Modeling with Homophily Driven Heterogeneous Data in Gossip Learning

Abhirup Ghosh , **Cecilia Mascolo**
University of Cambridge, UK
{ag2187, cm542}@cam.ac.uk,

Abstract

Training deep learning models on data distributed and local to edge devices such as mobile phones is a prominent recent research direction. In a Gossip Learning (GL) system, each participating device maintains a model trained on its local data and iteratively aggregates it with the models from its neighbours in a communication network. While the fully distributed operation in GL comes with natural advantages over the centralized orchestration in Federated Learning (FL), its convergence becomes particularly slow when the data distribution is heterogeneous and aligns with the clustered structure of the communication network. These characteristics are pervasive across practical applications as people with similar interests (thus producing similar data) tend to create communities.

This paper proposes a data-driven neighbor weighting strategy for aggregating the models: this enables faster diffusion of knowledge across the communities in the network and leads to quicker convergence. We augment the method to make it computationally efficient and fair: the devices quickly converge to the same model. We evaluate our model on real and synthetic datasets that we generate using a novel generative model for communication networks with heterogeneous data. Our exhaustive empirical evaluation verifies that our proposed method attains a faster convergence rate than the baselines. For example, the median test accuracy for a decentralized bird image classifier application reaches 81% with our proposed method within 80 rounds, whereas the baseline only reaches 46%.

1 Introduction

Smart end devices like phones and wearables collect massive amounts of data characterizing personal, environmental, and societal signals using sensors like camera, microphone, accelerometer, etc. Learning patterns from such data is at the heart of many socio-technological systems across healthcare systems [Warnat-Herresthal *et al.*, 2021], smart homes [Aïvodji *et al.*, 2019], and smart cities [Zheng *et al.*, 2022]. However, a traditional machine learning system needs the data to be aggregated from all the devices: this imposes high communication costs and compromise user privacy, especially when the server cannot be trusted.

To mitigate such issues, a recent stream of research has proposed to keep the data local to the devices and iteratively aggregate the models learned from the data. The two most popular settings in this regime are Federated Learning (FL) [Kairouz *et al.*, 2021; McMahan *et al.*, 2017] and Gossip Learning (GL) [Koloskova *et al.*, 2020]. While the former needs a central server to aggregate the models, the latter operates in a completely distributed way.

A GL system works in rounds where at every round a device (also called node) aggregates (e.g., weighted average) the model parameters it receives from its neighbors in a communication network, updates the aggregated model using its local data, and shares it with its neighbors. The majority of the FL settings ignore the device-to-device direct communication opportunities and asks every device to connect to a single server which makes the server a single point of failure and highly costly to maintain. Due to its fully distributed operation, GL not only avoids these problems but also enjoys better privacy as the models are only communicated to a limited set of trusted neighbors instead of a potentially untrusted FL server.

Although the fully distributed operations in GL yield the above attractive characteristics, it introduces unique challenges due to inherent heterogeneity in the distribution of local datasets and the topology of the communication network. Below we discuss two major structures in such a communication network that are crucial for GL.

Communities or groups of densely connected nodes, naturally arise in communication networks [Onnela *et al.*, 2007; Pietilänen and Diot, 2012]. Nodes inside a community often are strongly connected in terms of the amount of time spent together or the level of interaction, etc. [Granovetter, 1973]. For example, family or close friendships form such tightly coupled communities.

Homophily is defined as the tendency of similar agents to connect to each other. It is one of the most basic social processes observed across human and animal societies [Talaga and Nowak, 2019; McPherson *et al.*, 2001; Hristova *et al.*, 2014]. For example, friends may have similar photos in their phones as they may go out together. Homophily has long been studied in various fields including complex networks [Hristova *et al.*, 2014], graph neural networks [Veličković *et al.*, 2017], and general machine learning [McAuley and Leskovec, 2012].

Community structure coupled with homophily gives rise to a data distribution that remains consistent among intra-cluster nodes but changes between clusters. For example, different families in a neighborhood will have different vacation photos, wildlife photographers from different countries photograph different geo-located species, etc.

1.1 Challenges

Such a homophily induced heterogeneity makes the convergence of existing GL aggregation methods [Koloskova *et al.*, 2020] extremely slow especially because the sparse inter-cluster connections hinder knowledge diffusion across communities. Existing literature takes two avenues to solve this issue: $i)$ by optimizing the communication network topology [Vanhaesebrouck *et al.*, 2017; Lian *et al.*, 2017] but in a cross-device setting, the network topology is naturally produced and cannot be optimized, for example, the friendship relations are not subject to computational optimization. $ii)$ customizing aggregation [Vogels *et al.*, 2021] which increases memory requirement at the nodes and thus is not suitable for our cross-device setting. Further, the gradient based bias correction FL methods developed to deal with heterogeneous data [Li *et al.*, 2020; Karimireddy *et al.*, 2020] become meaningless in GL as the gradients at different gossiping nodes are not aligned. Section 3.1 discusses these challenges in detail.

1.2 Our Contributions

To the best of our knowledge, we are the first to study the connection between the statistical data heterogeneity and the community structure of the communication network in the context of GL. Here we propose several important advancements toward making gossip learning amenable to real-world applications. We summarize our contributions as follows.

- We formally analyze why the most popular existing GL aggregation method is inadequate for our purpose and then propose a data-driven model aggregation strategy for faster convergence. Our main insight is that when aggregating the neighbors' models at a node u, a neighbor is more important if it has access to a different data distribution than u. We approximate this using the difference in the softmax distributions that the models produce given the same validation sample.
- We make the proposed method fair in the sense that every node reaches the same optimization level simultaneously. This is achieved by regularized local training prohibiting local models to go further from the aggregation state. As GL does not have a global synchronizing step a quick global consensus is non-trivial. Further, computational efficiency is attained by caching the neighbour importances saving their computation cost at each communication round and reducing the size of the validation set used in inferring the importance of a neighbor.

Algorithm 1: Gossip Learning

```
1: for Each round $t$ in $1, \dots$ do
2:     for (Each node $i \in 1, \dots, n$ in parallel) do
3:         Run $n_{ep}$ epochs of minibatch gradient descend with batch size $B$ on $M_i^{(t)}$ using loss function $\mathcal{L}(M_i^{(t)}, D_i)$ to get $M_i^{(t+1/2)}$
4:         Send $M_i^{(t+1/2)}$ to all neighbors of $i$ in $G$: $Ngh(i)$
5:         $M_i^{(t+1)} = W_{ii} M_i^{(t+1/2)} + \sum_{j \in Ngh(i)} W_{ij} M_j^{(t+1/2)}$
6:     end for
7: end for
```

- To better understand the interplay between the communication network topology and heterogeneous data distribution, we propose a generative model for the communication network. Conveniently, it has only two hyper-parameters controlling the data heterogeneity and homophily. Thus it is simple to understand but capable of generating a wide variety of datasets.
- We extensively test our method using synthetic and real-world datasets. We have used real interaction networks of students, the residence of a village, and a location based contact network for a bird image classification task. In all the experiments our method achieves faster convergence than the baseline.

2 Gossip Learning

Let us start by briefly discussing the basic working principles of GL. It (Algorithm 1) works in synchronous rounds. At the beginning of each round, a node i runs n_{ep} epochs of mini-batch gradient descent using its local dataset D_i on its local model M_i. One can choose a loss function suitable for the application at hand, we consider the cross entropy, $\mathcal{L}_{CE}(.)$ for simplicity. It then sends the updated model to its neighbors. Physically the communication can use Bluetooth, WiFi direct, other peer to peer (P2P) communication technologies, or end to end encrypted messages over the internet. Here, we abstract such a medium and assume that the underlying communication is reliable and accurate.

After receiving the models from the neighbors, each node computes a weighted average (with weights W_{ij} – weight of M_j at node i) of the model parameters (Line 5).

Aggregation Method. All prior works compute the neighbor weights using simple topological measures independent of the data distribution, for example, putting equal weights on all the neighbors. A popular method for aggregation is to use Metropolis-Hastings weighting strategy (MH. weighting) [Koloskova *et al.*, 2020] where $W_{ij} = W_{ji} = min\{\frac{1}{degree(i)+1}, \frac{1}{degree(j)+1}\}$.

3 Setup and Problem Formulation

Here, we formalize the setup and state our assumptions, constraints, and formally analyze why the above aggregation methods do not work.

Communication Network. We focus on cross-device GL between smart end devices that are inherently mobile and thus come in contact of different devices. To preserve privacy we consider the GL devices to communicate with a small set of trusted others. For simplicity, we extract and operate on a static communication network, $G = (V, E)$ from the dynamic communication opportunities, for example, considering that an edge exists when the communication frequency is beyond a threshold.

Data Distribution. Suppose, $\mathcal{D}$ denotes the aggregated data in the system (although such aggregation never happens). Each sample j is a tuple of feature and the target class, $(x^{(j)}, y^{(j)})$ and a dataset at node i is D_i, further all nodes have the same number of training samples.

This paper focuses on the data heterogeneity from the perspective of label skew across devices, i.e., local dataset D_i of a node i contains a subset of the classes or more generally a unbalanced number of samples from different classes.

Objective. Here, each node wants to learn about all the classes that exist in the system. In other words the test data contains i.i.d. samples from $\mathcal{D}$. As a node does not contain enough samples from all the classes, it needs to collaborate with others.

We aim to enable the nodes to reach high test accuracy quickly. This is driven by the fact that a quicker convergence will reduce the overall resource consumption. An extended goal is to make the convergence speed similar across nodes.

3.1 Problem With Existing GL Aggregation

A machine learning model becomes the most accurate when it is trained on the centralized data, i.e., local data from the source devices are aggregated. In the following we show that the existing gossip learning aggregation takes a different parameter update direction compared to such a centralized model and thus is sub-optimal.

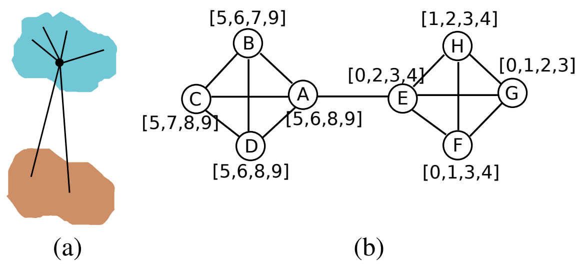


Figure 1: **(a)** Schematic graph used in Section 3.1. We show the connections of the node U_1. The graph has two clusters: G_R (top-blue) and G_L (bottom-brown). **(b)** A special case of (a) with a single bridge edge. Each node contains 100 samples from each noted class.

Let us consider a GL system as shown in Figure 1(a). For simplicity, assume that every node is connected to d nodes from its own cluster except, U_1 which connects to ψd nodes from G_R and $(1-\psi)d$ from G_L for $0 < \psi < 1$. Every node in G_L has classes $\{C_3, C_4\}$ and every node in G_R has classes $\{C_1, C_2\}$. Every node contains uniformly randomly sampled Δ samples of each class it has. For simplicity, we consider the classes are overall balanced, thus $n_L \approx n_R$ where n_L and n_R represent the number of nodes in G_L and G_R respectively. However, note that the label distribution across nodes is skewed.

Let us now analyze why existing gossip averaging scheme will not work in this distribution. While the simplistic setting is only for the theoretical understanding, the empirical studies consider more complex topology, data distribution, and more classes.

A typical neural network contains a feature extractor ($f_\phi : \mathbb{R}^{data\_dim} \rightarrow \mathbb{R}^h$) and a classifier ($g_\theta : \mathbb{R}^h \rightarrow \mathbb{R}^4$). For simplicity, let us consider that the feature extractor is frozen (e.g., using the weights from a model pre-trained on a larger dataset) and g_θ (one dense layer) is trained using GL. In the following we shall analyze how a parameter vector corresponding to a class C_i denoted as $\theta_{C_i}(t)$ changes as it is trained for more batches (t). For a data sample, x_i the network outputs a softmax probability $P_\theta(x_i, C_k) = \frac{exp(\theta_{C_k}^T f_\phi(x_i))}{\sum_{j\in\{1,\cdots 4\}} exp(\theta_{C_j}^T f_\phi(x_i))}$, i.e., the probability of x_i belonging to a class C_k.

Now, let us investigate θ_{C_1} in the following. Suppose, the network uses cross entropy error, $\mathcal{L}_{CE} = \sum_{x_i\in\mathcal{D}, j\in\{1\cdots 4\}} \mathcal{I}(y_i = C_j) \log P_\theta(f_\phi(x_i), C_j)$, where $\mathcal{I}$ is an identity function. Thus a full batch gradient decent will have $\theta_{C_1}(t+1) = \theta_{C_1}(t) - \eta\frac{\partial\mathcal{L}}{\partial\theta_{C_1}(t)}$, where η is the learning rate and $\frac{\partial\mathcal{L}}{\partial\theta_{C_1}(t)} = -\sum_{x_i\in\mathcal{D}, y_i=C_1}(1 - P_\theta(x_i, C_1))f(x_i) + \sum_{x_i\in D, y_i\neq C_1} P_\theta(x_i, C_1)f(x_i)$ is the gradient at $\theta_{C_1}(t)$. Let us denote the $\sum_{x_i\in\mathcal{D}, y_i=C_1}(1 - P_\theta(x_i, C_1))f(x_i)$ as $F_{pull}(C_1, |\mathcal{D}|)$ and $\sum_{x_i\in\mathcal{D}, y_i\neq C_1} P_\theta(x_i, C_1)f(x_i)$ as $F_{push}(\neg C_1, |\mathcal{D}|)$ as they pull and push the representation vector θ_{C_1} w.r.t. the embeddings of class C_1 and others respectively [Li and Zhan, 2021]($\neg C_1$ represents the set of classes other than C_1).

Further, we assume that the push and pull forces computed at different devices remain within a multiplicative factor of $c > 1$ to the expected force over the devices. For example, for pull forces, $\frac{1}{c} \leq \frac{F_{pull}(C_1,\Delta)}{\mathbb{E}[F_{pull}(C_1,\Delta)]} \leq c$. This is inline with the assumptions in a typical decentralized learning setup [Li *et al.*, 2019]. Moreover, assume that every device (and also the centralized model) initializes its model using the same weight vector, θ (say delivered along with the app).

Observation 3.1. When a model is trained on the data accumulated across all the devices then θ_{C_1} becomes following after one full batch gradient step:

$$\theta_{C_1}(0) + \frac{\eta}{c}z \leq \theta_{C_1}(1) \leq \theta_{C_1}(0) + \eta c.z \qquad (1)$$

Where, $z = \big(n_R\mathbb{E}[F_{pull}(C_1, \Delta)] - n_R\mathbb{E}[F_{push}(C_2, \Delta)] - n_L\mathbb{E}[F_{push}(C_3, \Delta)] - n_L\mathbb{E}[F_{push}(C_4, \Delta)]\big)$.

Now, we shall analyze GL operation at U_1, especially the parameter $\theta_{C_1}^{(U_1)}(1)$.

Lemma 3.1. Given the data distribution and communication network as described above, the model weight at U_1 becomes the following after one aggregation step using Metropolis-Hastings weighting.

$$\theta_{C_1}(0) + \frac{\eta.d}{c(d+1)} z \leq \theta_{C_1}^{(U_1)}(1) \leq \theta_{C_1}(0) + \frac{\eta.c.d}{d+1} z \quad (2)$$

Where $z = \psi(\mathbb{E}[F_{pull}(C_1, \Delta] - \mathbb{E}[F_{push}(C_2, \Delta)]) - (1 - \psi)(\mathbb{E}[F_{push}(C_3, \Delta] + \mathbb{E}[F_{push}(C_4, \Delta)])$

Proof. Let us only analyze the upper bound as the lower bound will follow a similar reasoning. After 1 full batch gradient decent, every device $U_i \in G_R$ will have $\theta_{C_1}^{U_i}(1/2) \leq \theta_{C_1}(0) + \eta c \mathbb{E}[F_{pull}(C_1, \Delta)] - \eta c \mathbb{E}[F_{push}(C_2, \Delta)]$. Similarly, any device $j \in G_L$ will have $\theta_{C_1}^{U_j}(1/2) \leq \theta_{C_1}(0) - c\eta\big(\mathbb{E}[F_{pull}(C_3, \Delta)] - \eta \mathbb{E}[F_{push}(C_4, \Delta)]\big)$. Thus, after aggregation at U_1, it'll have $\theta_{C_1}^{U_1}(1) \leq \theta_{C_1} + \frac{1}{d+1}\big(\theta_{C_1}^{U_1}(1/2) + \psi \sum_{U_i \in G_R \cap Ngh(U_1)} \theta_{C_1}^{U_i}(1/2) + (1 - \psi) \sum_{U_j \in G_L \cap Ngh(U_1)} \theta_{C_1}^{U_j}(1/2)\big)$. The statement follows trivially from this. $\square$

Discussions

Comparing the GL update in Equation 2 with the centralized model (Equation 1) we can see that the update vectors will have the same direction [1] when $n_R/(n_R + n_L) = \psi$. However, this contradicts with our homophily assumption; according to which ψ should be close to 1. Further, given this result, a node that is only connected to the nodes in the same cluster ($\psi = 1$) naturally will have its update in a different direction than the centralized update. These results are due to the difference in the data distribution among neighbors and globally across all nodes (which is likely to happen in reality).

Also note that when a node's model reaches another node it is reduced exponentially w.r.t. the distance, τ between the two nodes. In the above setting, it will reduce by a factor of $\frac{1}{(d+1)^\tau}$, simply due to multiple averaging steps along the path. As a consequence, while considering multiple GL rounds, the model parameters from distant nodes (e.g., ones in G_L) has limited effect.

It is possible to correct the direction in Lemma 3.1 to match the centralized update in Observation 3.1 if ψ is known. However, knowing this as part of system configuration violets privacy and it becomes a complex measurement when we consider multiple hops in the graph and multiple rounds in GL. Our method in the following section is a way to approximate this quantity.

4 Algorithms

We will now propose our algorithm for GL aggregation. The method follows Algorithm 1 and modify the W_{ii} and W_{ij} using a data-driven strategy. We first discuss the way to decide W_{ij} followed by how to choose W_{ii}.

[1]Two vectors $\overline{v}, \overline{u}$ are parallel when $\exists a$ such that $\overline{v} = a\overline{u}$.

Algorithm 2: Softmax-distribution-based weighting at node i with validataion set Q_i and model M_i

```
1: Compute $H_i^{(k)} = M_i^{(t)}(q_i^{(k)})$ for all $q_i^{(k)} \in Q_i$
2: Compute $H_j^{(k)} = M_j^{(t)}(q_i^{(k)})\ \forall q_i^{(k)} \in Q_i, j \in Ngh(i)$
3: for Each neighbor $j \in Ngh(i)$ do
4:     $\Delta_{ij}$ = pairwise cosine distances between $H_i^{(k)}$, $H_j^{(k)}$
5:     $dist_{ij}$ = Average($\Delta_{ij}$) over samples in $Q_i$
6: end for
7: $W_{ij} = (1 - W_{ii}).\frac{dist_{ij}^\alpha}{\sum_{j \in Ngh(i)} dist_{ij}^\alpha}$
```

4.1 Softmax-Distribution-Based Weighting

Let us consider the set up in Figure 1(b). This graph has minimum diffusion opportunity (only one bridge edge) and maximal dispersion of the data distribution (no common class between clusters). Intuitively, following are the required characteristics of a neighbor weighting scheme: i) the bridge edge AE is the most important edge in this network as it connects two clusters containing knowledge about different sets of classes. ii) the node importance is asymmetric, for example, A is important to D as it gives access to the other cluster. However, D is not important to A as they both have access to similar classes and A has other neighbours B and C with similar classes as D.

Algorithm 2 summarizes the proposed method. Each node i sets aside a set of validation examples, Q_i from its local data. Then upon receiveing the models from its neighbors, it infers the softmax probability distribution for a validation example $q_i^{(k)} \in Q_i$ using its own model $H_i^{(k)} = M_i^{(t)}(q_i^{(k)})$ and as well as using the model from its neighbor j: $H_j^{(k)} = M_j^{(t)}(q_i^{(k)}) \forall j \in Ngh(i)$. Models trainined with similar data distribution will produce similar softmax probabilities whereas the softmax probabilities will differ when the training distribution differs between the nodes. We compute the mean of the cosine distances, $dist_{ij}$, between each pair $H_i^{(k)}$ and $H_j^{(k)}$ over all validation samples in Q_i. We then set the neighbor weight as $W_{ij} \propto (dist_{ij})^\alpha$.

The hyperparameter α is an amplification factor. When $\alpha = 0$, all the neighbours have equal importance and a larger value of α can amplify the small differences between nodes.

4.2 Deciding W_{ii}

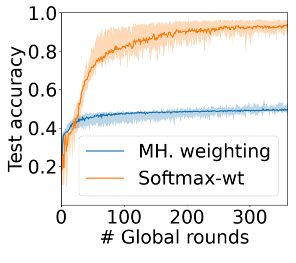
We initialize W_{ii} to 0.5 and reduce it every epoch using the function: $W_{ii} = 0.5(\log(t))^{-\beta}$, where t denotes the cumulative number of local epochs till now. The hyper parameter β controls the rate of decrease, for example if $\beta = 0$, then W_{ii} remains fixed at 0.5 and a larger value of β reduces W_{ii} at a faster rate.

Ideally, β should depend on the dataset and factors related to training, however, we keep this out of the scope of the current work.

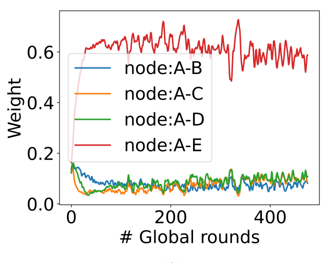
4.3 Empirical Insights for Algorithm 2

Let us now empirically analyse the effect of Algorithm 2 on the convergence rate using the dataset in Figure 1(b).

The convergence for Algorithm 2 is much faster than the baseline strategy (Figure 2 (b)). The neighbor weights also capture our intuitions (Figure 2(c) and results in technical supplementary [2]). The baseline [Koloskova *et al.*, 2020] test accuracy plateaus at the 50%, i.e., the nodes learn the classes in the same cluster but the knowledge does not percolate across clusters. Slight variations of this baseline show similar pattern as shown in the supplementary.



(a)



(b)

Figure 2: **(a)** Algorithm 2 achieves much faster convergence than the baseline. Here, $\alpha = 4$ and $\beta = 4$, Validation set contains random 10% samples from local data at each node. The shaded region contains the test accuracy of all the nodes in the network and the bold line is the median. **(c)** neighbor weights at A (one of the two bridge nodes). They correctly capture the intuition of 'important' nodes.

5 Practicalities: Fairness and Efficiency

To make gossip learning practical, this section considers two important factors: how to make the convergence speed similar at different nodes and how to make the method more efficient.

5.1 Fairness of Node Accuracy

Unlike FL where there is a single global model parameter state, gossiping nodes can have very different model parameters, especially if they are far apart in the communication network. As a result, nodes will have different convergence rates based on their local data and position in the network. Here, we extend Algorithm 2 to make the model parameters across all the nodes achieve similar convergence rates. Many applications need this to make the model quality fair across participants.

To achieve such fairness, node i regularizes its local training using the following loss function at epoch t

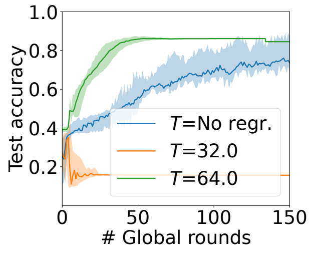
$$\mathcal{L} = (1-\lambda_t)\mathcal{L}_{CE}(M_i^{(t)}, D_i) + \lambda_t \left\| M_i^{(t)} - M_i^{(t-1)} \right\|_2^2 \quad (3)$$

$\mathcal{L}_{CE}$ denotes the cross entropy loss. The relative weighting of the two loss terms is controlled by λ_t which changes as $\lambda_t = \left(1 + \exp\left(\frac{-t}{T}\right)\right)$. Here, $\lambda_t \to 1/2$ when $T \to \infty$ and for $T \to 0$ we get $\lambda_t \to 1$. The hyperparameter T controls the growth rate of λ_t. Setting λ_t to a constant does not work as at the initial stage of the optimization, a heavy regularization is detrimental and a light regularization allows the models to diverge at a later stage. Figure 3(a) empirically verifies the strategy and shows that fairness is attained with our strategy.

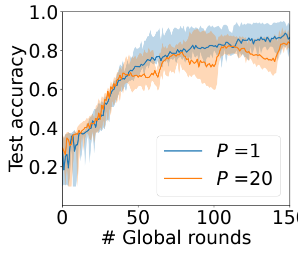
[2]Technical supplementary and code: https://t.ly/xSnS

5.2 Compute Efficiency

Algorithm 2 needs more computation time and resources than the vanilla GL as at each round a node needs to run inferences on $|Q_i| \times (|Ngh(i)| + 1)$ samples. Here we consider caching the weights (W_{ij}) and recomputing them every P rounds. Figure 3(b) shows that the node accuracies decrease with increasing values of P, however, the reduction is gradual and thus can be used to trade-off between the efficiency of the method versus the convergence speed. Further, we show in the supplementary that with smaller $|Q_i|$ the accuracy degrades, but still remains usable.



(a)



(b)

Figure 3: All experiments here use Figure 1(b). **(a)** A lower $T(32)$ makes the regularization so strong from the beginning that none of the nodes learns anything useful. With $T = 64$ all the nodes quickly converge at the same model parameters and all of them achieve a decent accuracy. **(b)** Here we have $\alpha = 4$ and $\beta = 4$.

6 Generative Model for a Synthetic Dataset

To evaluate our proposed method along with real datasets, here we create a simple generative framework for synthetic datasets. While both social network datasets and machine learning datasets are abundant in the literature, it is rare to find their combination, especially with human interaction networks. Moreover, while modeling social network has been studied extensively [Kempe, 2011], no existing model captures the heterogeneous distribution of data on the network. This framework naturally produces communities in the network and the data distribution aligns with the communities.

For a $|C|$ class classification task, following the literature in FL [Hsieh *et al.*, 2020] here each node samples $Z \sim Dir(\frac{\gamma}{|C|}\mathbb{I}^{|C|}), \gamma \in \mathbb{R}$. It then chooses Z_i fraction of its local dataset uniformly randomly from the i-th class. The non-iidness is controlled by γ. A smaller γ produces more imbalanced distribution.

Following a popular generative social network model called as Social Distance Attachment (SDA) [Boguná *et al.*, 2004; Talaga and Nowak, 2019] we connect a node pair i, j with probability $p_{ij} = \frac{1}{1+(d_{ij}/b)^{\xi}}$ where d_{ij} is the distance in the feature space, b is the characteristic distance, and ξ controls the extent of homophily. We use a k-neare-t neighbor based distance: $d_{ij} = max(k_{ij}, k_{ji})$, where the node i is the k_{ij}th nearest neighbor of node j and j is the k_{ji}th neighbor of i. We use such a distance as it has consistent scale across any data distribution and number of classes (making $b = 1$). However, it can be easily replaced by other distance metrics

	$\lvert V\rvert$	$\lvert E\rvert$
Network in Figure 1(b)	8	13
Contacts of school students	64	127
Contacts of village residents	19	29
Contacts of bird image contributors	28	66
Synthetic data using SDAH	80	variable

Table 1: Summary of communication networks evaluated.

such as Euclidean. We call it SDA for Heterogeneous data (SDAH). We theoretically and empirically analyze this model in supplementary.

7 Experiments

The gossip learning setup has two main components – the data on the nodes and the communication network. Here the Algorithm 2 is evaluated using four image datasets, one time series dataset, two synthetic and three real-world communication networks. We distribute MNIST dataset on different communication networks in Table 1 (except the bird image contributor contacts). We use the network in Figure 1(b) to test other datasets: Fashion MNIST [Xiao *et al.*, 2017], CIFAR-10 [3], and UCI Human Activity Recognition (UCI-HAR) [Anguita *et al.*, 2013]. Following are the main results. Details on setups and datasets are in supplementary.

- In all the evaluation settings Algorithm 2 achieves a faster convergence than the baseline strategy.
- We build a location based communication network between contributors for a bird image classification task. It verifies that our assumptions on data distribution hold in the real world. Our method achieves faster convergence in this dataset too.
- Algorithm 2 achieves slower convergence compared to FedAvg but achieves better final accuracy numbers.

Impact of Hyperparameters. Figure 4 shows that with different α and β convergence rates vary, but the final accuracies remain similar. One needs to tune these hyperparameters to get the best convergence rate. However, we leave this to application developers.

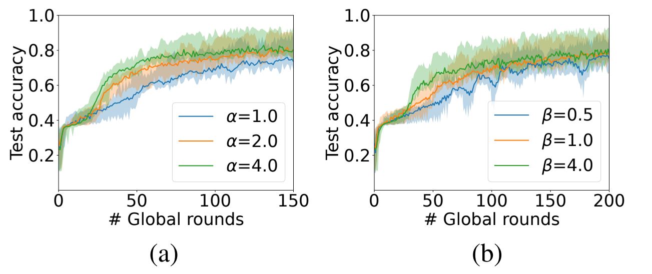


Figure 4: α and β are set to 1 when not varied. All experiments use the setup in Figure 1(b). **(a)** With increasing α convergence speed increases. **(b)** Higher β produces faster convergence.

[3]https://pytorch.org/vision/stable/generated/torchvision.datasets.CIFAR10.html

Using Public Datasets. We use the communication network in Figure 1(b) along with Fashion-MNIST (result in technical supplementary), Cifar-10, and UCI-HAR datasets. While the former two datasets have 10 classes, UCI-HAR has 6 activity classes. For the UCI-HAR dataset we put three random classes in each cluster in Figure 1(b). CIFAR-10 is distributed using the dirichlet distribution with $\gamma = 1$. Our method achieves faster convergence across all three datasets (Figure 5).

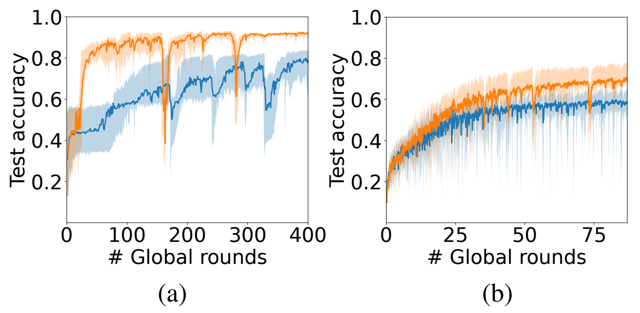


Figure 5: Results for UCI-HAR, and CIFAR-10 distributed on the synthetic communication network in Figure 1(b). All of them use $\alpha = 1$ and $\beta = 1$. The color scheme follows Figure 2(a).

Using Real-world Contact Graphs. We distribute MNIST on two real-world contact graphs: i) from primary school students [Génois and Barrat, 2018] considering that the students from the same class will have access to the similar class examples. ii) contact graph of residences of a village [Ozella *et al.*, 2021] considering that the members of the same family have access to the similar classes. Our method achieves faster convergence compared to baseline (Figure 6).

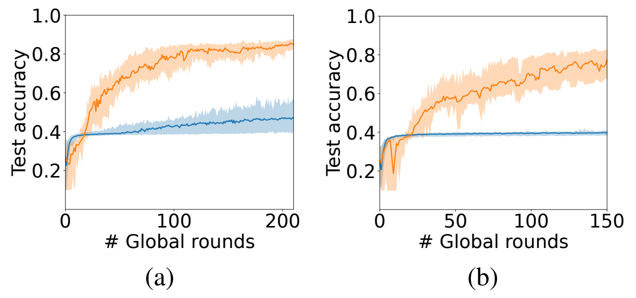
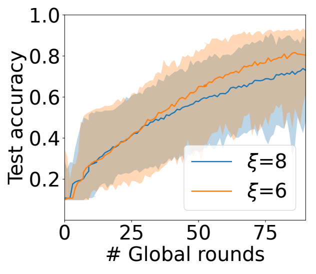


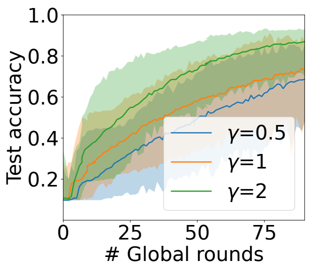
Figure 6: Our method achieves faster convergence rate for both the face to face communication network for the primary school students **(a)** and of village residences **(b)**. Both use $\alpha = 16$ and $\beta = 4$. The color scheme follows Figure 2(a).

Evaluating on SDAH Graphs. Multiple datasets are generated using our SDAH model with varying homophily (ξ) and heterogeneity (γ). We distribute MNIST images on the generated graphs. Algorithm 2 achieves faster convergence than baseline and follows natural trends (Figure 7).

Bird Image Classification. A dataset built from iNatural-



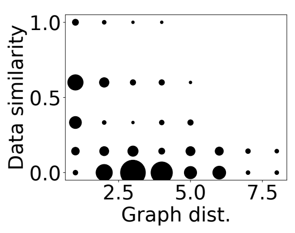
(a)



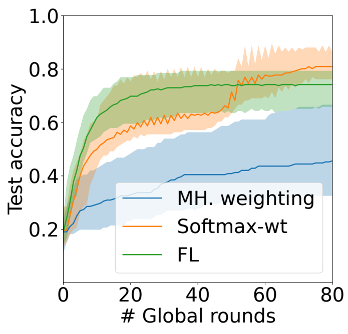
(b)

Figure 7: **(b)** With increasing ξ the network contains more sparsely connected communities and thus the knowledge percolation becomes more challenging. Here $\gamma = 1$ and $b = 10$. **(c)** With decreasing γ the data becomes more heterogeneous and learning becomes more challenging. Here $\xi = 8$ and $b = 10$. In all experiments here $\alpha = 4, \beta = 4$.

ists [4]. We include 20 popular bird species from four continents. A communication network among the contributors is built using their photo locations – connect two nodes if they have uploaded photos from the same state. Figure 8(a) shows that there is no node pair that are far apart in the network but have similar classes. This verifies that our assumptions on data distribution hold in practice. We use an EfficientNet B0 [Tan and Le, 2019] model pretrained with Imagenet. We only train the last layer with a fully connected layer and freeze the rest of the model. Algorithm 2 achieves faster convergence than the baseline (Figure 8(b)).



(a)



(b)

Figure 8: **(a)** Comparison of shortest path distance versus the similarity of the local class distribution **(b)** Our method ($\alpha = 16$ and $\beta = 4$) converges faster than the baseline. It produces slightly higher final accuracy compared to FedAvg but slower convergence.

Comparison With Federated Learning. In Figure 8(b), we keep the local data distribution fixed as the above bird image classification and compare with FedAvg [McMahan *et al.*, 2017]. The convergence in FL is faster as the central server 'sees' all the nodes at every round whereas the GL needs several rounds to spread information across all nodes. We believe the increase in the accuracy for GL stems from the effect of class imbalance: some classes are globally rare (see supplementary). While FL has known weakness in learning rare classes [Wang *et al.*, 2021], due to limited mixing of models, GL learns them better.

[4]https://www.inaturalist.org/

8 Related Works

While decentralized convex optimization has long been studied [Tsitsiklis, 1984], only recently it has been applied for machine learning [Koloskova *et al.*, 2020; Taheri *et al.*, 2020; Lalitha *et al.*, 2018; Lian *et al.*, 2018]. It is shown that the data heterogeneity reduces the convergence speed [Koloskova *et al.*, 2020]. Several recent papers have improved the distributed SGD method to address this [Tang *et al.*, 2018; Lin *et al.*, 2021; Yuan *et al.*, 2021]. However, the interplay between the network topology is not considered.

Further, it has been established that the topology is important in GL [Neglia *et al.*, 2020; Ying *et al.*, 2021; Giaretta and Girdzijauskas, 2019], e.g., local connectivity features are more useful than global spectral gap in theoretical analysis [Vogels *et al.*, 2023]. However, these works do not optimize learning for heterogeneous data and topology.

A few recent papers build an optimal communication network to alleviate data heterogeneity [Bellet *et al.*, 2021; Bars *et al.*, 2022; Onoszko *et al.*, 2021]. However, in our applications both the communication network and the data distribution are naturally fixed, for example, the trust relationships do not change based on optimization objective. [Vogels *et al.*, 2021] has addresses heterogeneity by building a spanning tree so that each node has the correct average model from all nodes. However, each node needs to maintain a model corresponding to each of its neighbors which will be challenging for resource constraint devices. Further, [Dandi *et al.*, 2022] proposes a solution that needs an all-reduce step which defeats the purpose of a fully distributed operation.

Data heterogeneity has been well studied in FL [Li *et al.*, 2020; Karimireddy *et al.*, 2020]. However, as these methods regularize the local gradient updates they are unsuitable for GL as gossiping nodes start a round at different states. Further, the topology of the network makes the setting more complex. Moreover, the alignment based model fusions in FL [Wang *et al.*, 2020] are orthogonal to our method and can potentially be combined.

9 Conclusion

We have demonstrated that a softmax-distribution-based neighbor weighting strategy achieves faster convergence speed for a variety of machine learning tasks and communication networks. Unlike prior strategies, our method is data-driven and thus can identify the important neighbors even over multiple hops. The method does not need to be fed with the topology and testing its suitability in dynamic networks can be an interesting future research. We also have proposed a method to generate synthetic data distribution over a communication network and a novel bird classification dataset. Our paper takes important steps toward making gossip learning practical and paves future research directions.

Acknowledgements

We acknowledge ERC Project grant 833296 (EAR).

References

[Aïvodji *et al.*, 2019] Ulrich Matchi Aïvodji, Sébastien Gambs, and Alexandre Martin. Iotfla: A secured and privacy-preserving smart home architecture implementing federated learning. In *2019 IEEE Security and Privacy Workshops (SPW)*, pages 175–180. IEEE, 2019.

[Anguita *et al.*, 2013] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra Perez, and Jorge Luis Reyes Ortiz. A public domain dataset for human activity recognition using smartphones. In *Proceedings of the 21th international European symposium on artificial neural networks, computational intelligence and machine learning*, pages 437–442, 2013.

[Bars *et al.*, 2022] B Le Bars, Aurélien Bellet, Marc Tommasi, and Anne-Marie Kermarrec. Yes, topology matters in decentralized optimization: Refined convergence and topology learning under heterogeneous data. *arXiv preprint arXiv:2204.04452*, 2022.

[Bellet *et al.*, 2021] Aurélien Bellet, Anne-Marie Kermarrec, and Erick Lavoie. D-cliques: Compensating noniidness in decentralized federated learning with topology. *arXiv preprint arXiv:2104.07365*, 2021.

[Boguná *et al.*, 2004] Marián Boguná, Romualdo Pastor-Satorras, Albert Díaz-Guilera, and Alex Arenas. Models of social networks based on social distance attachment. *Physical review E*, 70(5):056122, 2004.

[Dandi *et al.*, 2022] Yatin Dandi, Anastasia Koloskova, Martin Jaggi, and Sebastian U Stich. Data-heterogeneity-aware mixing for decentralized learning. *arXiv preprint arXiv:2204.06477*, 2022.

[Génois and Barrat, 2018] Mathieu Génois and Alain Barrat. Can co-location be used as a proxy for face-to-face contacts? *EPJ Data Science*, 7(1):11, 2018.

[Giaretta and Girdzijauskas, 2019] Lodovico Giaretta and Šarūnas Girdzijauskas. Gossip learning: Off the beaten path. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 1117–1124. IEEE, 2019.

[Granovetter, 1973] Mark S Granovetter. The strength of weak ties. *American journal of sociology*, 78(6):1360–1380, 1973.

[Hristova *et al.*, 2014] Desislava Hristova, Mirco Musolesi, and Cecilia Mascolo. Keep your friends close and your facebook friends closer: A multiplex network approach to the analysis of offline and online social ties. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 8, pages 206–215, 2014.

[Hsieh *et al.*, 2020] Kevin Hsieh, Amar Phanishayee, Onur Mutlu, and Phillip Gibbons. The non-iid data quagmire of decentralized machine learning. In *International Conference on Machine Learning*, pages 4387–4398. PMLR, 2020.

[Kairouz *et al.*, 2021] 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. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

[Karimireddy *et al.*, 2020] 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*, pages 5132–5143. PMLR, 2020.

[Kempe, 2011] David Kempe. Structure and dynamics of information in networks. *Lecture Notes*, 2011.

[Koloskova *et al.*, 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 *International Conference on Machine Learning*, pages 5381–5393. PMLR, 2020.

[Lalitha *et al.*, 2018] Anusha Lalitha, Shubhanshu Shekhar, Tara Javidi, and Farinaz Koushanfar. Fully decentralized federated learning. In *Third workshop on Bayesian Deep Learning (NeurIPS)*, 2018.

[Li and Zhan, 2021] Xin-Chun Li and De-Chuan Zhan. Fedrs: Federated learning with restricted softmax for label distribution non-iid data. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 995–1005, 2021.

[Li *et al.*, 2019] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*, 2019.

[Li *et al.*, 2020] 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, 2020.

[Lian *et al.*, 2017] Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. *Advances in Neural Information Processing Systems*, 30, 2017.

[Lian *et al.*, 2018] Xiangru Lian, Wei Zhang, Ce Zhang, and Ji Liu. Asynchronous decentralized parallel stochastic gradient descent. In *International Conference on Machine Learning*, pages 3043–3052. PMLR, 2018.

[Lin *et al.*, 2021] Tao Lin, Sai Praneeth Karimireddy, Sebastian U Stich, and Martin Jaggi. Quasi-global momentum: Accelerating decentralized deep learning on heterogeneous data. *arXiv preprint arXiv:2102.04761*, 2021.

[McAuley and Leskovec, 2012] Julian McAuley and Jure Leskovec. Image labeling on a network: using social-network metadata for image classification. In *European conference on computer vision*, pages 828–841. Springer, 2012.

[McMahan *et al.*, 2017] 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*, pages 1273–1282. PMLR, 2017.

[McPherson *et al.*, 2001] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, pages 415–444, 2001.

[Neglia *et al.*, 2020] Giovanni Neglia, Chuan Xu, Don Towsley, and Gianmarco Calbi. Decentralized gradient methods: does topology matter? In *International Conference on Artificial Intelligence and Statistics*, pages 2348–2358. PMLR, 2020.

[Onnela *et al.*, 2007] Jukka-Pekka Onnela, Jari Saramäki, Jörkki Hyvönen, Gábor Szabó, M Argollo De Menezes, Kimmo Kaski, Albert-László Barabási, and János Kertész. Analysis of a large-scale weighted network of one-to-one human communication. *New journal of physics*, 9(6):179, 2007.

[Onoszko *et al.*, 2021] Noa Onoszko, Gustav Karlsson, Olof Mogren, and Edvin Listo Zec. Decentralized federated learning of deep neural networks on non-iid data. *arXiv preprint arXiv:2107.08517*, 2021.

[Ozella *et al.*, 2021] Laura Ozella, Daniela Paolotti, Guilherme Lichand, Jorge P Rodríguez, Simon Haenni, John Phuka, Onicio B Leal-Neto, and Ciro Cattuto. Using wearable proximity sensors to characterize social contact patterns in a village of rural malawi. *EPJ Data Science*, 10(1):46, 2021.

[Pietilänen and Diot, 2012] Anna-Kaisa Pietilänen and Christophe Diot. Dissemination in opportunistic social networks: the role of temporal communities. In *Proceedings of the thirteenth ACM international symposium on Mobile Ad Hoc Networking and Computing*, pages 165–174, 2012.

[Taheri *et al.*, 2020] Hossein Taheri, Aryan Mokhtari, Hamed Hassani, and Ramtin Pedarsani. Quantized push-sum for gossip and decentralized optimization over directed graphs. *arXiv preprint arXiv:2002.09964*, 2020.

[Talaga and Nowak, 2019] Szymon Talaga and Andrzej Nowak. Homophily as a process generating social networks: insights from social distance attachment model. *arXiv preprint arXiv:1907.07055*, 2019.

[Tan and Le, 2019] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[Tang *et al.*, 2018] Hanlin Tang, Xiangru Lian, Ming Yan, Ce Zhang, and Ji Liu. D2: Decentralized training over decentralized data. In *International Conference on Machine Learning*, pages 4848–4856. PMLR, 2018.

[Tsitsiklis, 1984] John Nikolas Tsitsiklis. Problems in decentralized decision making and computation. Technical report, Massachusetts Inst of Tech Cambridge Lab for Information and Decision Systems, 1984.

[Vanhaesebrouck *et al.*, 2017] Paul Vanhaesebrouck, Aurélien Bellet, and Marc Tommasi. Decentralized collaborative learning of personalized models over networks. In *Artificial Intelligence and Statistics*, pages 509–517. PMLR, 2017.

[Veličković *et al.*, 2017] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[Vogels *et al.*, 2021] Thijs Vogels, Lie He, Anastasiia Koloskova, Sai Praneeth Karimireddy, Tao Lin, Sebastian U Stich, and Martin Jaggi. Relaysum for decentralized deep learning on heterogeneous data. *Advances in Neural Information Processing Systems*, 34:28004–28015, 2021.

[Vogels *et al.*, 2023] Thijs Vogels, Hadrien Hendrikx, and Martin Jaggi. Beyond spectral gap (extended): The role of the topology in decentralized learning. *arXiv preprint arXiv:2301.02151*, 2023.

[Wang *et al.*, 2020] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. *arXiv preprint arXiv:2002.06440*, 2020.

[Wang *et al.*, 2021] Lixu Wang, Shichao Xu, Xiao Wang, and Qi Zhu. Addressing class imbalance in federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10165–10173, 2021.

[Warnat-Herresthal *et al.*, 2021] Stefanie Warnat-Herresthal, Hartmut Schultze, Krishnaprasad Lingadahalli Shastry, Sathyanarayanan Manamohan, Saikat Mukherjee, Vishesh Garg, Ravi Sarveswara, Kristian Händler, Peter Pickkers, N Ahmad Aziz, et al. Swarm learning for decentralized and confidential clinical machine learning. *Nature*, 594(7862):265–270, 2021.

[Xiao *et al.*, 2017] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[Ying *et al.*, 2021] Bicheng Ying, Kun Yuan, Yiming Chen, Hanbin Hu, Pan Pan, and Wotao Yin. Exponential graph is provably efficient for decentralized deep training. *Advances in Neural Information Processing Systems*, 34:13975–13987, 2021.

[Yuan *et al.*, 2021] Kun Yuan, Yiming Chen, Xinmeng Huang, Yingya Zhang, Pan Pan, Yinghui Xu, and Wotao Yin. Decentlam: Decentralized momentum sgd for large-batch deep training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3029–3039, 2021.

[Zheng *et al.*, 2022] Zhaohua Zheng, Yize Zhou, Yilong Sun, Zhang Wang, Boyi Liu, and Keqiu Li. Applications of federated learning in smart cities: recent advances, taxonomy, and open challenges. *Connection Science*, 34(1):1–28, 2022.