# LEASGD: an Efficient and Privacy-Preserving Decentralized Algorithm for Distributed Learning

Hsin-Pai (Dave) Cheng, Patrick Yu, Haojing Hu, Yiran Chen, Hai (Helen) Li
Department of Electrical and Computer Engineering, Duke University
University of Nevada, Reno

## Introduction

- Decentralized Topology has become a popularized efficient and faster alternative to the Centralized Topology
- Differential Privacy has been previously applied in a centralized setting
- Yet limited work has been done in decentralized *and* differentially private collaborative learning, and existing implementations are sensitive to noise and larger datasets

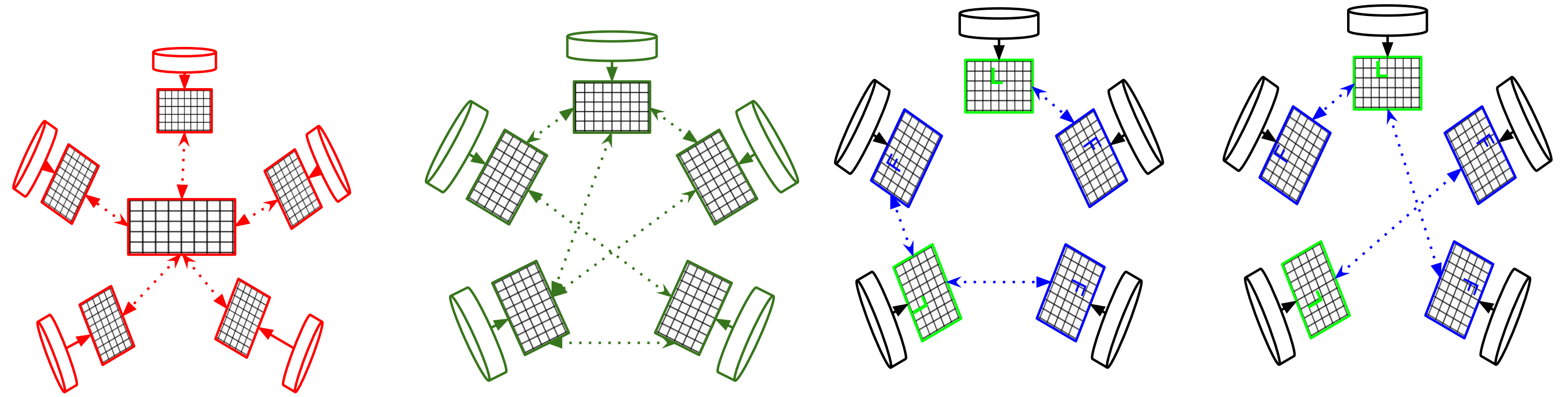


Figure 1. Left: A centralized topology involves a central server and various nodes that communicate with it. Middle: A decentralized topology avoids the central parameter server and instead relies on neighbor nodes to exchange gradients or weights. Right:  A LEASGD topology involves a changing, leader/follower based topology. L indicates leader nodes, and F indicates follower nodes.

## Motivation

***Decentralized Parallel Stochastic Gradient Descent***

- Based off asynchronous transfer and averaging of weights to help each other learn
- Sensitive to noise addition: differentially private noise causes accuracy to rapidly decline
- Based off a fixed topology - **slower data exchange and thus convergence**

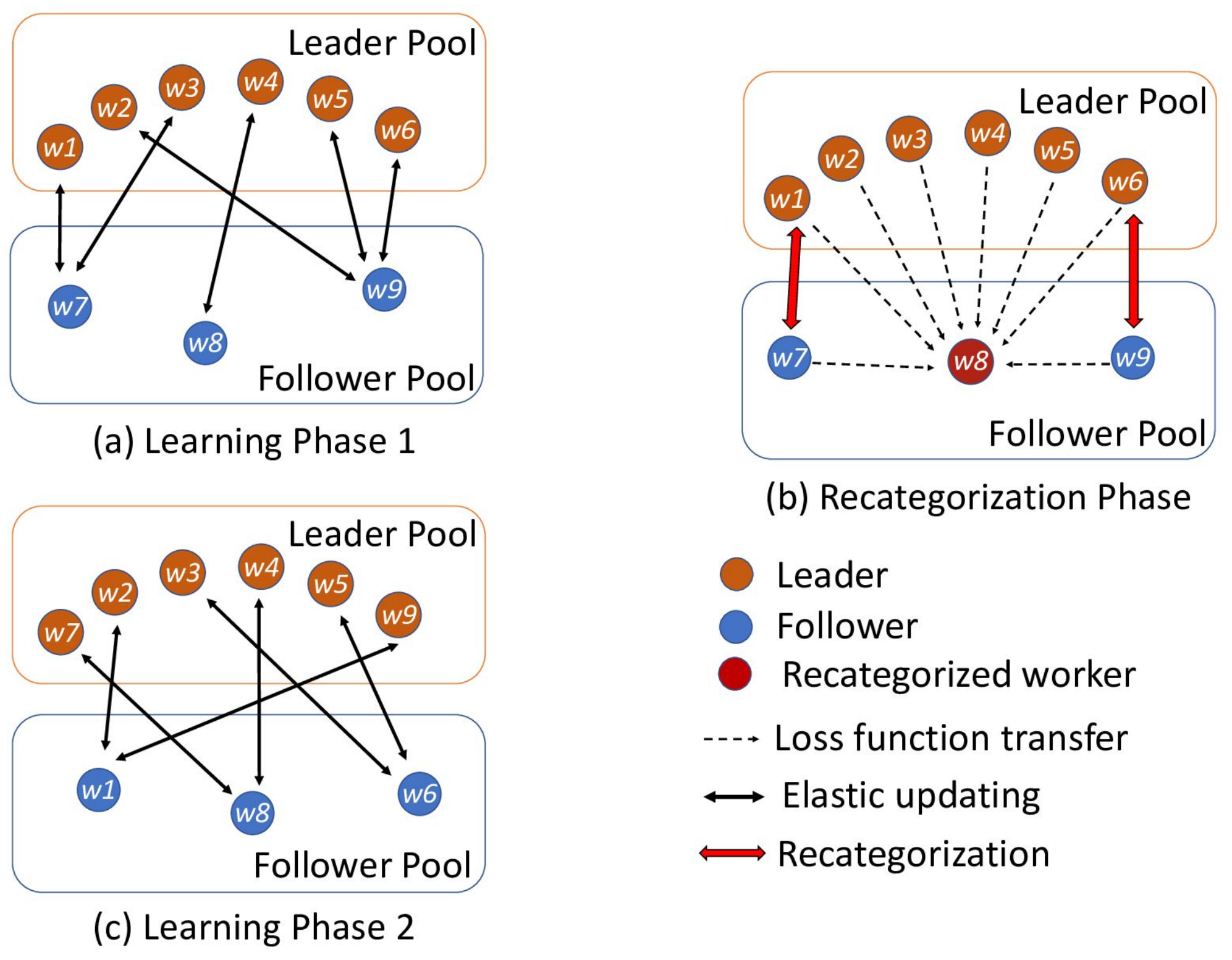
***Personalized and Private Peer-to-Peer Machine Learning***

- Implementation of DPSGD while introducing differential privacy
- Accurate for small networks and datasets
- **Unscalable**: fixed architecture makes it hard to scale up to bigger networks (Convolutional Networks such as VGG, etc.)

## Differentially Private Accountant

- Differential Privacy quantifies privacy in the form of a **privacy budget given by (ε,δ)** where epsilon quantizes the magnitude of privacy where a smaller epsilon indicates more privacy (due to noise), and delta is an "error term" that allows ε-privacy to be violated at a probability given by δ
- Our method uses the recently developed **Moments Accountant**, which provides tighter bound on ε for a certain privacy level - where the level of added noise to satisfy (ε,δ)-DP is proportional to the timestep/sampling propability and inversely proportional to epsilon and delta
- Our results show that the network is less susceptible to being affected by the addition of differentially private noise, when compared to larger networks such as the fixed topology given by DPSGD

## Leader-Follower Structure



(a) Learning Phase 1

(b) Recategorization Phase

(c) Learning Phase 2

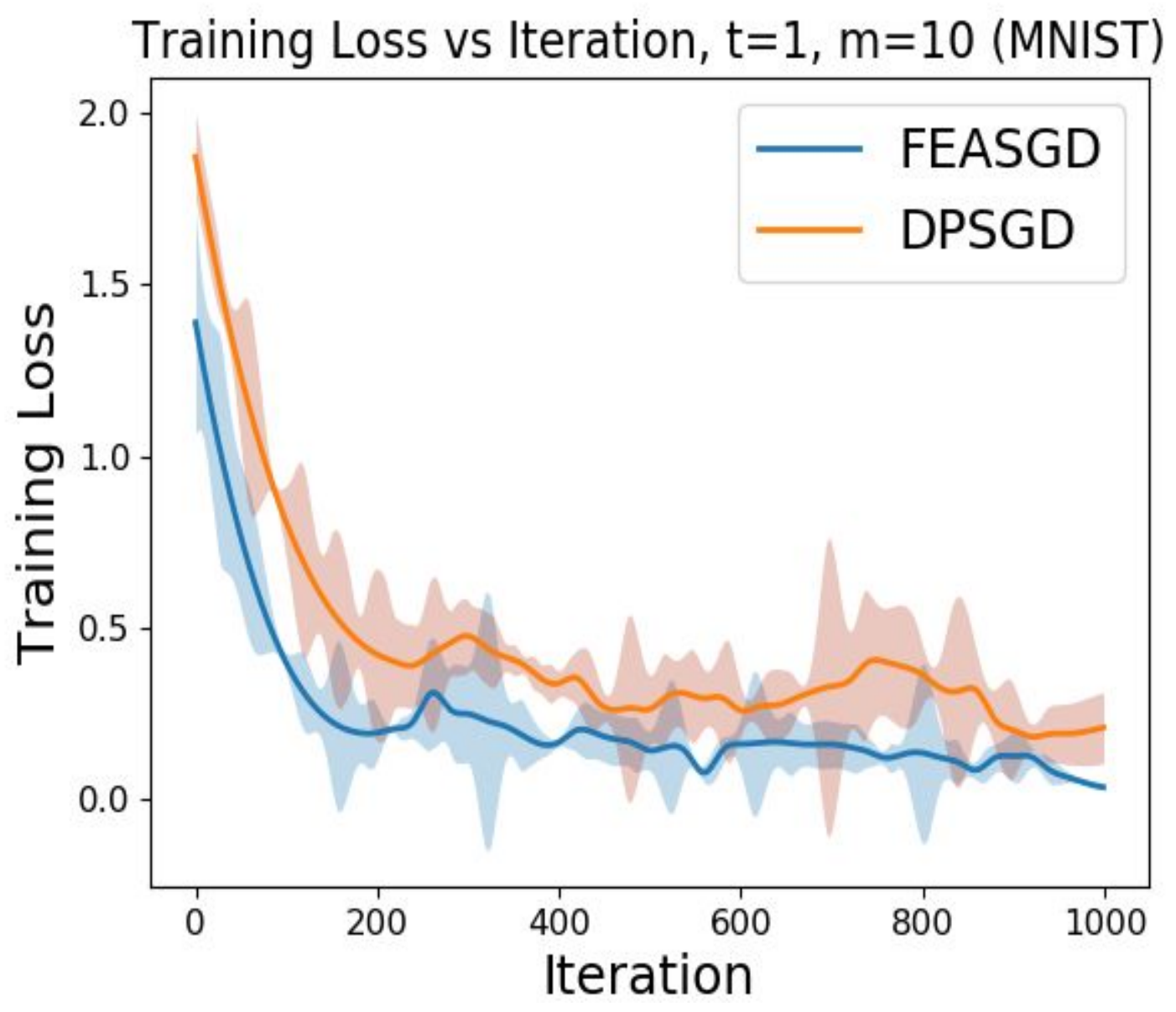
- The Leader-Follower structure aims to have higher-performing nodes interact more with and "teach" lower-performing nodes
- Performance of each node is determined by the nodes' loss function over a constantly updated minibatch
- Leaders/followers are periodically replaced in response to the changing behaviors, and thus performances, of individual nodes

## Loss Function

$$\arg\max_{w^i \in \Omega} \overline{F}(w,T) = \frac{1}{m}\sum_{i=1}^{m} f_T^i(w^i)$$
$$s.t.\, \Omega \subseteq \mathbb{R}^n \; and \, T \in \mathbb{R}$$
$$w^* = \{w^1, \dots, w^m\}$$
$$w_{t+1}^i = w_t^i - \eta g_t^i + \eta\rho(w_t^f - w_t^i)$$
$$w_{t+1}^f = w_t^f - \eta g_t^f + \eta\rho(w_t^i - w_t^f)$$
$$\tilde{g}_t^i = \overline{g}_t^i + \mathcal{N}(0, \sigma_2^2 C^2)$$

- The objective function aims to minimize the average SGD loss function over all the nodes
- Gradients of each weight is added with a elastic factor α = ηρ, representing the weight averaging operation
- The gradients are then added with Gaussian noise with magnitude proportional to sigma (noise as an effect of differential privacy) and a constant C.

## Results



| Algorithm | Final Accuracy | Total ε |
|---|---|---|
| LEASGD(m=5) | 0.97 | 4.183 |
| DPSGD(m=5) | 0.97 | 4.505 |
| LEASGD(m=15) | 0.97 | 4.651 |
| DPSGD(m=15) | 0.95 | 4.843 |

- LEASGD shows an improvement in both differentially-private and non-DP situations
- Consumes lower ε vs. DPSGD at similar accuracy percent levels
- Lower training loss at a specific iterations

## Discussion

*Given a system with $p \in \mathbb{N}$ workers with constant μ, weights **w**, gradient variance bound $\sigma_1$, learning rate **η** , exploration/exploitation tradeoff term **ρ**, and **α,β** as elastic transfer rates, number of followers L, if $0 \le \eta \le \frac{2(1-\beta)}{\mu+L}, 0 \le \alpha = \eta\rho < 1, 0 \le \beta = p\alpha < 1$ then we obtain bounds for the convergence rate of $\mathbf{d_t}$ as a function of time **t:***

$$d_t \le h^t d_0 + (c_0 - \frac{\eta^2\sigma_1^2}{\gamma})(1-\gamma)^t(1-(\frac{p}{p+1})^t)$$
$$+\, \eta^2\sigma_1^2\frac{1-h^t}{\gamma},$$
$$where\ 0 < h = \frac{p(1-\gamma)}{p+1} < 1, k = \frac{1-\gamma}{p+1}, \gamma = 2\eta\frac{\mu L}{\mu+L}$$
$$and\ c_0 = \max_{i=1,\dots,p,f} \| w_0^i - w^* \|^2$$

- The average gap between workers and the optimum has a convergence rate of $O(h^t)$ , showing an exponentially shrinking gap when $t \to \infty$
- The convergence rate in the DP setting can easily be obtained by changing noise variance $\sigma$
- The performance trade-off for privacy is formulated in the last term, which remains stable when scale of workers $p$ grows

## Conclusion

- The Leader-Follower structure helps followers learn from better-performing nodes to improve convergence rate
- LEASGD ensures the same level accuracy while consuming less "privacy" - by strategically selecting specific nodes to interact with
- As a result, LEASGD shows an improvement in both the differential privacy and non-differentially private case