proof of Theorem 2 is included in Appendix C. The proofs of lemmas and theorems in Section 4 are included in Appendix D.

REFERENCES

Heinz H Bauschke and Patrick L Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer, 2011.

Matias D Cattaneo, Jason M Klusowski, and Boris Shigida. On the implicit bias of adam. *arXiv preprint arXiv:2309.00079*, 2023.

Zachary Charles, Zachary Garrett, Zhouyuan Huo, Sergei Shmulyian, and Virginia Smith. On large-cohort training for federated learning. *Advances in neural information processing systems*, 34:20461–20475, 2021.

Hong-You Chen and Wei-Lun Chao. Fedbe: Making bayesian model ensemble applicable to federated learning. In *International Conference on Learning Representations*, 2021.

Lenaic Chizat and Francis Bach. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. In *Conference on learning theory*, pp. 1305–1338. PMLR, 2020.

Patrick L Combettes. Inconsistent signal feasibility problems: Least-squares solutions in a product space. *IEEE Transactions on Signal Processing*, 42(11):2955–2966, 1994.

Patrick L Combettes. The convex feasibility problem in image recovery. In *Advances in imaging and electron physics*, volume 95, pp. 155–270. Elsevier, 1996.

Patrick L Combettes. Convex set theoretic image recovery by extrapolated iterations of parallel subgradient projections. *IEEE Transactions on Image Processing*, 6(4):493–506, 1997.

Michael Crawshaw, Yajie Bao, and Mingrui Liu. Federated learning with client subsampling, data heterogeneity, and unbounded smoothness: A new algorithm and lower bounds. *Advances in Neural Information Processing Systems*, 36, 2023.

Michael Crawshaw, Blake Woodworth, and Mingrui Liu. Local steps speed up local GD for heterogeneous distributed logistic regression. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=lydPkW4lfz.

Alvaro Rodolfo de Pierro and Alfredo Noel Iusem. *A parallel projection method of finding a common point of a family of convex sets*. Inst. de matemática pura e aplicada, Conselho nacional de desenvolvimento . . . , 1984.

Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Local sgd optimizes overparameterized neural networks in polynomial time. In *International Conference on Artificial Intelligence and Statistics*, pp. 6840–6861. PMLR, 2022a.

Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Local sgd optimizes overparameterized neural networks in polynomial time. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera (eds.), *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pp. 6840–6861. PMLR, 28–30 Mar 2022b. URL https://proceedings.mlr.press/v151/deng22a.html.

Jacob Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pp. 647–655. PMLR, 2014.

Arthur Douillard, Qixuan Feng, Andrei A Rusu, Rachita Chhaparia, Yani Donchev, Adhiguna Kuncoro, Marc'Aurelio Ranzato, Arthur Szlam, and Jiajun Shen. Diloco: Distributed low-communication training of language models. *arXiv preprint arXiv:2311.08105*, 2023.

Itay Evron, Edward Moroshko, Rachel Ward, Nathan Srebro, and Daniel Soudry. How catastrophic can catastrophic forgetting be in linear regression? In *Conference on Learning Theory*, pp. 4028–4079. PMLR, 2022.

Itay Evron, Edward Moroshko, Gon Buzaglo, Maroun Khriesh, Badea Marjieh, Nathan Srebro, and Daniel Soudry. Continual learning in linear classification on separable data. *arXiv preprint arXiv:2306.03534*, 2023.

Peter Gilbert. Iterative methods for the three-dimensional reconstruction of an object from projections. *Journal of theoretical biology*, 36(1):105–117, 1972.

Daniel Goldfarb and Paul Hand. Analysis of catastrophic forgetting for random orthogonal transformation tasks in the overparameterized regime. In *International Conference on Artificial Intelligence and Statistics*, pp. 2975–2993. PMLR, 2023.

Xinran Gu, Kaifeng Lyu, Longbo Huang, and Sanjeev Arora. Why (and when) does local SGD generalize better than SGD? In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=svCcui6Drl.

Xinran Gu, Kaifeng Lyu, Sanjeev Arora, Jingzhao Zhang, and Longbo Huang. A quadratic synchronization rule for distributed deep learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=yroyhkhWS6.

Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Characterizing implicit bias in terms of optimization geometry. In *International Conference on Machine Learning*, pp. 1832–1841. PMLR, 2018a.

Suriya Gunasekar, Jason D Lee, Daniel Soudry, and Nati Srebro. Implicit bias of gradient descent on linear convolutional networks. *Advances in neural information processing systems*, 31, 2018b.

Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

Baihe Huang, Xiaoxiao Li, Zhao Song, and Xin Yang. Fl-ntk: A neural tangent kernel-based framework for federated learning analysis. In *International Conference on Machine Learning*, pp. 4423–4434. PMLR, 2021.

Yanping Huang, Youlong Cheng, Ankur Bapna, Orhan Firat, Dehao Chen, Mia Chen, HyoukJoong Lee, Jiquan Ngiam, Quoc V Le, Yonghui Wu, et al. Gpipe: Efficient training of giant neural networks using pipeline parallelism. *Advances in neural information processing systems*, 32, 2019.

Sami Jaghouar, Jack Min Ong, and Johannes Hagemann. Opendiloco: An open-source framework for globally distributed low-communication training. *arXiv preprint arXiv:2407.07852*, 2024.

Divyansh Jhunjhunwala, Shiqiang Wang, and Gauri Joshi. Fedexp: Speeding up federated averaging via extrapolation. In *The Eleventh International Conference on Learning Representations*, 2023.

Ziwei Ji and Matus Telgarsky. The implicit bias of gradient descent on nonseparable data. In *Conference on Learning Theory*, pp. 1772–1798. PMLR, 2019a.

Ziwei Ji and Matus Telgarsky. Gradient descent aligns the layers of deep linear networks. In *International Conference on Learning Representations*, 2019b.

Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Tuo Zhao. Smart: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 2177–2190, 2020.

Hyunji Jung, Hanseul Cho, and Chulhee Yun. Convergence and implicit bias of gradient descent on continual linear classification. In *The Thirteenth International Conference on Learning Representations*, 2025.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pp. 5132–5143. PMLR, 2020.

Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. Tighter theory for local sgd on identical and heterogeneous data. In *International Conference on Artificial Intelligence and Statistics*, pp. 4519–4529. PMLR, 2020.

Anastasia Koloskova, Nicolas Loizou, Sadra Boreiri, Martin Jaggi, and Sebastian Stich. A unified theory of decentralized SGD with changing topology and local updates. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5381–5393. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/koloskova20a.html.

Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2661–2671, 2019.

Yiwen Kou, Zixiang Chen, and Quanquan Gu. Implicit bias of gradient descent for two-layer relu and leaky relu networks on nearly-orthogonal data. *Advances in Neural Information Processing Systems*, 36, 2024.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020a.

Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning*, pp. 6357–6368. PMLR, 2021.

Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of FedAvg on non-IID data. In *International Conference on Learning Representations*, 2020b.

Sen Lin, Peizhong Ju, Yingbin Liang, and Ness Shroff. Theory on forgetting and generalization of continual learning. In *International Conference on Machine Learning*, pp. 21078–21100. PMLR, 2023.

Tao Lin, Sebastian U Stich, Kumar Kshitij Patel, and Martin Jaggi. Don't use large mini-batches, use local SGD. In *International Conference on Learning Representations*, 2019.

Shruti P Maralappanavar, Prashant Khanduri, and Bharath B N. Linear convergence of decentralized fedavg for PL objectives: The interpolation regime. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=Og3VxBFhwj.

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

Mor Shpigel Nacson, Nathan Srebro, and Daniel Soudry. Stochastic gradient descent on separable data: Exact convergence with a fixed learning rate. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 3051–3059. PMLR, 2019.

Kumar Kshitij Patel, Margalit Glasgow, Ali Zindari, Lingxiao Wang, Sebastian U Stich, Ziheng Cheng, Nirmit Joshi, and Nathan Srebro. The limits and potentials of local sgd for distributed heterogeneous learning with intermittent communication. In Shipra Agrawal and Aaron Roth (eds.), *Proceedings of Thirty Seventh Conference on Learning Theory*, volume 247 of *Proceedings of Machine Learning Research*, pp. 4115–4157. PMLR, 30 Jun–03 Jul 2024. URL https://proceedings.mlr.press/v247/patel24a.html.

Tiancheng Qin, S Rasoul Etesami, and César A Uribe. Faster convergence of local sgd for over-parameterized models. *arXiv preprint arXiv:2201.12719*, 2022.

Sashank J Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and Hugh Brendan McMahan. Adaptive federated optimization. In *International Conference on Learning Representations*, 2021.

Hamza Reguieg, Mohammed El Hanjri, Mohamed El Kamili, and Abdellatif Kobbane. A comparative evaluation of fedavg and per-fedavg algorithms for dirichlet distributed heterogeneous data. In *2023 10th International Conference on Wireless Networks and Mobile Communications (WINCOM)*, pp. 1–6. IEEE, 2023.

Alexander Sergeev and Mike Del Balso. Horovod: fast and easy distributed deep learning in tensorflow. *arXiv preprint arXiv:1802.05799*, 2018.

Bingqing Song, Prashant Khanduri, Xinwei Zhang, Jinfeng Yi, and Mingyi Hong. Fedavg converges to zero training loss linearly for overparameterized multi-layer neural networks. In *International Conference on Machine Learning*, pp. 32304–32330. PMLR, 2023.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878, 2018.

Sebastian U Stich. Local SGD converges fast and communicates little. In *International Conference on Learning Representations*, 2019.

Canh T Dinh, Nguyen Tran, and Josh Nguyen. Personalized federated learning with moreau envelopes. *Advances in Neural Information Processing Systems*, 33:21394–21405, 2020.

Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems*, 33:7611–7623, 2020.

Jianyu Wang, Rudrajit Das, Gauri Joshi, Satyen Kale, Zheng Xu, and Tong Zhang. On the unreasonable effectiveness of federated averaging with heterogeneous data. *Transactions on Machine Learning Research*, 2024.

Blake Woodworth, Kumar Kshitij Patel, Sebastian Stich, Zhen Dai, Brian Bullins, Brendan Mcmahan, Ohad Shamir, and Nathan Srebro. Is local SGD better than minibatch SGD? In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 10334–10343. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/woodworth20a.html.

Shuo Xie and Zhiyuan Li. Implicit bias of adamw: ℓ_∞-norm constrained optimization. In *International Conference on Machine Learning*, pp. 54488–54510. PMLR, 2024.

Hao Yu, Sen Yang, and Shenghuo Zhu. Parallel restarted sgd with faster convergence and less communication: Demystifying why model averaging works for deep learning. In *Proceedings of the AAAI conference on artificial intelligence*, pp. 5693–5700, 2019.

Chenyang Zhang, Difan Zou, and Yuan Cao. The implicit bias of adam on separable data. *Advances in Neural Information Processing Systems*, 37:23988–24021, 2024.

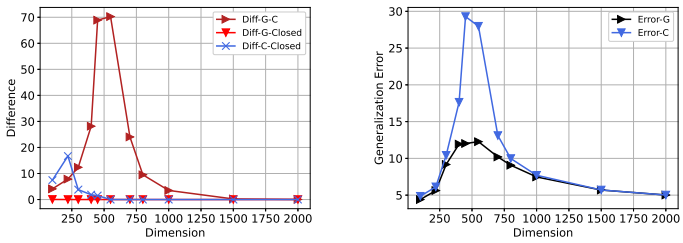
CONTENTS

A Additional Experiments

A.1 Experiments on Linear Regression

We simulated 10 compute nodes, each with 50 training samples. The label vector y_i at i-th compute node is exactly generated as (13), where ground truth model w_i^* is Gaussian vector with each element following $\mathcal{N}(0,4)$. Each ground truth model at different compute nodes is independently generated, thus the datasets can be very different from each other. The data matrix X_i also follows Gaussian distribution, with each element being $\mathcal{N}(0,1)$, and z_i is a Gaussian vector with $\mathcal{N}(0,0.04)$. In Local-GD, the number of local steps is $L=200$, number of rounds is also $R=200$, and the learning rate $\eta=0.0001$. Actually it just take a few local steps to converge locally at each round, but we set a large number of local steps to show it can be large at $O(\sqrt{T})$, where $T=L*R$ is the number of total iterations. We tested the global model (G) from Local-GD on squared loss, centralized model (C) trained from global dataset on squared loss, closed form of global model (G-Closed) in (17), closed form of centralized model (C-Closed) as solution of problem (18). The centralized model is trained 10000 steps with learning rate 0.0001.

Fig. 2(a) displays the difference between global model and centralized model, global model and its closed form, and centralized model and its closed form, with respect to model dimension. The difference between two models is $\|w_1-w_2\|/d$. Since it is always locally overparameterized, the difference between global model and the closed form is always zero. The difference between global model and centralized model has an obvious peak around 500, which is the number of total samples. The phenomenon that global model converges exactly to centralized model only happens when the model is sufficiently overparameterized. Fig. 2(b) shows the generalization error of global model and centralized model in linear regression. Since the data matrix is Gaussian, the generalization error of model w can be computed as $\frac{1}{M}\sum_{i=1}^{M}\|w-w_i^*\|^2$. We plot the generalization error divided by d. It is shown the global model and centralized model can get the same performance when model is sufficiently overparameterized.

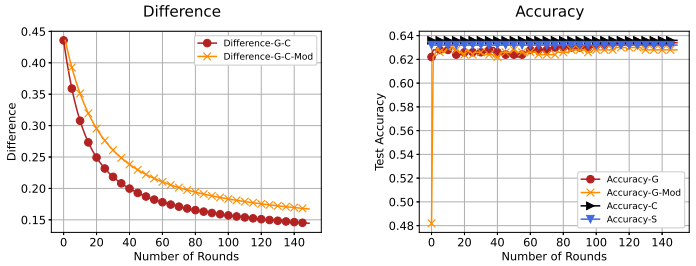


(a) (b)

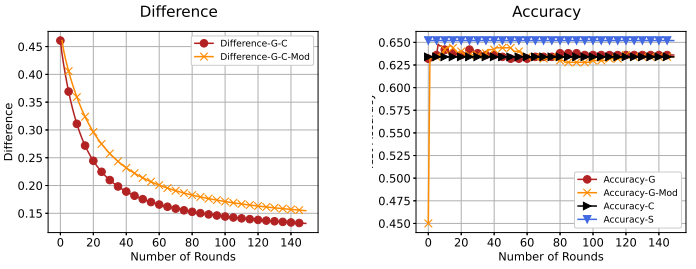
Figure 2: (a) Difference between global and centralized models plotted against increasing dimension. (b) Generalization error with respect to dimension.

A.2 Linear Classification with Dirichlet Distribution

In federated learning, the Dirichlet distribution is usually used to generate heterogeneous datasets across the compute nodes Hsu et al. (2019); Chen & Chao (2021); Reguieg et al. (2023). For binary classification problem, the Dirichlet distribution Dir(α) is used to unbalance the positive and negative samples. In the experiments we have 10 compute nodes. We generate 500 samples as $y_i=\text{sign}(x_i^T w^*)$ for $i\in[500]$ and use Dir(α) to distribute the 500 samples across 10 compute nodes. Note that the number of samples at each compute node is not necessarily identical. Fig. 3 shows performance of Local-GD for linear classification with different parameter α in Dirichlet distribution. The λ is set to be 0.0001 and model dimension is fixed as $d=1500$. The number of local steps L is 150 and number of communication rounds R is 150. The learning rate is 0.01. The centralized model is trained with the same learning rate for 22500 steps. We can see the global model and modified global model still converge to the centralized model in direction and get similar test accuracy.



(a) Difference with $\alpha = 0.5$. (b) Test Accuracy with $\alpha = 0.5$.



(c) Difference with $\alpha = 0.3$. (d) Test Accuracy with $\alpha = 0.3$.

Figure 3: Local-GD on linear classification with Dirichlet distribution.

B Local-GD for Linear Regression in Overparameterized Regime

In this section we give a extended description of Section 2 about linear regression in overparameterized regime.

B.1 Setting

The behavior of linear regression is very well-understood in high-dimensional statistics; and we can clearly convey our key message based on this fundamental setting.

At each compute node i, the dataset S_i consists of N tuples of samples and their corresponding labels, $(x,y) \in \mathbb{R}^d \times \mathbb{R}$. We assume the label y_{ij} is generated by

$$y_{ij} = x_{ij}^T w_i^* + z_{ij} \tag{13}$$

where $w_i^* \in \mathbb{R}^d$ is the ground truth model at i-th compute node, and z_{ij} is the added noise. Denote $X_i = [x_{i1}, x_{i2}, ..., x_{iN}]^T \in \mathbb{R}^{N \times d}$ as the data matrix at i-th compute node, and $y_i = [y_{i1}, y_{i2}, ..., y_{iN}] \in \mathbb{R}^N$ as the label vector, $z_i \in \mathbb{R}^N$ as the noise vector. In heterogeneous setting, the w_i^* can be very different to each other. Note that the convergence to centralized model does not rely on the generative model. We just make this assumption on generative model for deriving a more clear form of the aggregated global model.

Algorithm. At each round, the aggregator sends the global model w_0 to all the compute nodes. Each compute node minimizes the squared loss $f_i(w_i) = \frac{1}{2N} \|y_i - X_i w_i\|^2$ by a large number of gradient descent steps *until convergence*. Then each compute node sends back the local model and the aggregator aggregates all the local models to get the updated global model. The detailed algorithm is Local-GD in Algorithm 1 with $f_i(w_i)$ replaced in the update. Since minimizing squared loss is a quadratic problem, it is expected to reach convergence locally with a small number of gradient descent steps.

B.2 Implicit Bias of Local GD in Linear Regression

For each local problem, when the dimension of the model is larger than the number of samples at each compute node ($d > N$), i.e., locally overparameterized, there are multiple solutions corresponding to zero squared loss. However, gradient descent will lead the model converge to a specific solution, which corresponds to a minimum Euclidean distance to the initial point Gunasekar et al. (2018a); Evron et al. (2022). Formally, the

solution w_i^{k+1} obtained at k-th round and i-th node will converge to the solution of the optimization problem

$$\min_{w_i} \quad \|w_i - w_0^k\|^2 \quad \text{s.t.} \quad X_i w_i = y_i. \tag{14}$$

We can obtained the closed form solution of this optimization problem as (see Proof of Lemma 4 in Appendix B.4.1)

$$\begin{aligned} w_i^{k+1} &= \big(I - X_i^T (X_i X_i^T)^{-1} X_i\big) w_0^k + X_i^T (X_i X_i^T)^{-1} y_i \\ &= \big(I - X_i^T (X_i X_i^T)^{-1} X_i\big) w_0^k \\ &\quad + X_i^T (X_i X_i^T)^{-1} X_i w_i^* + X_i^T (X_i X_i^T)^{-1} z_i. \end{aligned} \tag{15}$$

Denote $P_i \triangleq X_i^T (X_i X_i^T)^{-1} X_i$ and $X_i^\dagger \triangleq X_i^T (X_i X_i^T)^{-1}$. The local model can be rewritten as $w_i^{k+1} = (I - P_i) w_0^k + P_i w_i^* + X_i^\dagger z_i$. We observe that P_i is the projection operator to the row space of X_i, and $X_i^\dagger$ is the pseudo inverse of X_i. After one round of iterations, the local model is actually an interpolation between the initial global model w_0^k at this round and the ground-truth model w_i^*, plus a noise term. We then obtain the closed form of global model by aggregation. After many rounds of communication, we can obtain the final trained global model from Local-GD.

Lemma 4. *When the local overparameterized linear regression problems are exactly solved by gradient descent, then after K rounds of communication, the global model w_0^K obtained from Local-GD is*

$$w_0^K = (I - \bar{P})^K w_0^0 + \sum_{k=0}^{K-1} (I - \bar{P})^k (\bar{Q} + \bar{Z}), \tag{16}$$

where $\bar{P} = \frac{1}{M}\sum_{i=1}^M P_i, \bar{Q} = \frac{1}{M}\sum_{i=1}^M P_i w_i^*, \bar{Z} = \frac{1}{M}\sum_{i=1}^M X_i^\dagger z_i$.

Note that $\bar{P}, \bar{Q}, \bar{Z}$ are constant after the data is generated. Since we only know the $\{X_i, y_i\}_{i=1}^M$ in the training process, we can also write it as

$$w_0^K = (I - \bar{P})^K w_0^0 + \sum_{k=0}^{K-1} (I - \bar{P})^k \bar{Y}, \tag{17}$$

where $\bar{Y} = \frac{1}{M}\sum_{i=1}^M X_i^\dagger y_i$. Then we can directly get the final model from the training set.

Singularity of $\bar{P}$. If $\bar{P}$ is invertible, we can further simplify the form of global model. However, since $P_i \in \mathbb{R}^{d\times d}$ is the projection operator onto row space of X_i, its rank is at most N. The $\bar{P}$ is the average of P_is, thus its rank is at most MN. Note that we consider the overparameterized regime both locally and globally, i.e., $d \gg MN$. Then $\bar{P}$ is singular, and the sum $\sum_{k=0}^{K-1}(I - \bar{P})^k$ approaches KI when d becomes very large. We cannot get more properties of the final global model from (17), but we can compare it to the centralized model trained with all of the data.

B.3 Convergence to Centralized Model

Let $X_c = [X_1^T, \dots, X_M^T]^T \in \mathbb{R}^{MN\times d}$ be the data matrix consisting of all the local data, and $y_c = [y_1^T, \dots, y_M^T]^T \in \mathbb{R}^{MN\times 1}$ be the label vector consisting of the local labels. If we train the centralized model from initial point 0 with squared loss, then the gradient descent will lead the model to the solution of the optimization problem

$$\min_{w} \quad \|w\|^2 \quad \text{s.t.} \quad X_c w = y_c \tag{18}$$

We can write the closed form of centralized model as $w_c = X_c^T (X_c X_c^T)^{-1} y_c$.

Due to the constraint in problem (18), for each compute node i, we have $X_i w_c = y_i$. We replace y_i in the local model (15), then we have

$$w_i^{k+1} - w_c = (I - P_i)(w_0^k - w_c). \tag{19}$$

The RHS is projecting the difference between global model and centralized model onto null space of X_i. After averaging all the local models at the aggregator, we have

$$w_0^{k+1} - w_c = (I - \bar{P})(w_0^k - w_c). \tag{20}$$

In the training process the difference between global model and centralized model is iteratively projected onto the null space of span of row spaces of X_is. It implies that the difference on the span of data matrix gradually decreases until zero. Based on the evolution of the difference, we can prove the Theorem 1 and we restate it here:

Theorem 5. *For the linear regression problem, suppose the initial point w_0^0 is 0 and $d > MN$ and the minimum eigenvalue of $\bar{P}$, $\lambda_{\min}$ is larger than 0, then the global model obtained by Local-GD, w_0^K, converges to the centralized solution w_c as the number of communication rounds $K \to \infty$ as $\|w_0^K - w_c\| \leq (1-\lambda_{\min})^K \|w_c\|$.*

The proof is in Appendix B.4.2. The key step is to show the initial difference is already in the data space, and no residual in the null space of row spaces of X_is. The convergence to the centralized model is at exponential rate.

Due to the linearity of the regression problem, we can theoretically show the global model can exactly converge to the centralized model with implicit bias on overparameterized regime. Note that the proof does not rely on the generative model and assumption on data heterogeneity. It implies that, even if we use a large number of local steps to exactly solve the local problems on very heterogeneous data, the performance of Local-GD is equivalent to train a model with all the data in one place.

B.4 Proofs in Linear Regression

B.4.1 Proof of Lemma 4

At each compute node, the local model converges to the solution of problem

$$\min_{w_i} \quad \|w_i - w_0^k\|^2 \quad \text{s.t.} \quad X_i w_i = y_i. \tag{21}$$

Using Lagrange multipliers, we can write the Lagrangian as

$$\frac{1}{2}\|w_i - w_0^k\|^2 + \beta^T (X_i w_i - y_i) \tag{22}$$

Setting the derivative to 0, we know the optimal $\tilde{w}_i$ satisfies

$$\tilde{w}_i - w_0^k + X_i^T \beta = 0, \tag{23}$$

and then

$$\tilde{w}_i = w_0^k - X_i^T \beta. \tag{24}$$

Also by the constraint $y_i = X_i \tilde{w}_i$, we can get

$$y_i = X_i w_0^k - (X_i X_i^T)\beta. \tag{25}$$

Since the model is overparameterized ($d > N$), $X_i X_i^T \in \mathbb{R}^{d \times d}$ is invertible. Then we have

$$\beta = -(X_i X_i^T)^{-1}(y_i - X_i w_0^k). \tag{26}$$

Plugging the β back, we can get the closed form solution as

$$\tilde{w}_i = w_0^k + X_i^T (X_i X_i^T)^{-1}(y_i - X_i w_0^k). \tag{27}$$

We update the local model $w_i^{k+1} = \tilde{w}_i$.

We can also write the closed form solution as

$$\begin{aligned} w_i^{k+1} &= w_0^k + X_i^T (X_i X_i^T)^{-1}(y_i - X_i w_0^k) \\ &= \big(I - X_i^T (X_i X_i^T)^{-1} X_i\big) w_0^k + X_i^T (X_i X_i^T)^{-1} y_i \end{aligned} \tag{28}$$

If we plug in the generative model $y_i = X_i w_i^* + z_i$, then the solution is

$$\begin{aligned} w_i^{k+1} &= \big(I - X_i^T (X_i X_i^T)^{-1} X_i\big) w_0^k + X_i^T (X_i X_i^T)^{-1} X_i w_i^* + X_i^T (X_i X_i^T)^{-1} z_i \\ &= (I - P_i) w_0^k + P_i w_i^* + X_i^\dagger z_i. \end{aligned} \tag{29}$$

where $P_i = X_i^T (X_i X_i^T)^{-1} X_i$ is the projection operator to the row space of X_i, and $X_i^\dagger = X_i^T (X_i X_i^T)^{-1}$ is the pseudo inverse of X_i. It is an interpolation between the initial global model w_0^k and the local true model w_i^*, plus a noise term.

After aggregating all the local models, the global model is

$$\begin{aligned} w_0^{k+1} &= \frac{1}{m}\sum_{i=1}^{m}(I-P_i)w_0^k + \frac{1}{m}\sum_{i=1}^{m}P_i w_i^* + \frac{1}{m}\sum_{i=1}^{m}X_i^\dagger z_i \\ &= (I-\bar{P})w_0^k + \bar{Q} + \bar{Z}, \end{aligned} \tag{30}$$

where $\bar{P} = \frac{1}{m}\sum_{i=1}^m P_i, \bar{Q} = \sum_{i=1}^m P_i w_i^*, \bar{Z} = \frac{1}{m}\sum_{i=1}^m X_i^\dagger z_i$.

After K rounds of communication, the global model is

$$w_0^K = (I-\bar{P})^K w_0^0 + \sum_{k=0}^{K-1}(I-\bar{P})(\bar{Q}+\bar{Z}). \tag{31}$$

If we start from $w_0^0 = 0$, then the solution will converge to $\sum_{k=0}^{K-1}(I-\bar{P})(\bar{Q}+\bar{Z})$.

B.4.2 Proof of Theorem 1

We know the difference between global model and centralized model is iteratively projected onto the null space of span of row spaces of X_is:

$$w_0^{k+1} - w_c = (I-\bar{P})(w_0^k - w_c). \tag{32}$$

We can formally describe it as follows. Since the problem is overparameterized globally, we can assume each X_i has full rank N. We apply singular value decomposition (SVD) to X_i as $X_i = U_i \Sigma_i V_i^T$, where $U_i \in \mathbb{R}^{N\times N}, V_i \in \mathbb{R}^{d\times N}$. Then $P_i = X_i^T (X_i X_i^T)^{-1} X_i = V_i V_i^T$, which is the projection matrix to the row space of X_i.

We apply eigenvalue decomposition on $\bar{P}$ to get $\bar{P} = Q\Sigma Q^T$, where $Q \in \mathbb{R}^{d\times n'}$ and n' is the rank of $\bar{P}$. It satisfies $N \le n' \le MN$. Since $\bar{P}$ is a linear combination of P_is, the space of column space of Q is the space spanned by all the vectors $v_{ij}, i = 1,...,M, j = 1,...,N$.

We also construct a matrix $Q' \in \mathbb{R}^{d\times(d-n')}$, which consists of orthonomal vectors perpendicular to Q. We can project the difference onto column space of Q and Q' respectively.

$$\begin{aligned} Q^T(w_0^{k+1} - w_c) &= Q^T(I - Q\Sigma Q^T)(w_0^k - w_c) = (I-\Sigma)Q^T(w_0^k - w_c) \\ Q'^T(w_0^{k+1} - w_c) &= Q'^T(I - Q\Sigma Q^T)(w_0^k - w_c) = Q'^T(w_0^k - w_c) \end{aligned} \tag{33}$$

After K rounds of communication, we can decomposite $w_0^K - w_c$ into two parts:

$$w_0^K - w_c = QQ^T(w_0^K - w_c) + Q'Q'^T(w_0^K - w_c). \tag{34}$$

Then we can obtain

$$\begin{aligned} w_0^K - w_c &= QQ^T(w_0^K - w_c) + Q'Q'^T(w_0^K - w_c) \\ &= Q(I-\Sigma)^K Q^T(w_0^0 - w_c) + Q'Q'^T(w_0^0 - w_c). \end{aligned}$$

It shows the initial difference on the column space of Q continues to decrease until zero if K is sufficiently large. And the initial difference on the null space of Q remains constant.

To show the difference $w_0^K - w_c$ goes to zero entirely, we just need to choose an initial point such that initial difference is on the column space of Q. When we choose $w_0^0 = 0$, the initial difference is w_c itself. Moreover, the centralized solution $w_c = X_c^T (X_c X_c^T)^{-1} y_c$ exactly lies in the data space spanned by vectors $\{v_{ij}\}_{i=1,j=1}^{M,N}$ since it is a linear combination of columns of X_c^T. So if we start from $w_0^0 = 0$, then $w_0^K - w_c$ will go to zero when K is sufficiently large.

When starting from 0, the difference between the global model and the centralized model becomes

$$\begin{aligned}\|w_0^K - w_c\|^2 &= \|Q(I-\Sigma)^K Q^T w_c\|^2 \\ &= \big(Q(I-\Sigma)^K Q^T w_c\big)^T \big(Q(I-\Sigma)^K Q^T w_c\big) \\ &= \big(Q^T w_c\big)^T (I-\Sigma)^{2K} \big(Q^T w_c\big). \end{aligned} \tag{35}$$

Since $I-\Sigma$ is a diagonal matrix, we can get

$$\|w_0^K - w_c\|^2 \leq (1-\lambda_{\min})^{2K} \|Q^T w_c\|^2, \tag{36}$$

where $\lambda_{\min}$ is the minimum eigenvalue of matrix $\bar{P}$. Also since Q is an orthogonal matrix, we have $\|Q^T w_c\|^2 = \|w_c\|^2$. Then we can get

$$\|w_0^K - w_c\| \leq (1-\lambda_{\min})^K \|w_c\|. \tag{37}$$

It shows the difference between trained global model and centralized model converge to zero at an exponential rate.

C Proofs of Implicit Bias for Linear Classification in Section 3

We give the detailed proofs of Theorem 2 in this section. The proof framework is inspired by the analysis of implicit bias of SGD Nacson et al. (2019). Intuitively, we can regard one local dataset as a "batch" in SGD for sampling without replacement. But we perform multiple gradient steps in the same "batch", not just one step of gradient descent. The challenge is to handle local steps in the same local dataset and the aggregation after one round of local training. Here we restate the Theorem 2.

Theorem 6. *Under assumptions 1, 2, 3, if the learning rate satisfies* $\eta \leq \min\Big(\frac{1}{2L\sigma_{max}^2\beta}, \frac{\gamma^2}{4L\sigma_{\max}^3\beta(\gamma+\sigma_{\max})}\Big)$, *then for the process of Local-GD, we have,*

- ***Claim 1****: Every data point is classified correctly finally:* $\lim_{k\to\infty} x_s^T w_0^k = \infty, \forall s \in S$.
- ***Claim 2****: The global model obtained from Local-GD will behave as*

$$w_0^k = \log(Lk)\hat{w} + \rho^k, \quad \textit{and,} \quad \left\| \frac{w_0^k}{\|w_0^k\|} - \frac{\hat{w}}{\|\hat{w}\|} \right\| = O\Big(\frac{1}{\log Lk}\Big) \tag{38}$$

  and $\|\rho^k\| < \infty$ *for all* k. *This implies, the normalized global model converges to the global max-margin solution.*

- ***Claim 3****: The loss function* $f(w_0^k)$ *decreases to zero as* $f(w_0^k) = O\big(\frac{1}{Lk}\big)$.

For the three claims in Theorem 2, we will give separable (but sequential) proofs below.In the proofs of linear classification, for ease of notation, we redefine the samples $y_s x_s$ to x_s to subsume the labels.

C.1 Proof of Claim 1

In this proof, we rely on the key property of linearly separable data.

Lemma 5 (Lemma 2 and (17) in Nacson et al. (2019))**.** *Suppose that Assumptions 1 and 2 hold. For any* $w \in \mathbb{R}^d$,

$$\|\nabla f(w)\| \geq \frac{\gamma}{M} \sqrt{\sum_{s\in S} [g'(x_s^T w)]^2}.$$

Lemma 6. *Suppose that Assumptions 1 and 2 hold and* $k \in \mathbb{N}$. *Then we have*

$$\|w_i^{k,l} - w_0^k + \eta\big(l\nabla f_i(w_0^k)\big)\| \leq \frac{\eta^2 L\sigma_{\max}^3 \beta M l}{\gamma(1 - l\eta\beta\sigma_{\max}^2)} \|\nabla f(w_0^k)\|. \tag{39}$$

$$\|w_i^{k,l} - w_0^k\| \leq \frac{\eta L\sigma_{\max} M}{\gamma(1 - l\eta\beta\sigma_{\max}^2)} \|\nabla f(w_0^k)\|. \tag{40}$$

$$\|\nabla f(w_i^{k,l}) - \nabla f(w_0^k)\| \leq \frac{\eta L\sigma_{\max}^3 \beta M}{\gamma(1 - l\eta\beta\sigma_{\max}^2)} \|\nabla f(w_0^k)\|. \tag{41}$$

The proof can be seen in Section C.1.1.

Note that $f(w) = \frac{1}{M}\sum_{i=1}^M f_i(w) = \frac{1}{M}\sum_{s\in S} g(x_s^T w)$, and $g(u)$ is a β-smooth function from Assumption 2. Then $f(w)$ is a $\frac{\beta\sigma_{\max}^2}{M}$-smooth function. Then we can get

$$\begin{aligned}
&f(w_0^{k+1}) - f(w_0^k) - \frac{\sigma_{\max}^2\beta}{2M}\|w_0^{k+1} - w_0^k\|^2 \\
\leq& \langle \nabla f(w_0^k), (w_0^{k+1} - w_0^k)\rangle \\
=& \langle \nabla f(w_0^k), w_0^{k+1} - w_0^k - \eta L\nabla f(w_0^k) + \eta L\nabla f(w_0^k)\rangle \\
\leq& -\eta L\|\nabla f(w_0^k)\|^2 + \|\nabla f(w_0^k)\|\|w_0^{k+1} - w_0^k + \eta L\nabla f(w_0^k)\|,
\end{aligned} \tag{42}$$

where the second inequality is from Cauchy-Schwarz inequality.

For the second term, we have

$$\begin{aligned}
&\|w_0^{k+1}-w_0^k+\eta L\nabla f(w_0^k)\| \\
=&\|\frac{1}{M}\sum_{i=1}^{M}w_i^{k+1}-w_0^k+\eta L\frac{1}{M}\sum_{i=1}^{M}\nabla f_i(w_0^k)\| \\
\leq&\frac{1}{M}\sum_{i=1}^{M}\|w_i^{k+1}-w_0^k+\eta L\nabla f_i(w_0^k)\| \\
\leq&\frac{1}{M}\sum_{i=1}^{M}\frac{\eta^2L^2\sigma_{\max}^3\beta M}{\gamma(1-L\eta\beta\sigma_{\max}^2)}\|\nabla f(w_0^k)\| \\
=&\frac{\eta^2L^2\sigma_{\max}^3\beta M}{\gamma(1-L\eta\beta\sigma_{\max}^2)}\|\nabla f(w_0^k)\| \qquad (43)
\end{aligned}$$

where the first inequality is triangle inequality and second inequality is from Lemma 6.

We also have

$$\begin{aligned}
\|w_0^{k+1}-w_0^k\|^2=&\|\frac{1}{M}\sum_{i=1}^{M}w_i^{k+1}-w_0^k\|^2 \\
\leq&\frac{1}{M}\sum_{i=1}^{M}\|w_i^{k+1}-w_0^k\|^2 \\
\leq&\frac{\eta^2L^2\sigma_{\max}^2M^2}{\gamma^2(1-L\eta\beta\sigma_{\max}^2)^2}\|\nabla f(w_0^k)\|^2 \qquad (44)
\end{aligned}$$

where the second inequality is from Lemma 6. Plug above two inequalities into (42), we can get

$$f(w_0^{k+1})-f(w_0^k)\leq-\eta L\left(1-\frac{\eta L\sigma_{\max}^3\beta M}{\gamma(1-L\eta\beta\sigma_{\max}^2)}-\frac{\eta L\sigma_{\max}^4\beta M}{2\gamma^2(1-L\eta\beta\sigma_{\max}^2)^2}\right)\|\nabla f(w_0^k)\|^2 \qquad (45)$$

If we choose $\eta\leq\frac{1}{2L\sigma_{\max}^2\beta}$, then $\frac{1}{1-L\eta\beta\sigma_{\max}^2}\leq 2$. Thus we can obtain

$$\begin{aligned}
f(w_0^{k+1})-f(w_0^k)\leq&-\eta L\left(1-\eta L\sigma_{\max}^3\beta M(\frac{2}{\gamma}+\frac{2\sigma_{\max}}{\gamma^2})\right)\|\nabla f(w_0^k)\|^2 \\
=&-\eta L(1-\eta L\beta')\|\nabla f(w_0^k)\|^2 \qquad (46)
\end{aligned}$$

where $\beta'=\frac{2\sigma_{\max}^3\beta M(\gamma+\sigma_{\max})}{\gamma^2}$.

If we also choose $\eta\leq\frac{1}{2L\beta'}$, then

$$f(w_0^{k+1})-f(w_0^k)\leq-\frac{\eta L}{2}\|\nabla f(w_0^k)\|^2, \qquad (47)$$

which means the loss continues to decrease.

Combining the two condition on step size, we require

$$\eta\leq\min\left(\frac{1}{2L\sigma_{max}^2\beta},\frac{\gamma^2}{4L\sigma_{\max}^3\beta M(\gamma+\sigma_{\max})}\right). \qquad (48)$$

Summing up from $k=0$ to ∞, we have

$$\sum_{k=0}^{\infty}\|\nabla f(w_0^k)\|^2\leq\frac{2(f(w_0^0)-f(w_0^\infty))}{\eta L}\leq\frac{2f(w_0^0)}{\eta L}<\infty \qquad (49)$$

The boundedness means $\lim_{k\to\infty}\|\nabla f(w_0^k)\|^2=0$. From Lemma 5, we can also know $\lim_{k\to\infty}g'(x_s^Tw_0^k)=0,\forall s\in S$. From Assumption 2, $g'(u)\to 0$ only when $u\to\infty$, thus $x_s^Tw_0^k\to\infty,\forall s\in S$, which means all the training samples can be correctly classified. This proves Claim 1 in Theorem 2.

We also bound the change of weights across iterations here, which is useful in the proof of Claim 2. since $\nabla f_i(w)=\sum_{s\in S_i} g'(x_s^T w)x_s$ we can have

$$\begin{aligned}\frac{1}{M}\sum_{i=1}^{M}\|w_i^{k,l+1}-w_i^{k,l}\|&=\frac{1}{M}\sum_{i=1}^{M}\eta\|\nabla f_i(w_i^{k,l})\|\\&=\frac{1}{M}\sum_{i=1}^{M}\eta\|\sum_{s\in S_i} g'(x_s^T w_i^{k,l}x_s)\|\\&\le\frac{1}{M}\sum_{i=1}^{M}\eta\sigma_{\max}\sqrt{\sum_{s\in S_i}\Big(g'(x_s^T w_i^{k,l})\Big)^2}\\&\le\frac{1}{M}\sum_{i=1}^{M}\eta\sigma_{\max}\sqrt{\sum_{s\in S}\Big(g'(x_s^T w_i^{k,l})\Big)^2}\\&\le\frac{\eta\sigma_{\max}}{\gamma}\sum_{i=1}^{M}\|\nabla f(w_i^{k,l})\|,\end{aligned}\tag{50}$$

where the first inequality is from the fact $\|\sum_{s\in S}a_s x_s\|\le\sigma_{\max}\sqrt{\sum_{s\in S}a_s^2}$ for $\forall a_s\in\mathbb{R}$, the second inequality is due to $S_i\subset S$, and the final inequality is from Lemma 5. Further we can obtain

$$\begin{aligned}\|\nabla f(w_i^{k,l})\|&\le\|\nabla f(w_0^k)\|+\|\nabla f(w_i^{k,l})-\nabla f(w_0^t))\|\\&\le\|\nabla f(w_0^k)\|+\frac{\eta L\sigma_{\max}^3\beta M}{\gamma(1-l\eta\beta\sigma_{\max}^2)}\|\nabla f(w_0^k)\|\\&=\left(1+\frac{\eta L\sigma_{\max}^3\beta M}{\gamma(1-l\eta\beta\sigma_{\max}^2)}\right)\|\nabla f(w_0^k)\|\end{aligned}\tag{51}$$

where the second inequality is from Lemma 6. Then we have

$$\begin{aligned}\frac{1}{M}\sum_{i=1}^{M}\|w_i^{k,l+1}-w_i^{k,l}\|^2&\le\frac{1}{M}\sum_{i=1}^{M}\frac{\eta^2\sigma_{\max}^2M^2}{\gamma^2}\left(1+\frac{\eta L\sigma_{\max}^3\beta M}{\gamma(1-l\eta\beta\sigma_{\max}^2)}\right)^2\|\nabla f(w_0^k)\|^2\\&\le\frac{\eta^2\sigma_{\max}^2M^2}{\gamma^2}\left(1+\frac{\eta L\sigma_{\max}^3\beta M}{\gamma(1-L\eta\beta\sigma_{\max}^2)}\right)^2\|\nabla f(w_0^k)\|^2\end{aligned}\tag{52}$$

Summing up all the changes, we can finally have

$$\frac{1}{M}\sum_{k=0}^{\infty}\sum_{l=1}^{L-1}\sum_{i=1}^{M}\|w_i^{k,l+1}-w_i^{k,l}\|^2\le\frac{\eta^2\sigma_{\max}^2LM^2}{\gamma^2}\left(1+\frac{\eta L\sigma_{\max}^3\beta M}{\gamma(1-L\eta\beta\sigma_{\max}^2)}\right)^2\sum_{k=0}^{\infty}\|\nabla f(w_0^k)\|^2<\infty.\tag{53}$$

C.1.1 Proof of Lemma 6

Proof. We start from the update rule:

$$w_i^{k,l}=w_0^k-\eta\left(\sum_{l'=0}^{l-1}\nabla f_i(w_i^{k,l'})\right).\tag{54}$$

Define $\Delta := w_i^{k,l} - w_0^k + \eta\big(l\nabla f_i(w_0^k)\big)$. Then by triangle inequality, we have

$$\begin{aligned}
\|\Delta\| &= \|-\eta\sum_{l'=0}^{l-1}\nabla f_i(w_i^{t,l'}) + \eta l\nabla f_i(w_0^k))\| \\
&= \eta\|\sum_{l'=0}^{l-1}\Big(\nabla f_i(w_i^{k,l'}) - \nabla f_i(w_0^k)\Big)\| \\
&\le \eta\sum_{l'=0}^{l-1}\|f_i(w_i^{k,l'}) - \nabla f_i(w_0^k)\| \\
&\le \eta\beta_i\sum_{l'=0}^{l-1}\|w_i^{k,l'} - w_0^k\| \qquad (55)
\end{aligned}$$

where β_i is the smoothness parameter of $f_i(w)$. Since each local dataset of a subset of global dataset, $\forall i \in [1,M], \beta_i \le \beta\sigma_{\max}$.

In addition, since $\nabla f_i(w) = \sum_{s\in S_i} g'(x_s^T w)x_s$ we can have

$$\begin{aligned}
&\|w_i^{k,l} - w_0^k\| \\
&= \|w_i^{k,l} - w_0^k + \eta l\nabla f_i(w_0^k) - \eta l\nabla f_i(w_0^k)\| \\
&\le \|w_i^{k,l} - w_0^k + \eta l\nabla f_i(w_0^k)\| + \eta\|l\sum_{s\in S_i} g'(x_s^T w_0^k)x_s\| \\
&\le \|\Delta\| + \eta l\sigma_{\max}\sqrt{\sum_{s\in S_i}\big(g'(x_s^T w_0^k)\big)^2} \\
&\le \|\Delta\| + \eta L\sigma_{\max}\sqrt{\sum_{s\in S}\big(g'(x_s^T w_0^k)\big)^2} \\
&\le \|\Delta\| + \frac{\eta L\sigma_{\max} M}{\gamma}\|f(w_0^k)\| \qquad (56)
\end{aligned}$$

where the second inequality is from the fact $\|\sum_{s\in S} a_s x_s\| \le \sigma_{\max}\sqrt{\sum_{s\in S} a_s^2}$ for $\forall a_s \in \mathbb{R}$, the third inequality is due to $S_i \subset S$, and the final inequality is from Lemma 5. Then we plug in $\|\Delta\|$ and get

$$\|w_i^{k,l} - w_0^k\| \le \eta\beta\sigma_{\max}^2\sum_{l'=0}^{l-1}\|w_i^{k,l'} - w_0^k\| + \frac{\eta L\sigma_{\max} M}{\gamma}\|f(w_0^k)\|. \qquad (57)$$

Now we use another lemma from Nacson et al. (2019):

Lemma 7 (Lemma 4 in Nacson et al. (2019)). *Let ϵ and θ be positive constants. If $\delta_k \le \theta + \epsilon\sum_{u=0}^{k-1}\delta_u$, then*

$$\delta_k \le \frac{\theta}{1-k\epsilon} \quad \textit{and} \quad \sum_{u=0}^{k-1}\delta_u \le \frac{k\theta}{1-k\epsilon}.$$

Directly applying this lemma to (57), we can obtain

$$\|w_i^{k,l} - w_0^k\| \le \frac{\eta L\sigma_{\max} M}{\gamma(1 - l\eta\beta\sigma_{\max}^2)}\|\nabla f(w_0^k)\|. \qquad (58)$$

Then we further have

$$\|\Delta\| \le \eta\beta\sigma_{\max}^2\sum_{l'=0}^{l-1}\|w_i^{k,l'} - w_0^k\| \le \frac{\eta^2 L\sigma_{\max}^3\beta M l}{\gamma(1 - l\eta\beta\sigma_{\max}^2)}\|\nabla f(w_0^k)\|. \qquad (59)$$

By smoothness, we also have

$$\|\nabla f(w_i^{k,l}) - \nabla f(w_0^k)\| \le \sigma_{\max}^2\beta\|w_i^{k,l} - w_0^k\| \le \frac{\eta L\sigma_{\max}^3\beta M}{\gamma(1 - l\eta\beta\sigma_{\max}^2)}\|\nabla f(w_0^k)\|. \qquad (60)$$

□

C.2 Proof of Claim 2

In this section, we prove our implicit bias result. Recall that $\hat{w}$ is the global max-margin solution defined in (4). We denote the set of support vectors in S as V. Thus the max-margin solution is $\hat{w}=\sum_{s\in S}\alpha_s x_s$, where $\alpha_s>0,\forall s\in V;\alpha_s=0,\forall s\notin V$. We further define a vector $\tilde{w}$, which satisfies

$$\alpha_s=\eta\exp(-x_s^T\tilde{w})\quad \forall s\in V. \tag{61}$$

From Lemma 12 in Soudry et al. (2018), this solution exists for almost every dataset. We also denote the minimum margin to a non-support vector as

$$\theta=\min_{s\notin V}x_s^T\hat{w}>1. \tag{62}$$

We will use the following Lemma:

Lemma 8. *There exists $m_i(k,l)$ such that*

$$L\sum_{u=1}^{k-1}\frac{1}{u}\frac{1}{M}\sum_{s\in V}\alpha_s x_s+\frac{l}{k}\sum_{s\in V_i}\alpha_s x_s=\frac{L}{M}\log(k)\hat{w}+\frac{L}{M}\zeta\hat{w}+m_i(k,l),\quad \forall l\in[1,L] \tag{63}$$

$$m_i(k+1,0)\triangleq\frac{1}{M}\sum_{i=1}^{M}m_i(k,L),\quad \forall i\in[1,M] \tag{64}$$

where $\|m_i(k,l)\|=o(k^{-1})$ and $\|m_i(k,l+1)-m_i(k,l)\|=O(k^{-1})$. ζ is Euler-Mascheroni constant, which is used to calculate $\sum_{u=1}^{k}\frac{1}{u}=\log k+\zeta+O(k^{-1})$.

Now we define $r_i^{k,l},\rho_i^{k,l}$ as

$$\begin{aligned}w_i^{k,l}&=\log(Lk)\hat{w}+\rho_i^{k,l}\\&=\log(Lk)\hat{w}+\tilde{w}+\frac{M}{L}m_i(k,l)+r_i^{k,l},\quad \forall l\in[1,L].\end{aligned} \tag{65}$$

Also, define $r_0^{k+1}=\frac{1}{M}\sum_{i=1}^{M}r_i^{k,L}$ and $\rho^{k+1}=\frac{1}{M}\sum_{i=1}^{M}\rho_i^{k,L}$. Thus

$$w_0^k=\frac{1}{M}\sum_{i=1}^{M}w_i^{k,L}=\log(Lk)\hat{w}+\rho^k=\log(Lk)\hat{w}+\tilde{w}+\frac{M}{L}\frac{1}{M}\sum_{i=1}^{M}m_i(k,l)+r_0^k \tag{66}$$

We also define

$$\rho_i^{k,0}=\rho^k,\quad r_i^{k,0}=r_0^k \tag{67}$$

Then for $l=0$, we have

$$w_i^{k+1,0}=w_0^{k+1}=\log(Lk)\hat{w}+\tilde{w}+m_i(k+1,0)+r_i^{k+1,0}. \tag{68}$$

We aim to bound $\|\rho^k\|$, and we can see that it is enough to prove $\|r_0^k\|$ is bounded to achieve this goal.

We first write for a constant $k_1>0$ (defined later) and all $K\geq k_1$

$$\begin{aligned}\|r_0^K\|^2-\|r_0^{k_1}\|^2&=\sum_{u=k_1}^{K}\|r_0^{u+1}\|^2-\|r_0^u\|^2\\&\leq\sum_{u=k_1}^{K}\frac{1}{M}\sum_{i=1}^{M}\left(\|r_i^{u,L}\|^2-\|r_i^{u,0}\|^2\right)\\&=\frac{1}{M}\sum_{u=k_1}^{K}\sum_{l=0}^{L-1}\sum_{i=1}^{M}\|r_i^{u,l+1}\|^2-\|r_i^{u,l}\|^2\\&=\frac{1}{M}\sum_{u=k_1}^{K}\sum_{l=0}^{L-1}\sum_{i=1}^{M}2\left\langle r_i^{u,l+1}-r_i^{u,l},r_i^{u,l}\right\rangle+\|r_i^{u,l+1}-r_i^{u,l}\|^2\end{aligned} \tag{69}$$

We will handle the inner product and squared norm items respectively. Here we need a lemma to characterize the behavior of inner product $\langle r_i^{u,l+1}-r_i^{u,l},r_i^{u,l}\rangle$, which can be adapted from a lemma in Nacson et al. (2019) and its proof is omitted here:

Lemma 9 (Adapted from Lemma 6 in Nacson et al. (2019)). *Under Assumptions 1, 2, 3,* $\exists \tilde{t}, C_1, C_2 > 0$ *such that* $\forall k > \tilde{k}$,

$$\langle r_i^{k,l+1} - r_i^{k,l}, r_i^{k,l} \rangle \leq C_1 (Lk)^{-\theta} + \frac{C_2 M}{L} k^{-1-0.5\tilde{\mu}}, \forall l \in [0, L-1] \tag{70}$$

, *where* $\tilde{\mu} = \min\{\mu_+, \mu_-, 0.25\}$.

Let $a_i^{k,l} = \frac{M}{L}(m_i(k,l+1) - m_i(k,l))$ and we know $\|m_i(k,l+1) - m_i(k,l)\| = O(k^{-1})$ from Lemma 8. Then we can handle the squared norm item:

$$\begin{aligned}
&\frac{1}{M} \sum_{u=k_1}^{K} \sum_{l=0}^{L-1} \sum_{i=1}^{M} \|r_i^{u,l+1} - r_i^{u,l}\|^2 \\
=&\frac{1}{M} \sum_{u=k_1}^{K} \sum_{l=0}^{L-1} \sum_{i=1}^{M} \|w_i^{k,l+1} - w_i^{k,l} - a_i^{k,l}\|^2 \\
=&\frac{1}{M} \sum_{u=k_1}^{K} \sum_{l=0}^{L-1} \sum_{i=1}^{M} \|w_i^{u,l+1} - w_i^{u,l}\|^2 + \frac{1}{M} \sum_{u=k_1}^{K} \sum_{l=0}^{L-1} \sum_{i=1}^{M} 2\left\langle w_i^{u,l} - w_i^{u,l+1}, a_i^{t,k} \right\rangle + \frac{1}{M} \sum_{u=k_1}^{K} \sum_{l=0}^{L-1} \sum_{i=1}^{M} \|a_i^{u,l}\|^2 \\
\leq&\frac{1}{M} \sum_{u=k_1}^{T} \sum_{l=0}^{L-1} \sum_{i=1}^{M} \|w_i^{u,l+1} - w_i^{u,l}\|^2 + \frac{2}{M} \sqrt{\sum_{u=k_1}^{K} \sum_{l=0}^{L-1} \sum_{i=1}^{M} \|w_i^{u,l+1} - w_i^{u,l}\|^2 \sum_{u=k_1}^{K} \sum_{l=0}^{L-1} \sum_{i=1}^{M} \|a_i^{u,l}\|^2} \\
&+ \frac{1}{M} \sum_{u=k_1}^{K} \sum_{l=0}^{L-1} \sum_{i=1}^{M} \|a_i^{u,l}\|^2
\end{aligned} \tag{71}$$

Since $\|a_i^{k,l}\| = O(\frac{M}{Lk})$, we can find a k_1 such that $\forall k \geq k_1, \forall l \in [0, L-1], \forall i \in [1, M]$ we have $\|a_i^{k,l}\| \leq \frac{M}{Lk}$. Also, we know $\frac{1}{M} \sum_{k=t_1}^{K} \sum_{l=0}^{L-1} \sum_{i=1}^{M} \|w_i^{k,l+1} - w_i^{k,l}\|^2 < \infty$ from the proof of Claim 1 (53). Then we can obtain

$$\begin{aligned}
&\frac{1}{M} \sum_{k=k_1}^{K} \sum_{l=0}^{L-1} \sum_{i=1}^{M} \|r_i^{k,l+1} - r_i^{k,l}\|^2 \\
\leq&\frac{1}{M} \sum_{k=k_1}^{K} \sum_{l=0}^{L-1} \sum_{i=1}^{M} \|w_i^{k,l+1} - w_i^{k,l}\|^2 + 2 \sqrt{\frac{1}{M} \sum_{k=k_1}^{K} \sum_{l=0}^{L-1} \sum_{i=1}^{M} \|w_i^{k,l+1} - w_i^{k,l}\|^2 \frac{1}{M} \sum_{k=k_1}^{K} \sum_{l=0}^{L-1} \sum_{i=1}^{M} \frac{M^2}{L^2} k^{-2}} \\
&+ \frac{1}{M} \sum_{k=k_1}^{K} \sum_{l=0}^{L-1} \sum_{i=1}^{M} \frac{M^2}{L^2} k^{-2} \\
<&\infty.
\end{aligned} \tag{72}$$

With Lemma 9 and the fact that $\forall c > 1, \sum_{u=1}^{\infty} u^{-c} < \infty$, we can finally get

$$\|r_0^k\|^2 - \|r_0^{k_1}\|^2 \leq \frac{1}{M} \sum_{k=k_1}^{K} \sum_{l=0}^{L-1} \sum_{i=1}^{M} \left(2\left\langle r_i^{u,l+1} - r_i^{u,l}, r_i^{u,l} \right\rangle + \|r_i^{u,l+1} - r_i^{u,l}\|^2 \right) < \infty. \tag{73}$$

The $\|r_0^k\|^2$ is bounded, then $\|\rho^k\|$ is also bounded. We can know w_0^k converges to $\hat{w}$ in direction: $w_0^{k+1} = \log(Lk)\hat{w} + \rho^k$.

Then we can analyze the dependence of $\|\rho^k\|$ on L. From (53) and the condition on learning rate $\eta = O(L^{-1})$ we can know

$$\frac{1}{M} \sum_{k=k_1}^{K} \sum_{l=0}^{L-1} \sum_{i=1}^{M} \|w_i^{k,l+1} - w_i^{k,l}\|^2 \leq O(L^{-1}) \sum_{k=0}^{\infty} \|\nabla f(w_0^k)\|^2. \tag{74}$$

Then we can write 72 as

$$\begin{aligned}
&\frac{1}{M}\sum_{k=k_1}^{K}\sum_{l=0}^{L-1}\sum_{i=1}^{M}\|r_i^{k,l+1}-r_i^{k,l}\|^2\\
\leq& O(L^{-1})\sum_{k=k_1}^{\infty}\|\nabla f(w_0^k)\|^2+2\sqrt{O(L^{-1})\sum_{k=k_1}^{\infty}\|\nabla f(w_0^k)\|^2\cdot\frac{M^2}{L}\sum_{k=k_1}^{K}k^{-2}+\frac{M^2}{L}\sum_{k=k_1}^{K}k^{-2}}\\
\leq& O(L^{-1})\left(\sum_{k=k_1}^{\infty}\|\nabla f(w_0^k)\|^2+\sqrt{\sum_{k=k_1}^{\infty}\|\nabla f(w_0^k)\|^2\sum_{k=k_1}^{K}k^{-2}}+\sum_{k=k_1}^{K}k^{-2}\right)
\end{aligned}\tag{75}$$

From Lemma 9, since $\theta>1$ we can know

$$\langle r_i^{k,l+1}-r_i^{k,l},r_i^{k,l}\rangle\leq C_1(Lk)^{-\theta}+\frac{C_2M}{L}k^{-1-0.5\tilde{\mu}}=O(L^{-1})(k^{-\theta}+k^{-1-0.5\tilde{\mu}})\tag{76}$$

Then we can obtain

$$\begin{aligned}
&\|r_0^k\|^2-\|r_0^{k_1}\|^2\\
\leq&\frac{1}{M}\sum_{k=k_1}^{K}\sum_{l=0}^{L-1}\sum_{i=1}^{M}\left(2\left\langle r_i^{u,l+1}-r_i^{u,l},r_i^{u,l}\right\rangle+\|r_i^{u,l+1}-r_i^{u,l}\|^2\right)\\
\leq&O(1)\sum_{k=k_1}^{K}(k^{-\theta}+k^{-1-0.5\tilde{\mu}})+O(L^{-1})\left(\sum_{k=k_1}^{\infty}\|\nabla f(w_0^k)\|^2+\sqrt{\sum_{k=k_1}^{\infty}\|\nabla f(w_0^k)\|^2\sum_{k=k_1}^{K}k^{-2}}+\sum_{k=k_1}^{K}k^{-2}\right)\\
<&\infty
\end{aligned}\tag{77}$$

and the dominating term on L is $O(1)$. By definition $\rho_i^{k,l}=\log(Lk)\hat{w}+\tilde{w}+\frac{M}{L}m_i(k,l)+r_i^{k,l}$, we can get $\|\rho_i^{k,l}\|$ is bounded with $k\to\infty$ and $O(1)$ on L.

Now we can get the convergence rate of the direction.

$$\begin{aligned}
&\frac{w_0^{k+1}}{\|w_0^{k+1}\|}\\
=&\frac{\log(Lk)\hat{w}+\rho^k}{\sqrt{\rho^{kT}\rho^k+\hat{w}^T\hat{w}\log^2(Lk)+2\rho^{kT}\hat{w}\log(Lk)}}\\
=&\frac{\rho^k/\log(Lk)+\hat{w}}{\|\hat{w}\|\sqrt{1+\frac{2\rho^{kT}\hat{w}}{\|\hat{w}\|^2\log(Lk)}+\frac{\|\rho^k\|^2}{\|\hat{w}\|^2\log^2(Lk)}}}\\
=&\frac{1}{\|\hat{w}\|}\left(\frac{\rho^k}{\log(Lk)}+\hat{w}\right)\left[1-\frac{\rho^{kT}\hat{w}}{\|\hat{w}\|^2\log(Lk)}+\left(\frac{3}{2}\left(\frac{\rho^{kT}\hat{w}}{\|\hat{w}\|^2}\right)^2-\frac{\|\rho^k\|^2}{2\|\hat{w}\|^2}\right)\frac{1}{\log^2(Lk)}+O\left(\frac{1}{\log^3(Lk)}\right)\right]\\
=&\frac{\hat{w}}{\|\hat{w}\|}+\left(\frac{\rho^k}{\|\hat{w}\|}-\frac{\hat{w}}{\|\hat{w}\|}\frac{\rho^{kT}\hat{w}}{\|\hat{w}\|^2}\right)\frac{1}{\log(Lk)}+O(\frac{1}{\log^2(Lk)})\\
=&\frac{\hat{w}}{\|\hat{w}\|}+\left(I-\frac{\hat{w}\hat{w}^T}{\|\hat{w}\|^2}\right)\frac{\rho^k}{\|\hat{w}\|}\frac{1}{\log(Lk)}+O(\frac{1}{\log^2(Lk)}),
\end{aligned}\tag{78}$$

where the third equality is from $\frac{1}{\sqrt{1+x}}=1-\frac{1}{2}x+\frac{3}{4}x^2+O(x^3)$. Thus we can get

$$\left\|\frac{w_0^k}{\|w_0^k\|}-\frac{\hat{w}}{\|\hat{w}\|}\right\|=O\left(\frac{1}{\log(Lk)}\right).\tag{79}$$

C.2.1 Proof of Lemma 8

Proof. We first write

$$\begin{aligned}
&L\sum_{u=1}^{k-1}\frac{1}{u}\frac{1}{M}\sum_{s\in V}\alpha_s x_s+\frac{l}{k}\sum_{s\in V_i}\alpha_s x_s\\
=&\frac{L}{M}\hat{w}\sum_{u=1}^{k-1}\frac{1}{u}+\frac{l}{k}\sum_{s\in V_i}\alpha_s x_s\\
=&\frac{L}{M}\hat{w}(\log(k)+\zeta+O(k^{-1}))+\frac{l}{k}\sum_{s\in V_i}\alpha_s x_s\\
=&\frac{L}{M}\log(k)\hat{w}+\frac{L\zeta}{M}\hat{w}+O(k^{-1})\hat{w}+\frac{l}{k}\sum_{s\in V_i}\alpha_s x_s,
\end{aligned}\tag{80}$$

where the first equality is definition of $\hat{w}$, the second equality is from the fact

$$\sum_{u=1}^{k}\frac{1}{u}=\log k+\zeta+O(k^{-1})\tag{81}$$

$$\text{and}\quad \log k-\log(k-1)=O(k^{-1}).\tag{82}$$

Then we define

$$m_i(k,l)=L\sum_{u=1}^{k-1}\frac{1}{u}\frac{1}{M}\sum_{s\in V}\alpha_s x_s+\frac{l}{k}\sum_{s\in V_i}\alpha_s x_s-\frac{L}{M}\log(k)\hat{w}-\frac{L\zeta}{M}\hat{w},\quad \forall l\in[1,L]\tag{83}$$

and

$$m_i(k+1,0)=\frac{1}{M}\sum_{i=1}^{M}m_i(k,L)=L\sum_{u=1}^{k}\frac{1}{u}\frac{1}{M}\sum_{s\in V}\alpha_s x_s-\frac{L}{M}\log(k)\hat{w}-\frac{L\zeta}{M}\hat{w},\quad \forall i\in[1,M].\tag{84}$$

We can obviously see $\|m_i(k,l)\|=O(k^{-1})$. For the difference, we can get

$$\|m_i(k,l+1)-m_i(k,l)\|=\|\frac{1}{k}\sum_{s\in V_i}\alpha_s x_s\|=O(k^{-1}),\quad \forall l\in[1,L-1]\tag{85}$$

$$\|m_i(k,1)-m_i(k,0)\|=\|\frac{1}{k}\sum_{s\in V_i}\alpha_s x_s-\frac{L}{M}(\log(k+1)-\log k)\|=O(k^{-1}).\tag{86}$$

□

C.3 Proof of Claim 3

In the proof of Claim 1, we already know $f(w_0^k)$ would continue to decrease to zero when $k\to\infty$. Now we establish the convergence rate of $f(w_0^k)$. Recall V is the set of support vectors and θ is the minimum margin for non-support vectors. From Assumptions 2 and 3, we can get

$$\begin{aligned}
f(w_0^k)\le&\frac{1}{M}\sum_{s\in S}\big(1+\exp(-\mu_+x_s^Tw_0^k)\big)\exp(-x_s^Tw_0^k)\\
=&\frac{1}{M}\sum_{s\in S}\big(1+\exp(-\mu_+x_s^T(\hat{w}\log(Lk)+\rho^k))\big)\exp(-x_s^T(\hat{w}\log(Lk)+\rho^k))\\
=&\frac{1}{M}\sum_{s\in S}\Big(1+(Lk)^{-\mu_+x_s^T\hat{w}}\exp(-\mu_+x_s^T\rho^k)\Big)\exp(-x_s^T\rho^k)(Lk)^{-x_s^T\hat{w}}\\
=&\frac{1}{M}\sum_{s\in S}\Big[\exp(-x_s^T\rho^k)(Lk)^{-x_s^T\hat{w}}+(Lk)^{-\mu_+x_s^T\hat{w}}\exp(-\mu_+x_s^T\rho^k)\exp(-x_s^T\rho^k)(Lk)^{-x_s^T\hat{w}}\Big]
\end{aligned}\tag{87}$$

We can divide the dataset S into set V with support vectors and the complementary set. For samples in the set V, we have $x_s^T \hat{w} = 1$ and we can write

$$\sum_{s\in V} \exp(-x_s^T \rho^k)(Lk)^{-x_s^T \hat{w}} + (Lk)^{-\mu_+ x_s^T \hat{w}} \exp(-\mu_+ x_s^T \rho^k) \exp(-x_s^T \rho^k)(Lk)^{-x_s^T \hat{w}}$$
$$= \sum_{s\in V} \frac{1}{Lk} \exp(-x_s^T \rho^k) + \frac{1}{(Lk)^{1+\mu_+}} \exp(-(1+\mu_+) x_s^T \rho^k) \tag{88}$$

For samples not in the set V, we have $x_s^T \hat{w} \geq \theta$ since θ is the minimum margin for non-support vectors. Then we can write

$$\sum_{s\notin V} \exp(-x_s^T \rho^k)(Lk)^{-x_s^T \hat{w}} + (Lk)^{-\mu_+ x_s^T \hat{w}} \exp(-\mu_+ x_s^T \rho^k) \exp(-x_s^T \rho^k)(Lk)^{-x_s^T \hat{w}}$$
$$\leq \sum_{s\notin V} \frac{1}{(Lk)^{\theta}} \exp(-x_s^T \rho^k) + \frac{1}{(Lk)^{(1+\mu_+)\theta}} \exp(-(1+\mu_+) x_s^T \rho^k) \tag{89}$$

Combining the two terms, we can have

$$f(w_0^k) \leq \frac{1}{M} \sum_{s\in S} \left[\exp(-x_s^T \rho^k)(Lk)^{-x_s^T \hat{w}} + (Lk)^{-\mu_+ x_s^T \hat{w}} \exp(-\mu_+ x_s^T \rho^k) \exp(-x_s^T \rho^k)(Lk)^{-x_s^T \hat{w}} \right]$$
$$= \left[\frac{1}{MLk} \sum_{s\in V} \exp(-x_s^T \rho^k) \right] + O((Lk)^{-\max(\theta, 1+\mu_+)}). \tag{90}$$

Thus the training loss $f(w_0^k) = O\left(\frac{1}{Lk}\right)$.

D PROOFS OF IMPLICIT BIAS WITH LEARNING RATE INDEPENDENT OF L IN SECTION 4

In this section we also redefine the samples $y_{ij}x_{ij}$ to x_{ij} to subsume the labels. With abuse of notation, we use S_i to denote the set of support vectors in i-th compute node and S is the set of support vectors in global dataset. The number of samples N is identical for all the compute nodes, and the local dataset is $\{x_{ij},y_{ij}\}_{j=1}^{N}$. Since the loss function is fixed as exponential loss in this section, the β in this section refers coefficient of support vectors, not smoothness parameter.

D.1 PROOFS OF LEMMA 1

We assume $\|w_0^k-\ln(\frac{1}{\lambda})\bar{w}_0^k\|=O(k\ln\ln\frac{1}{\lambda})$. In this case, since $\ln\frac{1}{\lambda}$ grows faster, when $\lambda\to 0$, we can have $\lim_{\lambda\to 0}\frac{w_0^k}{\|w_0^k\|}=\frac{\bar{w}_0^k}{\|\bar{w}_0^k\|}$ for any k at order $o\Big(\frac{\ln(1/\lambda)}{\ln\ln(1/\lambda)}\Big)$. We will prove it by induction. We define global and local residuals as $r^k=w_0^k-\ln(\frac{1}{\lambda})\bar{w}_0^k$ and $r_i^k=w_i^k-\ln(\frac{1}{\lambda})\bar{w}_i^k$.

When $k=0$, since $w_0^0=\bar{w}_0^0=0$, $r_i^0=0$ and the assumption trivially holds.

When $k\geq 1$, we have

$$\begin{aligned}\|r^k\|&=\left\|w_0^k-\ln(\frac{1}{\lambda})\bar{w}_0^k\right\|=\frac{1}{M}\left\|\sum_{i=1}^{M}w_i^k-\ln(\frac{1}{\lambda})\bar{w}_i^k\right\|\\&\leq\frac{1}{M}\sum_{i=1}^{M}\left\|w_i^k-\ln(\frac{1}{\lambda})\bar{w}_i^k\right\|=\frac{1}{M}\sum_{i=1}^{M}\|r_i^k\|.\end{aligned}\tag{91}$$

where the inequality is triangle inequality. We then focus on the local residual r_i^k. We choose an $O(1)$ vector $\tilde{w}_i^k$ and a sign $s_i^k\in\{-1,+1\}$ to show

$$\begin{aligned}\|r_i^k\|&=\left\|w_i^k-\left[\left(\ln(\frac{1}{\lambda})+s_i^k\ln\ln(\frac{1}{\lambda})\right)\bar{w}_i^k+\tilde{w}_i^k\right]+s_i^k\ln\ln(\frac{1}{\lambda})\bar{w}_i^k+\tilde{w}_i^k\right\|\\&\leq\left\|w_i^k-\left[\left(\ln(\frac{1}{\lambda})+s_i^k\ln\ln(\frac{1}{\lambda})\right)\bar{w}_i^k+\tilde{w}_i^k\right]\right\|+\ln\ln(\frac{1}{\lambda})\|\bar{w}_i^k\|+\|\tilde{w}_i^k\|\end{aligned}\tag{92}$$

Recall the w_i^k is the solution of optimization problem

$$\operatorname*{argmin}_{w_i}f_i(w_i)=\sum_{j=1}^{N}\exp\big(-x_{ij}^Tw_i\big)+\frac{\lambda}{2}\|w_i-w_0^{k-1}\|^2,\tag{93}$$

and the loss function $f_i(w_i)$ is a λ-strongly convex function. Thus we have

$$\|w_i^k-w\|\leq\frac{1}{\lambda}\|\nabla f_i(w)\|,\quad\text{for any } w.\tag{94}$$

Then back to 92, we have

$$\|r_i^k\|\leq\underbrace{\frac{1}{\lambda}\left\|\nabla f_i\left[\left(\ln(\frac{1}{\lambda})+s_i^k\ln\ln(\frac{1}{\lambda})\right)\bar{w}_i^k+\tilde{w}_i^k\right]\right\|}_{\|A_i\|}+\ln\ln(\frac{1}{\lambda})\|\bar{w}_i^k\|+\|\tilde{w}_i^k\|.\tag{95}$$

Next we need to show the first term A_i is at $O((k-1)\ln\ln(\frac{1}{\lambda}))$, and also since $\|\bar{w}_i^k\|$ and $\|\tilde{w}_i^k\|$ are $O(1)$ vectors, then $\|r_i^k\|$ is at order $O(k\ln\ln(\frac{1}{\lambda}))$. After averaging, $\|r^k\|$ is also at order $O(k\ln\ln(\frac{1}{\lambda}))$. This confirms the assumption made for induction.

Now we focus on the term A_i. The gradient of function $f_i(w)$ is

$$\nabla f_i(w_i)=\sum_j-x_{ij}\exp(-x_{ij}^Tw_i)+\lambda(w_i-w_0^{k-1}).\tag{96}$$

The term A_i is

$$\begin{aligned}A_i=&\frac{1}{\lambda}\nabla f_i\left[\left(\ln(\frac{1}{\lambda})+s_i^k\ln\ln(\frac{1}{\lambda})\right)\bar{w}_i^k+\tilde{w}_i^k\right]\\=&-\frac{1}{\lambda}\sum_j x_{ij}\exp\left(x_{ij}^T\ln\left(\lambda\ln^{-s_i^k}(\frac{1}{\lambda})\right)\bar{w}_i^k\right)\exp(-x_{ij}^T\tilde{w}_i^k)+\left(\ln(\frac{1}{\lambda})+s_i^k\ln\ln(\frac{1}{\lambda})\right)\bar{w}_i^k+\tilde{w}_i^k-w_0^{k-1}\\=&-\frac{1}{\lambda}\sum_j x_{ij}\left(\lambda\ln^{-s_i^k}(\frac{1}{\lambda})\right)^{x_{ij}^T\bar{w}_i^k}\exp(-x_{ij}^T\tilde{w}_i^k)+\left(\ln(\frac{1}{\lambda})+s_i^k\ln\ln(\frac{1}{\lambda})\right)\bar{w}_i^k+\tilde{w}_i^k-w_0^{k-1}.\end{aligned}\tag{97}$$

Then we define the set of support vectors as $S_i^k=\{x_{ij}|x_{ij}^T\bar{w}_i^k=1\}$. Recall that we assume $r^{k-1}=w_0^{k-1}-\ln(\frac{1}{\lambda})\bar{w}_0^{k-1}$ is at order $O((k-1)\ln\ln(\frac{1}{\lambda}))$. We can obtain

$$\begin{aligned}A_i=&-\frac{1}{\lambda}\left(\lambda\ln^{-s_i^k}(\frac{1}{\lambda})\right)^1\sum_{x_{ij}\in S_i^k}x_{ij}\exp(-x_{ij}^T\tilde{w}_i^k)-\frac{1}{\lambda}\sum_{x_{ij}\notin S_i^k}x_{ij}\left(\lambda\ln^{-s_i^k}(\frac{1}{\lambda})\right)^{x_{ij}^T\bar{w}_i^k}\exp(-x_{ij}^T\tilde{w}_i^k)\\&+\ln(\frac{1}{\lambda})(\bar{w}_i^k-\bar{w}_0^{k-1})-r^{k-1}+s_i^k\ln\ln(\frac{1}{\lambda})\bar{w}_i^k+\tilde{w}_i^k\\=&-\ln^{-s_i^k}(\frac{1}{\lambda})\sum_{x_{ij}\in S_i^k}x_{ij}\exp(-x_{ij}^T\tilde{w}_i^k)-\sum_{x_{ij}\notin S_i^k}x_{ij}\lambda^{x_{ij}^T\bar{w}_i^k-1}\left(\ln(\frac{1}{\lambda})\right)^{-s_i^kx_{ij}^T\bar{w}_i^k}\exp(-x_{ij}^T\tilde{w}_i^k)\\&+\ln(\frac{1}{\lambda})(\bar{w}_i^k-\bar{w}_0^{k-1})-r^{k-1}+s_i^k\ln\ln(\frac{1}{\lambda})\bar{w}_i^k+\tilde{w}_i^k.\end{aligned}\tag{98}$$

By the triangle inequality, we have

$$\begin{aligned}\|A_i\|\leq&\underbrace{\left\|\ln(\frac{1}{\lambda})(\bar{w}_i^k-\bar{w}_0^{k-1})-\ln^{-s_i^k}(\frac{1}{\lambda})\sum_{x_{ij}\in S_i^k}x_{ij}\exp(-x_{ij}^T\tilde{w}_i^k)\right\|}_{B_1}\\&+\underbrace{\left\|\sum_{x_{ij}\notin S_i^k}x_{ij}\lambda^{x_{ij}^T\bar{w}_i^k-1}\left(\ln(\frac{1}{\lambda})\right)^{-s_i^kx_{ij}^T\bar{w}_i^k}\exp(-x_{ij}^T\tilde{w}_i^k)\right\|}_{B_2}\\&+\underbrace{\|r^{k-1}\|}_{O((k-1)\ln\ln(\frac{1}{\lambda}))}+\ln\ln(\frac{1}{\lambda})\underbrace{\|\bar{w}_i^k\|}_{O(1)}+\underbrace{\|\tilde{w}_i^k\|}_{O(1)}.\end{aligned}\tag{99}$$

We just need to show B_1 and B_2 approach to 0 then $\|A_i\|$ can approach to $O(k\ln\ln(\frac{1}{\lambda}))$.

We divide it into two cases.

1. When $\bar{w}_i^k=P(\bar{w}_0^{k-1})\neq\bar{w}_0^{k-1}$, meaning $\bar{w}_0^{k-1}$ is not in the convex set C_i. In this case we choose $s_i^k=-1$ then

$$\begin{aligned}B_1=&\left\|\ln(\frac{1}{\lambda})(\bar{w}_i^k-\bar{w}_0^{k-1})-\ln(\frac{1}{\lambda})\sum_{x_{ij}\in S_i^k}x_{ij}\exp(-x_{ij}^T\tilde{w}_i^k)\right\|\\=&\ln(\frac{1}{\lambda})\left\|(\bar{w}_i^k-\bar{w}_0^{k-1})-\sum_{x_{ij}\in S_i^k}x_{ij}\exp(-x_{ij}^T\tilde{w}_i^k)\right\|.\end{aligned}\tag{100}$$

We now want to choose $\tilde{w}_i^k$ to make B_1 as 0. Since $\bar{w}_i^k$ is the solution of SVM problem (9), by the KKT condition of SVM problem, it can be written as

$$\bar{w}_i^k=\bar{w}_0^{k-1}+\sum_{x_{ij}\in S_i^k}\beta_{ij}x_{ij}\tag{101}$$

where β_{ij} is the dual varible corresponding to x_{ij} in the set of support vectors. Thus we want to choose $\tilde{w}_i^k$ as

$$\sum_{x_{ij}\in S_i^k} \exp(-x_{ij}^T \tilde{w}_i^k) x_{ij} = \sum_{x_{ij}\in S_i^k} \beta_{ij} x_{ij}. \tag{102}$$

We can prove such a $\tilde{w}_i^k$ almost surely exists in Lemma 10.

For the term B_2, since $\lim_{\lambda\to 0} \lambda^{c-1} \ln^c(\frac{1}{\lambda}) \to 0$ for any constant $c>1$, and $x_{ij}^T \bar{w}_i^k - 1 > 0$ for any x_{ij} being not a support vector, then we can see

$$B_2 = \left\| \sum_{x_{ij}\notin S_i^k} x_{ij} \lambda^{x_{ij}^T \bar{w}_i^k - 1} \left(\ln(\frac{1}{\lambda}) \right)^{x_{ij}^T \bar{w}_i^k} \exp(-x_{ij}^T \tilde{w}_i^k) \right\| \xrightarrow{\lambda\to 0} 0. \tag{103}$$

Here we choose $\tilde{w}_i^k$ and s_i^k to make $B_1 = 0$ and $B_2 \to 0$.

2. When $\bar{w}_i^k = P(\bar{w}_0^{k-1}) = \bar{w}_0^{k-1}$, meaning $\bar{w}_0^{k-1}$ is already in the convex set C_i. Then $\bar{w}_i^k - \bar{w}_0^{k-1} = 0$. In this case we choose $\tilde{w}_i^k = 0$ and $s_i^k = +1$. We can have

$$B_1 = \ln^{-1}(\frac{1}{\lambda}) \left\| \sum_{x_{ij}\in S_i^k} x_{ij} \right\| \xrightarrow{\lambda\to 0}, \tag{104}$$

since $\ln^{-1}(\frac{1}{\lambda}) \xrightarrow{\lambda\to 0} 0$ and $\left\| \sum_{x_{ij}\in S_i^k} x_{ij} \right\|$ is $O(1)$.

And since $x_{ij}^T \bar{w}_i^k - 1 > 0$ for any x_{ij} being not a support vector, we have

$$B_2 = \left\| \sum_{x_{ij}\notin S_i^k} x_{ij} \lambda^{x_{ij}^T \bar{w}_i^k - 1} \left(\ln(\frac{1}{\lambda}) \right)^{-x_{ij}^T \bar{w}_i^k} \right\| \xrightarrow{\lambda\to 0} 0, \tag{105}$$

where $\lambda^{x_{ij}^T \bar{w}_i^k - 1} \xrightarrow{\lambda\to 0} 0$ and $\left(\ln(\frac{1}{\lambda})\right)^{-x_{ij}^T \bar{w}_i^k} \xrightarrow{\lambda\to 0} 0$. Thus we choose $\tilde{w}_i^k$ and s_i^k to make $B_1 \to 0$ and $B_2 \to 0$.

Plugging 99 back into 95, we can obtain

$$\begin{aligned} \|r_i^k\| &\le \|A_i^k\| + \ln\ln(\frac{1}{\lambda}) \|\bar{w}_i^k\| + \|\tilde{w}_i^k\| \\ &\le \underbrace{B_1 + B_2}_{\to 0} + 2\ln\ln(\frac{1}{\lambda}) \|\bar{w}_i^k\| + 2\|\tilde{w}_i^k\| + \|r^{k-1}\| \\ &\le 2\ln\ln(\frac{1}{\lambda}) \|\bar{w}_i^k\| + 2\|\tilde{w}_i^k\| + \|r^{k-1}\|. \end{aligned} \tag{106}$$

By the assumption $\|r^{k-1}\| = O((k-1)\ln\ln(\frac{1}{\lambda}))$ and $\|\bar{w}_i^k\| = O(1)$, $\|\tilde{w}_i^k\| = O(1)$, we have $\|r_i^k\| = O(k\ln\ln(\frac{1}{\lambda}))$.

From 91, we finally obtain

$$\|r^k\| \le \frac{1}{M} \|r_i^k\| = O(k\ln\ln(\frac{1}{\lambda})), \tag{107}$$

which confirms our assumption. Then we have $\lim_{\lambda\to 0} \frac{w_0^k}{\|w_0^k\|} = \frac{\bar{w}_0^k}{\|\bar{w}_0^k\|}$ for any k at order $o\left(\frac{\ln(1/\lambda)}{\ln\ln(1/\lambda)}\right)$.

D.2 Proofs of Auxiliary Lemmas

Lemma 10. *For the sequence $\{\bar{w}_0^k\}$ generated by sequential SVM problems 9 and aggregations, and for almost all datasets sampled from M continuous distributions, the unique dual solution $\beta_i^k \in \mathbb{R}^{|S_i|\times 1}$ satisfying the KKT conditions of SVM problem 9 has non-zero elements. Then there exists $\tilde{w}_i^k$ satisfying $X_{S_i} \tilde{w}_i^k = -\ln\beta_i^k$.*

For almost all datasets, a hyperplane can be determined by d points. Thus there are at most d support vectors and the set of support vectors is linearly independent.

Proof. By the KKT condition of SVM problem, we can write the solution as

$$\bar{w}_i^k = \bar{w}_0^{k-1} + \sum_{x_{ij}\in S_i} \beta_{ij}^k x_{ij} = \bar{w}_0^{k-1} + X_{S_i}^T \beta_i^k. \tag{108}$$

where $X_{S_i} \in \mathbb{R}^{|S_i|\times d}$ is the data matrix with all the support vectors, and $\beta_i^k \in \mathbb{R}^{|S_i|\times 1}$ is the dual variable vector. Thus we can obtain

$$\beta_i^k = \left(X_{S_i} X_{S_i}^T\right)^{-1} X_{S_i}(\bar{w}_i^k - \bar{w}_0^{k-1}) = \left(X_{S_i} X_{S_i}^T\right)^{-1} \mathbf{1}_{S_i} - \left(X_{S_i} X_{S_i}^T\right)^{-1} X_{S_i} \bar{w}_0^{k-1}, \tag{109}$$

where $X_{S_i} X_{S_i}^T$ is invertible since X_{S_i} has full row rank $|S_i|$, and the second equality is from $X_{S_i}\bar{w}_i^k = \mathbf{1}_{S_i}$ with $\mathbf{1}_{S_i} \in \mathbb{R}^{|S_i|\times 1}$ being all one vector. Plugging β_i^k back, we have

$$\bar{w}_i^k = \left[I - X_{S_i}^T \left(X_{S_i} X_{S_i}^T\right)^{-1} X_{S_i}\right] \bar{w}_0^{k-1} + X_{S_i}^T \left(X_{S_i} X_{S_i}^T\right)^{-1} \mathbf{1}_{S_i}. \tag{110}$$

After averaging, the global model is

$$\bar{w}_0^k = \left[I - \frac{1}{M}\sum_{i=1}^M X_{S_i}^T \left(X_{S_i} X_{S_i}^T\right)^{-1} X_{S_i}\right] \bar{w}_0^{k-1} + \frac{1}{M}\sum_{i=1}^M X_{S_i}^T \left(X_{S_i} X_{S_i}^T\right)^{-1} \mathbf{1}_{S_i}. \tag{111}$$

It implies $\bar{w}_0^k$ is a rational function in the components of $X_1, X_2, \dots, X_M$, and also β_i^k is also a rational function in the components of data matrices. So its entries can be expressed as $\beta_{ij}^k = p_{ij}^k(X_1, X_2, \dots, X_M)/q_{ij}^k(X_1, X_2, \dots, X_M)$ for some polynomials p_{ij}^k, q_{ij}^k. Note that $\beta_{ij}^k = 0$ only if $p_{ij}^k(X_1, X_2, \dots, X_M) = 0$, and the components of $X_1, X_2, \dots, X_M$ must constitute a root of polynomial p_{ij}^k. However, the root of any polynomial has measure zero, unless the polynomial is the zero polynomial, i.e., $p_{ij}^k(X_1, X_2, \dots, X_M) = 0$ for any $X_1, X_2, \dots, X_M$.

Next we need to show p_{ij}^k cannot be zero polynomials. To do this, we just need to construct a specific $X_1, X_2, \dots, X_M$ where the p_{ij}^k is not zero polynomial. Denote $e_i \in \mathbb{R}^d$ as the i-th standard unit vector, and $v_1, v_2, \dots, v_M$ be the number of support vectors at M compute nodes. We construct the datasets as

$$X_i = r_i [e_1, e_2, \dots, e_{v_i}]^T, \text{ for all } i. \tag{112}$$

where r_i are positive constants that will be chosen later. For these datasets, the set of support vector is dataset itself, i.e., $X_{S_i} = X_i$. We can calculate

$$X_i X_i^T = r_i^2 I_{v_i},\ X_i^T X_i = r_i^2 \begin{bmatrix} I_{v_i} & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{(d-v_i)\times(d-v_i)} \end{bmatrix},\ X_i^T \mathbf{1}_{S_i} = r_i \begin{bmatrix} \mathbf{1}_{v_i} \\ \mathbf{0}_{d-v_i} \end{bmatrix} \tag{113}$$

Thus we have

$$\bar{w}_i^k = \left(I_d - \begin{bmatrix} I_{v_i} & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{(d-v_i)\times(d-v_i)} \end{bmatrix}\right) \bar{w}_0^{k-1} + \frac{1}{r_i} \begin{bmatrix} \mathbf{1}_{v_i} \\ \mathbf{0}_{d-v_i} \end{bmatrix}. \tag{114}$$

After averaging, the global model in 111 becomes

$$\bar{w}_0^k = \underbrace{\begin{bmatrix} 0 & & & & & & & & \\ & \ddots & & & & & & & \\ & & 0 & & & & & & \\ & & & a_1 & & & & & \\ & & & & \ddots & & & & \\ & & & & & a_{v_{\max}-v_{\min}} & & & \\ & & & & & & 1 & & \\ & & & & & & & \ddots & \\ & & & & & & & & 1 \end{bmatrix}}_{A} \bar{w}_0^{k-1} + \underbrace{\begin{bmatrix} b_1 \\ \vdots \\ b_{v_{\max}} \\ \mathbf{0}_{d-v_{\max}} \end{bmatrix}}_{b}. \tag{115}$$

where $a_j \in \{\frac{1}{M}, \frac{2}{M}, ..., \frac{M-1}{M}\}$ is a constant in the range (0,1), $b_j = \frac{1}{M}\sum_{i \in B_j} \frac{1}{r_i}$ is a positive constant and $B_j \in [M]$ is a set consisting of some compute nodes. Note that A and b are fixed in the iterations and A is a diagonal matrix.

By recursively applying $\bar{w}_0^k = A\bar{w}_0^{k-1} + b$, due to $\bar{w}_0^0 = 0$, we can obtain

$$\bar{w}_0^k = \left(I + A + A^2 + \cdots + A^{k-1}\right)b. \tag{116}$$

Since A is diagonal, the summation is

$$\sum_{j=0}^{k-1} A^j = \begin{bmatrix} 1 & & & & & & & & \\ & \ddots & & & & & & & \\ & & 1 & & & & & & \\ & & & \sum_{j=0}^{k-1} a_1^j & & & & & \\ & & & & \ddots & & & & \\ & & & & & \sum_{j=0}^{k-1} a_{v_{\max}-v_{\min}}^j & & & \\ & & & & & & k & & \\ & & & & & & & \ddots & \\ & & & & & & & & k \end{bmatrix} \tag{117}$$

Recall that

$$\begin{aligned} \beta_i^k &= \left(X_i X_i^T\right)^{-1} \mathbf{1}_{v_i} - \left(X_i X_i^T\right)^{-1} X_i \bar{w}_0^{k-1} \\ &= \frac{1}{r_i^2}\mathbf{1}_{v_i} - \frac{1}{r_i^2}(\bar{w}_0^{k-1})_{v_i} = \frac{1}{r_i^2}\left(\mathbf{1}_{v_i} - (\bar{w}_0^{k-1})_{v_i}\right). \end{aligned} \tag{118}$$

where $(\bar{w}_0^{k-1})_{v_i}$ is the vector with first v_i elements of $\bar{w}_0^{k-1}$.

We need every element of β_i^k to be positive, so that we require every element of $(\bar{w}_0^{k-1})_{v_i}$ is less than 1. Then it holds for any i-th compute node, thus we require every element of $(\bar{w}_0^{k-1})_{v_{\max}}$ is less than 1. Since $\bar{w}_0^{k-1} = \left(\sum_{j=0}^{k-2} A^j\right)b$, the largest value of $(\bar{w}_0^{k-1})_{v_{\max}}$ satisfies

$$\begin{aligned} (\bar{w}_0^{k-1})_{\text{largest}} &\leq \sum_{j=0}^{k-2}\left(\frac{M-1}{M}\right)^j \times \frac{1}{M}\sum_{i=1}^{M}\frac{1}{r_i^2} \\ &= M\left(1 - \left(\frac{M-1}{M}\right)^{k-1}\right) * \frac{1}{M}\sum_{i=1}^{M}\frac{1}{r_i^2} \end{aligned} \tag{119}$$

because the maximum value of a_j is $\frac{M-1}{M}$ and the maximum value of b_j is $\frac{1}{M}\sum_{i=1}^{M}\frac{1}{r_i^2}$.

Thus we require

$$\sum_{i=1}^{M}\frac{1}{r_i} < \frac{1}{1 - \left(\frac{M-1}{M}\right)^{k-1}}. \tag{120}$$

Since $\left(\frac{M-1}{M}\right)^{k-1} \to 0$ when $k \to \infty$, we only require the LHS is less than the lower bound of RHS:

$$\sum_{i=1}^{M}\frac{1}{r_i} < 1. \tag{121}$$

Therefore we can choose $r_i = M+1$ to make it happen.

Then we can obtain $\beta_{ij}^k > 0$ holds for any support vector x_{ij} and any round k. And the $\tilde{w}_i^k$ simply satisfies $X_{S_i}\tilde{w}_i^k = -\ln\beta_i^k$. □

D.3 Lemma and Proofs in Section 4.4

Here we provide a lemma of Modified Local-GD similar to Lemma 1 of vanilla Local-GD.

Lemma 11. *For almost all datasets sampled from a continuous distribution satisfying Assumption 1, we train the global model w_0 from Modified Local-GD and $\bar{w}_0$ from Modified PPM. The parameter is chosen as $\alpha^k = 1 - \frac{1}{k+1}$. With initialization $w_0^0 = \bar{w}_0^0 = 0$, we have $w_0^k \to \ln\left(\frac{1}{\lambda}\right)\bar{w}_0^k$, and the residual $\|w_0^k - \ln\left(\frac{1}{\lambda}\right)\bar{w}_0^k\| = O(k\ln\ln\frac{1}{\lambda})$, as $\lambda \to 0$. It implies that at any round $k = o\left(\frac{\ln(1/\lambda)}{\ln\ln(1/\lambda)}\right)$, w_0^k converges in direction to $\bar{w}_0^k$:*

$$\lim_{\lambda\to 0} \frac{w_0^k}{\|w_0^k\|} = \frac{\bar{w}_0^k}{\|\bar{w}_0^k\|}. \tag{122}$$

Proof. With initialization $w_0^0 = \bar{w}_0^0 = 0$, the Modified Local-GD is just a scaling of vanilla Local-GD:

$$w_0^{k+1} = \frac{k}{k+1}\frac{1}{M}\sum_{i=1}^{M} w_i^{k+1}. \tag{123}$$

Also, the Modified PPM is a scaling of vanilla PPM: $\bar{w}_0^{k+1} = \frac{k}{k+1}\frac{1}{M}\sum_{i=1}^{M}\bar{w}_i^{k+1}$.

When $k \geq 1$, we can know the residual between Modified Local-GD and Modified PPM is

$$\begin{aligned}\|r^k\| &= \left\| w_0^k - \ln(\frac{1}{\lambda})\bar{w}_0^k \right\| = \frac{k}{k+1}\frac{1}{M}\left\| \sum_{i=1}^{M} w_i^k - \ln(\frac{1}{\lambda})\bar{w}_i^k \right\| \\ &\leq \frac{1}{M}\sum_{i=1}^{M}\left\| w_i^k - \ln(\frac{1}{\lambda})\bar{w}_i^k \right\| = \frac{1}{M}\sum_{i=1}^{M}\|r_i^k\|. \end{aligned} \tag{124}$$

Then we can follow the same process in the proof of Lemma 1 to obtain

$$\|r^k\| \leq \frac{1}{M}\|r_i^k\| = O(k\ln\ln(\frac{1}{\lambda})), \tag{125}$$

As a result we have $\lim_{\lambda\to 0}\frac{w_0^k}{\|w_0^k\|} = \frac{\bar{w}_0^k}{\|\bar{w}_0^k\|}$.

□