Dyn-D^{2}P: Dynamic Differentially Private Decentralized Learning with Provable Utility Guarantee

Zehan Zhu[1], **Yan Huang**[1], **Xin Wang**[2], **Shouling Ji**[1] and **Jinming Xu**[1]*
[1]Zhejiang University, Hangzhou, China
[2]Qilu University of Technology, Jinan, China
{12032045, huangyan5616}@zju.edu.cn, xinwang@qlu.edu.cn, {sji, jimmyxu}@zju.edu.cn

Abstract

Most existing decentralized learning methods with differential privacy (DP) guarantee rely on constant gradient clipping bounds and fixed-level DP Gaussian noises for each node throughout the training process, leading to a significant accuracy degradation compared to non-private counterparts. In this paper, we propose a new Dynamic Differentially Private Decentralized learning approach (termed Dyn-D^{2}P) tailored for general time-varying directed networks. Leveraging the Gaussian DP (GDP) framework for privacy accounting, Dyn-D^{2}P dynamically adjusts gradient clipping bounds and noise levels based on gradient convergence. This proposed dynamic noise strategy enables us to enhance model accuracy while preserving the total privacy budget. Extensive experiments on benchmark datasets demonstrate the superiority of Dyn-D^{2}P over its counterparts employing fixed-level noises, especially under strong privacy guarantees. Furthermore, we provide a provable utility bound for Dyn-D^{2}P that establishes an explicit dependency on network-related parameters, with a scaling factor of $1/\sqrt{n}$ in terms of the number of nodes n up to a bias error term induced by gradient clipping. To our knowledge, this is the first model utility analysis for differentially private decentralized non-convex optimization with dynamic gradient clipping bounds and noise levels.

1 Introduction

Distributed learning has recently attracted significant attention due to its great potential in enhancing computing efficiency and has thus been widely adopted in various application domains [Langer *et al.*, 2020]. In particular, it can be typically modeled as a non-convex finite-sum optimization problem solved by a group of n nodes as follows:

$$\min_{x\in\mathbb{R}^d} f(x) \triangleq \frac{1}{n}\sum_{i=1}^{n} f_i(x) \tag{1}$$

with $f_i(x) = \frac{1}{J}\sum_{j=1}^{J} f_i(x;j)$, where J denotes the local dataset size of each node, $f_i(x;j)$ denotes the loss function of the j-th data sample at node i with respect to the model parameter $x \in \mathbb{R}^d$, and $f_i(x)$ and $f(x)$ denote the local objective function at node i and the global objective function, respectively. All nodes collaborate to seek the optimal model parameter to minimize the global loss $f(x)$, and each node i can only evaluate local stochastic gradient $\nabla f_i(x;\xi_i)$ where $\xi_i \in \{1,2,...,J\}$ is a randomly chosen sample.

Bottlenecks such as high communication overhead and the vulnerability of central nodes in parameter server-based methods [Zinkevich *et al.*, 2010; McMahan *et al.*, 2017] motivate researchers to investigate fully decentralized methods [Lian *et al.*, 2017; Tang *et al.*, 2018] to solve Problem (1), where a central node is not required and each node only communicates with its neighbors. We thus consider such a fully decentralized setting in this paper, with a particular focus on general and practical time-varying directed communication networks for communication among nodes. Decentralized learning involves each node performing local stochastic gradient descent to update its model parameters, followed by communication with neighboring nodes to share and mix model parameters before proceeding to the next iteration [Zhu *et al.*, 2024]. However, the frequent information exchange among nodes poses significant privacy concerns, as the exposure of model parameters could potentially be exploited to compromise the privacy of original data samples [Wang *et al.*, 2019]. To protect each node from these potential attacks, differential privacy (DP), as a theoretical tool to provide rigorous privacy guarantees and quantify privacy loss, can be integrated into each node within decentralized learning to enhance privacy protection [Cheng *et al.*, 2018; Yu *et al.*, 2021].

Existing decentralized learning algorithms with differential privacy guarantee for non-convex problems tend to employ a constant/fixed gradient clipping bound $\bar{C}$ [Yu *et al.*, 2021; Xu *et al.*, 2022; Li and Chi, 2025] to estimate the l_2 sensitivity of gradient update and uniformly distribute privacy budgets across all iterations. As a result, each node injects fixed-level DP Gaussian noises with a variance proportional to the estimated sensitivity (i.e., constant clipping bound $\bar{C}$) before performing local SGD at each iteration. However, our empirical observations indicate that the norm of gradient typically decays as training progresses and ultimately converges

*Corresponding author.

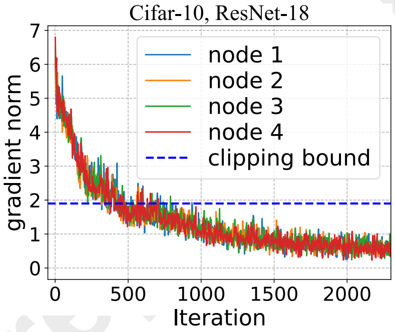


Figure 1: The evolution of the gradient norm when training ResNet-18 on Cifar-10 dataset in a fully decentralized setting with 4 nodes. It can be observed that the stochastic gradient norm at each node typically decays as training progresses, and eventually falls below the constant clipping bound (indicated by the blue dotted horizontal line) after certain number of iterations, which makes the clipping operation ineffective in the later stages of training.

to a small value (c.f., Figure 1). This observation suggests that using a constant clipping bound to estimate l_2 sensitivity throughout the training process may be overly conservative, as gradient norms are often smaller than the constant $\bar{C}$, especially in the later stages of training. Therefore, the added fixed-level Gaussian noise becomes unnecessary and instead degrades the model accuracy without providing additional privacy benefits. The following question thus arises naturally: *Can we design a decentralized learning method that dynamically adjusts the level of DP noise during the training process to minimize accuracy loss while maintaining privacy guarantee?*

To this end, we develop a new dynamic differentially private learning method for solving Problem (1) in fully decentralized settings, which enhances model accuracy while adhering to a total privacy budget constraint. The main contributions of this work[1] are threefold:

- We propose a differentially private decentralized learning method with a dynamic DP Gaussian noise strategy (termed Dyn-D^{2}P), tailored for general time-varying directed networks. In particular, each node adds noise with a variance calibrated by a dynamically decaying gradient clipping bound and an increasing per-step privacy budget appropriately allocated across iterations. This mechanism enables each node to apply dynamically decreasing noise, thereby enhancing model accuracy without compromising the total privacy budget.
- Theoretically, we investigate the impact of dynamic noise strategy on model utility for a general form of Dyn-D^{2}P where clipping bounds and noise levels can be arbitrary sequences (c.f., Theorem 1), revealing the advantages of using dynamically decaying clipping bounds (c.f., Remark 1). By employing exponentially decaying sequences, we prove the utility bound of Dyn-D^{2}P with explicit dependency on the network-related parameter, exhibiting a scaling factor of $1/\sqrt{n}$ in terms of the number of nodes up to a bias error term induced by gradient clipping (c.f., Corollary 1). To our knowledge, this is the first provable utility guarantee in the realm of dynamic differentially private decentralized learning.
- Extensive experiments are conducted to verify the performance of the proposed Dyn-D^{2}P. The results show that, under the same total privacy budget, Dyn-D^{2}P achieves superior accuracy compared to its counterpart using fixed-level DP Gaussian noise, especially under strong privacy guarantees. Moreover, we validate the robustness of Dyn-D^{2}P against certain hyper-parameters related to the varying rates of gradient clipping bound and per-step privacy budget, and verify the performance of Dyn-D^{2}P over different graphs and node numbers, which aligns with our theoretical findings.

2 Preliminary and Related Work

Differential privacy. Differential privacy was originally introduced in the seminal work by Dwork *et al.* [2006] as a foundational concept for quantifying the privacy-preserving capabilities of randomized algorithms, and has now found widespread applications in a variety of domains that necessitate safeguarding against unintended information leakage [Li *et al.*, 2020; Shin *et al.*, 2018; Wei *et al.*, 2021b]. We recall the standard definition of DP as follows.

Definition 1 ((ϵ, δ)-DP [Dwork *et al.*, 2014])**.** *A randomized mechanism $\mathcal{M}$ with domain $\mathcal{D}$ and range $\mathcal{R}$ satisfies (ϵ, δ)-differential privacy (or (ϵ, δ)-DP), if for any two adjacent inputs $\mathrm{x}, \mathrm{x}' \in \mathcal{D}$ differing on a single entry and for any subset of outputs $O \subseteq \mathcal{R}$, it holds that*

$$Pr\left[\mathcal{M}\left(\mathrm{x}\right) \in O\right] \leqslant e^{\epsilon} Pr\left[\mathcal{M}\left(\mathrm{x}'\right) \in O\right] + \delta, \tag{2}$$

where the privacy budget ϵ denotes the privacy lower bound to measure a randomized query and δ is the probability of breaking this bound. Note that a smaller value of ϵ implies a stronger privacy guarantee.

The following proposition provides the Gaussian DP (GDP) mechanism to ensure privacy guarantee.

Proposition 1 (μ-GDP [Dong *et al.*, 2022])**.** *Let $f : \mathcal{D} \to \mathbb{R}^d$ be a function and S be its l_2 sensitivity. Then, drawing a random noise N from Gaussian distribution with $N \sim \mathcal{N}\left(0, \sigma^2 \mathbb{I}_d\right)$ and adding it to f such that $\mathcal{M}(\mathrm{x}) = f(\mathrm{x}) + N$ satisfies μ-GDP if σ is set as*

$$\sigma = S/\mu, \tag{3}$$

where μ is the privacy budget in the GDP framework, and a smaller value of μ implies a stronger privacy guarantee.

The above proposition shows that the variance σ of the added noise required to ensure μ-GDP is dependent on both privacy budget μ and sensitivity S. It should be noted that the above privacy guarantee in the sense of GDP can be transformed into the standard DP by the following proposition which shows that there is a one-to-one correspondence between ϵ and μ values when fixing δ.

[1] Supplementary appendix can be found in our full version [Zhu *et al.*, 2025].

Proposition 2 (From μ-GDP to (ϵ, δ)-DP [Bu *et al.*, 2020])**.** *A random mechanism is μ-GDP if and only if it is (ϵ, δ)-DP for all $\epsilon \geqslant 0$, where*

$$\delta = \Phi\left(-\frac{\epsilon}{\mu} + \frac{\mu}{2}\right) - e^{\epsilon}\Phi\left(-\frac{\epsilon}{\mu} - \frac{\mu}{2}\right), \qquad (4)$$

and $\Phi(\cdot)$ is the Gaussian cumulative distribution function.

In what follows, we will review existing works related to achieving DP guarantees in machine learning and highlight the limitations in decentralized scenarios.

Decentralized learning methods with DP guarantee. DP guarantee is initially integrated into a centralized (single-node) setting for designing differentially private stochastic learning algorithms [Abadi *et al.*, 2016; Wang *et al.*, 2017; Iyengar *et al.*, 2019; Chen *et al.*, 2020; Wang *et al.*, 2020]. Further, DP guarantee is considered in distributed learning with server-client structures and the representative works include [McMahan *et al.*, 2018; Li *et al.*, 2019; Wang *et al.*, 2023; Wu *et al.*, 2020; Wei *et al.*, 2020; Zeng *et al.*, 2021; Wei *et al.*, 2021a; Li *et al.*, 2022; Liu *et al.*, 2022; Zhou *et al.*, 2023; Wei *et al.*, 2023]. Recently, there have been efforts to achieve DP guarantees for fully decentralized learning algorithms. For example, Cheng *et al.* [2018; 2019] achieve DP in fully decentralized learning for only strongly convex problems. Wang and Nedic [2024] achieve DP in fully decentralized architectures by tailoring gradient methods for deterministic optimization problems. For non-convex stochastic optimization problems as we consider in this work, Yu *et al.* [2021] present a differentially private decentralized learning method (DP^2-SGD) based on D-PSGD [Lian *et al.*, 2017], which relies on a fixed communication topology and uses the basic composition theorem to bound the overall privacy loss. To have a tight privacy guarantee, Xu *et al.* [2022] propose a differentially private asynchronous decentralized learning method (A(DP)^{2}SGD) based on AD-PSGD [Lian *et al.*, 2018], which provides privacy guarantee in the sense of Rényi differential privacy (RDP) [Mironov, 2017]. However, it should be noted that the aforementioned two algorithms [Yu *et al.*, 2021; Xu *et al.*, 2022] work only for undirected communication graphs which is often not satisfied in practical scenarios. Most recently, Li and Chi [2025] achieve DP guarantee as well as communication compression in decentralized learning for non-convex problems with the total privacy cost calculated via the moments accountant technique [Abadi *et al.*, 2016], while their methods are only applicable to time-invariant graphs.

Learning with dynamic DP Gaussian noise levels. For the aforementioned differentially private decentralized methods designed for non-convex stochastic optimization problems [Yu *et al.*, 2021; Xu *et al.*, 2022; Li and Chi, 2025], the injected fixed-level noise may exceed what is actually needed for privacy requirements as training progresses, especially during the later stages of training, since their estimated sensitivity based on constant/fixed gradient clipping bound $\bar{C}$ may not reflect the actual value of sensitivity (c.f., Figure 1). The overestimate of sensitivity may, indeed, lead to a waste of unnecessary privacy budget during the training process [Wei *et al.*, 2023]. Therefore, a tighter sensitivity estimate is useful for improving model accuracy without sacrificing privacy. There have been few works dedicated to tightly estimate the sensitivity in a dynamic manner. For instance, a scheme of decaying gradient clipping bounds has been employed to estimate the sensitivity in differentially private centralized learning [Du *et al.*, 2021; Wei and Liu, 2021], resulting in a decreasing amount of noise injection. In the realm of distributed learning, a similar strategy of dynamic clipping bounds is utilized in [Andrew *et al.*, 2021] to estimate sensitivity. Most recently, Wei *et al.* [2023] use the minimum of decaying clipping bounds and current gradient norms to estimate the l_2 sensitivity, leading to a less amount of noise injection. However, these distributed methods [Andrew *et al.*, 2021; Fu *et al.*, 2022; Wei *et al.*, 2023] only focus on server-client architecture and, most importantly, they do not provide any theoretical guarantee on model utility. In this paper, we aim to design a differentially private decentralized learning method that incorporates dynamic noise strategies in fully decentralized settings, and provide rigorous theoretical guarantee on model utility, as well as its utility-privacy trade-off.

3 Algorithm Development

In this section, we develop our differentially private decentralized learning methods using the Gaussian DP (GDP) framework as depicted in Proposition 1, which measures the privacy profile (ϵ, δ) in terms of μ according to Proposition 2. We consider solving Problem (1) over the following general peer-to-peer network model.

Network model. The communication topology is modeled as a sequence of time-varying directed graph $\mathcal{G}^k = \left(\mathcal{V}, \mathcal{E}^k\right)$, where $\mathcal{V} = \{1, 2, ..., n\}$ denotes the set of nodes and $\mathcal{E}^k \subset \mathcal{V} \times \mathcal{V}$ denotes the set of directed edges/links at iteration k. We associate each graph $\mathcal{G}^k$ with a non-negative mixing matrix $P^k \in \mathbb{R}^{n \times n}$ such that $(i, j) \in \mathcal{E}^k$ if $P_{i,j}^k > 0$, i.e., node i receiving a message from node j at iteration k. We assume that each node is an in-neighbor of itself.

The following assumptions are made on the mixing matrix and graph for the above network model to facilitate the subsequent utility analysis for our proposed algorithm.

Assumption 1 (Mixing matrix)**.** *The non-negative mixing matrix $P^k, \forall k$ is column-stochastic, i.e., $\mathbf{1}^\top P^k = \mathbf{1}^\top$, where $\mathbf{1}$ is an all-one vector.*

Assumption 2 (B-strongly connected)**.** *There exist positive integers B and $\triangle$ such that the graph formed by the edge set $\bigcup_{k=lB}^{(l+1)B-1} \mathcal{E}^k$ is strongly connected and has a diameter of at most $\triangle$ for $\forall l \geqslant 0$.*

Now, we present our differentially private decentralized learning algorithm (termed Dyn-D^{2}P) with a dynamic noise strategy, which works over the above general network model. The complete pseudocode is summarized in Algorithm 1. At a high level, Dyn-D^{2}P is comprised of local SGD and the averaging of neighboring information, following a framework similar to SGP [Assran *et al.*, 2019] which employs the Push-Sum protocol [Kempe *et al.*, 2003] to tackle the unblanceness of directed graphs. However, the key distinction lies in the gradient clipping operation and the injection of DP Gaussian

Algorithm 1 Dyn-D^{2}P

```
1: Initialization: DP budget $(\epsilon, \delta)$, $x_i^0 = z_i^0 \in \mathbb{R}^d$, $w_i^0 = 1$, step size $\gamma > 0$, total number of iterations $K$, initial clipping bound $C_0$ and hyper-parameters $\rho_c > 1$ and $\rho_\mu > 1$.
2: for $k = 0, 1, ..., K-1$, at node $i$, do
3:     Randomly samples a local training data $\xi_i^k$ with the sampling probability $\frac{1}{J}$;
4:     Computes stochastic gradient at $z_i^k$: $\nabla f_i(z_i^k; \xi_i^k)$;
5:     Calculates the clipping bound by: $C_k = C_0 \cdot (\rho_c)^{-\frac{k}{K}}$;
6:     Clips the stochastic gradient:
           $$\begin{aligned} g_i^k &= \operatorname{Clip}\left(\nabla f_i\left(z_i^k; \xi_i^k\right); C_k\right) \\ &= \nabla f_i\left(z_i^k; \xi_i^k\right) \cdot \min\left\{1, \frac{C_k}{\left\|\nabla f_i\left(z_i^k; \xi_i^k\right)\right\|}\right\}; \end{aligned} \tag{5}$$
7:     Calculates the per-step privacy budget by: $\mu_k = \mu_0 \cdot (\rho_\mu)^{\frac{k}{K}}$ with $\mu_0$ the solution of (8);
8:     Calculates the DP noise variance by:
           $$\sigma_k = \frac{C_k}{\mu_k} = \frac{C_0 \cdot (\rho_c)^{-\frac{k}{K}}}{\mu_0 \cdot (\rho_\mu)^{\frac{k}{K}}} = \frac{C_0}{\mu_0} \cdot (\rho_c \cdot \rho_\mu)^{-\frac{k}{K}};$$
9:     Draws randomized noise $N_i^k$ from the Gaussian distribution: $N_i^k \sim \mathcal{N}\left(0, \sigma_k^2 \mathbb{I}_d\right)$;
10:    Differentially private local SGD:
           $$x_i^{k+\frac{1}{2}} = x_i^k - \gamma(g_i^k + N_i^k);$$
11:    Sends $\left(x_i^{k+\frac{1}{2}}, w_i^k\right)$ to all out-neighbors and receives $\left(x_j^{k+\frac{1}{2}}, w_j^k\right)$ from all in-neighbors ;
12:    Updates $x_i^{k+1}$ by: $x_i^{k+1} = \sum_{j=1}^n P_{i,j}^k x_j^{k+\frac{1}{2}}$ ;
13:    Updates $w_i^{k+1}$ by: $w_i^{k+1} = \sum_{j=1}^n P_{i,j}^k w_j^k$;
14:    Updates $z_i^{k+1}$ by: $z_i^{k+1} = x_i^{k+1}/w_i^{k+1}$.
15: end for
```

noise before performing local SGD. In particular, each node i maintains three variables during the learning process: i) the model parameter x_i^k; ii) the scalar Push-Sum weight w_i^k and iii) the de-biased parameter $z_i^k = x_i^k / w_i^k$, with the initialization of $x_i^0 = z_i^0 \in \mathbb{R}^d$ and $w_i^0 = 1$ for all nodes $i \in \mathcal{V}$. At each iteration k, each node i updates as follows:

$$\texttt{Private local SGD: } x_i^{k+\frac{1}{2}} = x_i^k - \gamma\left(g_i^k + N_i^k\right),$$

$$\texttt{Averaging: } x_i^{k+1} = \sum_{j=1}^n P_{i,j}^k x_j^{k+\frac{1}{2}}, w_i^{k+1} = \sum_{j=1}^n P_{i,j}^k w_j^k,$$

$$\texttt{De-bias: } z_i^{k+1} = x_i^{k+1}/w_i^{k+1},$$

where $\gamma > 0$ is the step size and g_i^k is the clipped gradient (c.f., (5)). N_i^k denotes the injected random noise to ensure μ_k-GDP guarantee for node i at iteration k. This noise is drawn from a Gaussian distribution with variance σ_k^2, calibrated according to the dynamic clipping bound C_k and per-step privacy budget μ_k. We note that the two key mechanisms in the proposed Dyn-D^{2}P to achieve dynamic noise levels and enhance model accuracy include: i) dynamically decreasing clipping bounds (c.f., line 5 in Algorithm 1); ii) dynamically increasing per-step privacy budget (c.f., line 7 in Algorithm 1). The detailed design and motivation for these mechanisms will be explained as follows.

Dynamic decreasing clipping bounds. For differentially private decentralized learning algorithms, the gradient clipping operation is necessary for each node to bound the l_2 sensitivity of local SGD update and inject noise accordingly calibrated with l_2 sensitivity (i.e., clipping bound) and privacy budget. According to the previous discussion in Section 1 that the norm of stochastic gradient of each node typically decreases as training proceeds, we know that the stochastic gradient would not be clipped after some iteration k if we use the constant clipping bound $\bar{C}$ as Yu *et al.* [2021]; Xu *et al.* [2022]; Li and Chi [2025] did, thus resulting in adding unnecessary excessive noise calibrated with $\bar{C}$ in the later stage of training. To address this issue, we employ a dynamic gradient clipping scheme for each node to reduce the clipping bounds C_k across the updates. Compared to the fixed clipping bound $\bar{C}$ used in [Yu *et al.*, 2021; Xu *et al.*, 2022; Li and Chi, 2025], it can reduce the noise level after a particular k when $C_k < \bar{C}$, which is beneficial for stabilizing the updates. In particular, we set the clipping bound C_k as $C_k = C_0 \cdot (\rho_c)^{-\frac{k}{K}}$, where $\rho_c > 1$ is the hyper-parameter to control the decay rate of the clipping bound, C_0 is the initial clipping bound, and K is the total number of iterations.

Dynamic increasing per-step privacy budget. Given a total privacy budget, the existing differentially private decentralized learning methods [Yu *et al.*, 2021; Xu *et al.*, 2022; Li and Chi, 2025] uniformly distribute privacy budgets across all training steps. However, recent works [Zhu *et al.*, 2019; Wei and Liu, 2021; Wei *et al.*, 2023] point out that it is relatively easier to leak privacy at the initial stage of the training process, and it becomes increasingly difficult as the training progresses. To this end, we allocate a small privacy budget in the early stages and gradually increase privacy budgets, i.e., setting $\mu_{k+1} \geqslant \mu_k$ for all $k \in \{0, 1, 2, ..., K-1\}$. In addition, according to (3), we observe that small (resp. large) μ_k means adding large (resp. small) noise. Therefore, setting $\mu_{k+1} \geqslant \mu_k$ implies adding large (resp., small) noise in the early (resp., later) stages of decentralized training, which helps improve model accuracy. The intuition is that at the beginning of training, the model is far away from the optimum, and the gradient magnitudes are usually large (c.f., Figure 1); larger noise is thus allowed and even helps to quickly escape the saddle point [Ge *et al.*, 2015]. As training proceeds, the model approaches the optimum and the gradient magnitude converge, smaller noise is then desired to stabilize the update for convergence.

The following proposition provides a way for privacy accounting of dynamic non-uniform μ_k-GDP costs throughout the whole training process.

Proposition 3 (Composition theorem for GDP with varying μ_k [Du *et al.*, 2021])**.** *Consider a series of random mechanisms M_k for $k \in \{0,1,2,...,K-1\}$, where M_k is μ_k-GDP, and each mechanism works only on a subsampled dataset by independent Bernoulli trial with probability p. After K steps by composition of Gaussian mechanism, $M \triangleq M_{K-1} \otimes \cdots \otimes M_1 \otimes M_0$ satisfies μ_{tot}-GDP where*

$$\mu_{tot} = p \cdot \sqrt{\sum_{k=0}^{K-1} \left(e^{\mu_k^2} - 1\right)}. \tag{6}$$

To this end, we set the per-step privacy budget μ_k as

$$\mu_k = \mu_0 \cdot (\rho_\mu)^{\frac{k}{K}}, \tag{7}$$

where $\rho_\mu > 1$ is the hyper-parameter controlling the growth rate of μ_k, and μ_0 is the initial privacy budget. Given the target total privacy budget (ϵ, δ), the corresponding privacy budget in the GDP framework μ_{tot} can be obtained by (4). Substituting (7) into (6) with $p = \frac{1}{J}$, we have

$$J^2 \mu_{tot}^2 = \sum_{k=0}^{K-1} \left(e^{\left(\mu_0 \cdot (\rho_\mu)^{\frac{k}{K}}\right)^2} - 1 \right), \tag{8}$$

and μ_0 can be computed using a numerical method such as binary search. With a specific value of μ_0 and ρ_μ, the value of μ_k at each iteration k can be calculated by (7).

In addition, we provide two by-product differentially private decentralized learning algorithms (termed Dyn[C]-D^{2}P and Dyn[μ]-D^{2}P), which employ only dynamic clipping bound reduction method and dynamic per-step privacy budget growth method, respectively:

- Dyn[C]-D^{2}P: we set the clipping bound C_k as $C_k = C_0 \cdot (\rho_c)^{-\frac{k}{K}}$ (c.f., Algorithm 1), while maintaining the per-step privacy budget μ_k the same for all iterations, i.e., fixing $\mu_k = \bar{\mu}$, $\forall k$. By substituting $\mu_k = \bar{\mu}$ into (6) in Proposition 3 with $p = \frac{1}{J}$, we obtain the closed-form solution of $\bar{\mu}$:

  $$\bar{\mu} = \sqrt{\log\left(\frac{J^2 \mu_{tot}^2}{K} + 1\right)}. \tag{9}$$

  With specific values of C_k and $\bar{\mu}$, we can calibrate the DP noise variance at each iteration by $\sigma_k = \frac{C_k}{\bar{\mu}} = \frac{C_0}{\bar{\mu}} \cdot (\rho_c)^{-\frac{k}{K}}$. The pseudocode of Dyn[$C$]-D^{2}P can be found in Appendix D in our full version [Zhu *et al.*, 2025].
- Dyn[μ]-D^{2}P: we set the per-step privacy budget μ_k as $\mu_k = \mu_0 \cdot (\rho_\mu)^{\frac{k}{K}}$ (c.f., Algorithm 1), where μ_0 is the solution of (8), while employing the fixed clipping bound, i.e., $C_k = \bar{C}$, $\forall k$. Therefore, the DP noise variance σ_k at each iteration can be calculated by $\sigma_k = \frac{\bar{C}}{\mu_k} = \frac{\bar{C}}{\mu_0} \cdot (\rho_\mu)^{-\frac{k}{K}}$. The pseudocode of Dyn[$\mu$]-D^{2}P can be found in Appendix D in our full version [Zhu *et al.*, 2025].

It can be observed that, given ρ_c and ρ_μ, the noise decay rate of Dyn-D^{2}P is faster than that of Dyn[C]-D^{2}P and Dyn[μ]-D^{2}P. We also present an algorithm termed Const-D^{2}P as our baseline, which employs constant clipping bound (i.e., fixing $C_k = \bar{C}$, $\forall k$) and uniformly distributes privacy budgets across updates (i.e., $\mu_k = \bar{\mu}$, $\forall k$) as usually did in most of existing DP-based decentralized methods [Yu *et al.*, 2021; Xu *et al.*, 2022; Li and Chi, 2025]. According to the values of $\bar{C}$ and $\bar{\mu}$ (c.f., (9)), the constant DP noise variance is $\bar{\sigma} = \frac{\bar{C}}{\bar{\mu}}$. The pseudocode of Const-D^{2}P can be found in Appendix D in our full version [Zhu *et al.*, 2025].

Algorithm 2 General Form of Dyn-D^{2}P

```
Initialization: DP budget $(\epsilon, \delta)$, $x_i^0 = z_i^0 \in \mathbb{R}^d$, $w_i^0 = 1$, step size $\gamma > 0$, total number of iterations $K$, clipping bounds $C_0, ..., C_{K-1}$, and noise levels $\tilde{\sigma} \cdot \tilde{\sigma}_0, ...., \tilde{\sigma} \cdot \tilde{\sigma}_{K-1}$.
for $k = 0, 1, ..., K-1$, at node $i$, do
    Follows from line 3 and 4 in Algorithm 1;
    Gradient clipping: $g_i^k = \mathrm{Clip}\left(\nabla f_i\left(z_i^k; \xi_i^k\right); C_k\right)$;
    Draws randomized noise $N_i^k$ from the Gaussian distribution: $N_i^k \sim \mathcal{N}\left(0, \tilde{\sigma}^2 \mathbb{I}_d\right)$;
    Differentially private local SGD:
        $x_i^{k+\frac{1}{2}} = x_i^k - \gamma(g_i^k + \tilde{\sigma}_k N_i^k)$;
    Follows from line 11-14 in Algorithm 1.
end for
```

4 Theoretical Analysis

In this section, we theoretically investigate the impact of our dynamic noise strategy on model utility guarantee and provide the utility bound for proposed Dyn-D^{2}P given a certain privacy budget. We first present a general form of Dyn-D^{2}P where C_k and σ_k can be arbitrarily predefined sequences (c.f., Algorithm 2). Different from Algorithm 1, we denote $\sigma_k = \tilde{\sigma} \cdot \tilde{\sigma}_k$ for convenience in subsequent analysis, and we term $\tilde{\sigma}$ as noise scale without loss of generality.

Then, we show that the DP guarantee for each node in Algorithm 2 can be achieved by setting the noise scale $\tilde{\sigma}$ properly according to the given total privacy budget (ϵ, δ) as well as the equivalent privacy parameter μ_{tot}, which is summarized in the following proposition.

Proposition 4 (Privacy guarantee)**.** *Let K be the total number of iterations. Algorithm 2 achieves (ϵ, δ)-DP guarantee for each node if we set the noise scale as*

$$\tilde{\sigma} = \frac{1}{J \mu_{tot}} \sqrt{2 \sum_{k=0}^{K-1} \frac{C_k^2}{\tilde{\sigma}_k^2}}, \tag{10}$$

where μ_{tot} is the solution of (4) *with $\mu = \mu_{tot}$.*

Proof. Refer to Appendix C in our full version [Zhu *et al.*, 2025]. □

Next, we make the following commonly used assumption for the utility analysis of Algorithm 2.

Assumption 3 (L-smoothness)**.** *For each function $f_i, i \in \mathcal{V}$, there exists a constant $L > 0$ such that $\left\|\nabla f_i(x) - \nabla f_i(y)\right\| \leqslant L\left\|x-y\right\|$.*

Suppose Assumptions 1-3 hold, and assume that each per-sample gradient is upper-bounded, i.e., $\left\|\nabla f_i(z;\xi_i)\right\| \leqslant \varLambda$. Then, we are ready to provide the following theorem to characterize the utility guarantee of Algorithm 2.

Theorem 1 (Model utility)**.** *If we set the noise scale $\tilde{\sigma}$ as in* (10)*, Algorithm 2 can achieve (ϵ,δ)-DP guarantee for each node and has the following utility guarantee*

$$\begin{aligned}
&\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[\left\|\nabla f\left(\bar{x}^k\right)\right\|^2\right]\\
\leqslant&\frac{2\left(f\left(\bar{x}^0\right)-f^*\right)}{\gamma K}+\frac{3L^2\Psi^2\sum_{i=1}^n\left\|x_i^0\right\|^2}{(1-q)^2nK}\\
&+\frac{3\gamma^2L^2\Psi^2}{(1-q)^2}\cdot\frac{1}{K}\sum_{k=0}^{K-1}\cdot\frac{1}{n}\sum_{i=1}^n\mathbb{E}\left[\left\|g_i^k\right\|^2\right]\\
&+\gamma L\cdot\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[\left\|\frac{1}{n}\sum_{i=1}^n g_i^k\right\|^2\right]\\
&+\left(\frac{6\gamma^2KL^2\Psi^2d}{(1-q)^2J^2\mu_{tot}^2}+\frac{2\gamma KLd}{nJ^2\mu_{tot}^2}\right)\underbrace{\frac{1}{K}\sum_{k=0}^{K-1}\frac{C_k^2}{\tilde{\sigma}_k^2}\frac{1}{K}\sum_{k=0}^{K-1}\tilde{\sigma}_k^2}_{\triangleq T_1\text{: privacy noise term}},\\
&\underbrace{+2\mathbb{E}\left[\frac{1}{K}\sum_{k=0}^{K-1}\varLambda\left\|\nabla f\left(\bar{x}^k\right)\right\|\cdot\frac{1}{n}\sum_{i=1}^n\mathcal{P}_i^k\left(C_k\right)\right]}_{\triangleq T_2\text{: bias term}}
\end{aligned}\tag{11}$$

where $\bar{x}^k = \frac{1}{n}\sum_{i=1}^n x_i^k$, $f^* = \min_{x\in\mathbb{R}^d} f(x)$ while Ψ and $q \in [0,1)$ are positive constants[2] *depending on the diameter of the network $\triangle$ and the sequence of mixing matrices P^k, whose definition can be found in Lemma 2 in the appendix in our full version [Zhu* et al.*, 2025], $\mathcal{P}_i^k(C_k)$ denotes the probability of a stochastic gradient being clipped with clipping bound C_k for node i at iteration k.*

Proof. Refer to Appendix A in our full version [Zhu *et al.*, 2025]. □

Remark 1 (On clipping bounds C_k)**.** *The clipping of gradients will destroy the unbiased estimate of the local full gradient for each node, resulting in a constant bias error (reflected by T_2 in* (11)*). It is obvious that the smaller the clipping bound C_k, the greater the probability $\mathcal{P}_i^k(C_k)$ that clipping occurs at iteration k. Based on this fact, we know that in general, the bias term T_2 in* (11) *will be small if C_k's are large and vise versa, given the same distribution of gradients. This implies that we can use large clipping bounds to reduce the bias term T_2. However, this may make the privacy noise error term T_1 in* (11) *become larger. It is thus essential to choose a proper sequence of C_k that could effectively balance T_1 and T_2. However, finding an optimal sequence of clipping bounds to achieve the best trade-off necessitates knowledge of the distribution of the stochastic gradient, which is impossible to obtain in practice. The existing literature [Andrew* et al.*, 2021] points out that a noteworthy practical way is to keep the probability $\mathcal{P}_i^k\left(C_i^k\right)$ approximately constant. Considering that the norm of the gradient is decreasing as training progresses, C_k should also be decreasing so as to keep $\mathcal{P}_i^k(C_k)$ roughly constant, which supports the design of our algorithm (c.f., Algorithm 1).*

Next, we specify the sequences C_k and $\tilde{\sigma}_k$ in Algorithm 2, yielding a special instance of dynamic clipping bound and noise scheduling as used in Algorithm 1. We then provide the corresponding utility bound in the following corollary.

Corollary 1. *Under the same conditions of Theorem 1, by setting $C_k = \Theta\left((\rho_c)^{-\frac{k}{K}}\right)$ and $\tilde{\sigma}_k = \Theta\left((\rho_c\cdot\rho_\mu)^{-\frac{k}{K}}\right)$ for some $\rho_c > 1$ and $\rho_\mu > 1$, if we set $\gamma = \frac{1}{\sqrt{n}J\mu_{tot}}$, $\gamma K = \sqrt{n}J\mu_{tot}$ and assume that $J\mu_{tot} > \sqrt{n}$, we have*

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[\left\|\nabla f\left(\bar{x}^k\right)\right\|^2\right]\leqslant\mathcal{O}\left(\frac{1}{(1-q)^2\sqrt{n}J\mu_{tot}}\right)+T_2,\tag{12}$$

where $\mathcal{O}(\cdot)$ hides some constants, e.g., L, $\sum_{i=1}^n\left\|x_i^0\right\|^2$, $\varLambda$.

Proof. Refer to Appendix B in our full version [Zhu *et al.*, 2025]. □

Remark 2 (Tight utility bound)**.** *Setting aside the non-vanishing bias term T_2, the utility bound derived in Corollary 1 exhibits explicit dependency on the network-related parameter q, and reveals a scaling factor of $1/\sqrt{n}$ with respect to the number of nodes n, which has never been observed in the previous differentially private decentralized algorithms [Yu* et al.*, 2021; Xu* et al.*, 2022]. By setting $n = 1$ which corresponds to the centralized/single-node case, our utility result* (12) *can be reduced to $\mathcal{O}(1/J\mu_{tot})$, sharing the same polynomial order as existing analysis [Zhang* et al.*, 2017; Du* et al.*, 2021]. To our best knowledge, we provide the first theoretical utility analysis for decentralized non-convex stochastic optimization with dynamic gradient clipping bounds and noise levels, highlighting the utility-privacy trade-off.*

Remark 3 (Non-trivial analysis)**.** *The exponential decay scheme of the sequences C_k and $\tilde{\sigma}_k$ aligns with that of the sequences C_k and σ_k in Algorithm 1, up to a constant factor. Therefore, the theoretical result (utility bound) in Corollary 1 corroborates the parameter settings in Algorithm 1. We note that our analysis is non-trivial and can not be directly derived by extending the existing analysis [Du* et al.*, 2021] for the centralized/single-node case. The authors in [Du* et al.*, 2021] assume C_k and $\tilde{\sigma}_k$ to be constant when deriving utility bounds, thereby simplifying the analysis (c.f., Theorem 1 in [Du* et al.*, 2021]). In contrast, our analysis involves a more precise setting of C_k and $\tilde{\sigma}_k$ as outlined in Corollary 1, which requires a more sophisticated analysis (c.f., Eq.*(35) *to* (36) *in the appendix in our full version [Zhu* et al.*, 2025] for the derivation). We will also evaluate the impact of two*

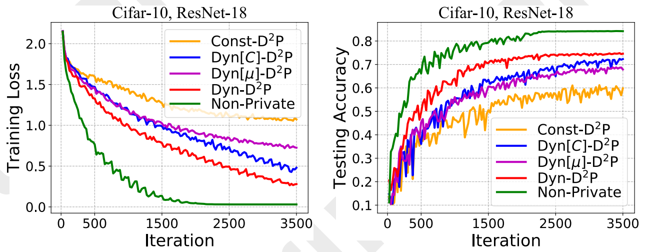
[2] q characterizes the speed of information propagation over the network. A smaller value of q indicates faster propagation.

Algorithm	$\epsilon = 0.3$	$\epsilon = 0.7$	$\epsilon = 1$	$\epsilon = 3$
Non-Private	82.16			
Const-D^{2}P	46.37	52.38	60.02	71.11
Dyn[μ]-D^{2}P	64.12	66.21	69.37	76.3
Dyn[C]-D^{2}P	66.93	68.37	72.02	77.65
Dyn-D^{2}P	**71.87**	**72.14**	**75.06**	**79.89**

Table 1: Final testing accuracy (%) for different algorithms when training ResNet-18 on Cifar-10 dataset, under different values of privacy budget ϵ.

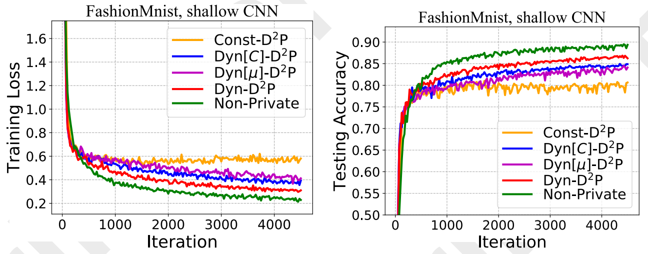
Algorithm	$\epsilon = 0.3$	$\epsilon = 0.7$	$\epsilon = 1$	$\epsilon = 3$
Non-Private	89.98			
Const-D^{2}P	45.37	58.63	74.65	80.81
Dyn[μ]-D^{2}P	81.12	82.98	82.23	84.36
Dyn[C]-D^{2}P	82.93	83.65	84.06	84.98
Dyn-D^{2}P	**84.88**	**85.36**	**86.21**	**87.89**

Table 2: Final testing accuracy (%) for different algorithms when training shallow CNN on FashionMnist dataset, under different values of privacy budget ϵ.



(a) Training loss (b) Testing accuracy

Figure 2: Comparison of convergence performance for five algorithms under $(1, 10^{-4})$-DP guarantee for each node, when training ResNet-18 on Cifar-10 dataset.



(a) Training loss (b) Testing accuracy

Figure 3: Comparison of convergence performance for five algorithms under $(3, 10^{-4})$-DP guarantee for each node, when training shallow CNN on FashionMnist dataset.

hyper-parameters ρ_μ and ρ_c on algorithm performance in the experimental section. The fixed schedule for the clipping bound and noise level in Corollary 1 is not the only option for our general algorithm form (Algorithm 2). Exploring alternative dynamic clipping bound mechanisms–such as using more complex schedules than exponential decaying method or adaptive clipping bound calculated based on the real-time gradient norm–would require additional heuristic efforts and complicate the theoretical utility analysis. We leave these explorations for future work.

5 Experiments

We conduct extensive experiments to verify the performance of proposed Dyn-D^{2}P (c.f., Algorithm 1), with comparison to our two by-product algorithms Dyn[C]-D^{2}P and Dyn[μ]-D^{2}P, and two baselines: i) Const-D^{2}P, which employs fixed-level noise; ii) non-private decentralized learning algorithm, which does not use gradient clipping or add DP noise, and thus serves as the upper bound of model accuracy. All experiments are deployed in a server with Intel Xeon E5-2680 v4 CPU @ 2.40GHz and 8 Nvidia RTX 3090 GPUs, and are implemented with distributed communication package *torch.distributed* in PyTorch [Paszke *et al.*, 2017], where a process serves as a node, and inter-process communication is used to mimic communication among nodes.

5.1 Experimental Setup

We compare five algorithms in a fully decentralized setting composed of 20 nodes, on two benchmark non-convex learning tasks: i) training ResNet-18 [He *et al.*, 2016] on Cifar-10 [Krizhevsky, 2009] dataset; ii) training shallow CNN model (composed of two convolution layers and two fully connected layers) on FashionMnist [Xiao *et al.*, 2017] dataset. We split shuffled datasets evenly to 20 nodes. For communication topology, unless otherwise stated, we use a time-varying directed exponential graph (refer to Appendix E in our full version [Zhu *et al.*, 2025] for its definition). The learning rate is set to be 0.05 for ResNet-18 training and 0.03 for shallow CNN model training. Privacy parameters δ is set to be 10^{-4}, and we test different values for ϵ which implies different levels of privacy guarantee. Other parameters such as $\bar{C}$, C_0, ρ_c and ρ_μ are detailed in Appendix F.1 in our full version [Zhu *et al.*, 2025]. Note that all experimental results are averaged over five repeated runs.

5.2 Superior Performance against Baseline Methods

For the ResNet-18 training task, we present the experimental result of final model accuracy in Table 1, and provide the plots of training loss/testing accuracy versus iteration in Figure 2 with a privacy budget of $\epsilon = 1$ (the plots with other values of ϵ can be found in Appendix G in our full version [Zhu *et al.*, 2025]). It can be observed that our Dyn-D^{2}P and two by-product algorithms (Dyn[C]-D^{2}P and Dyn[μ]-D^{2}P) consistently outperform the baseline algorithm Const-D^{2}P which employs constant noise. Among these above DP algorithms, Dyn-D^{2}P achieves the highest model accuracy while maintaining the same level of privacy protection. Furthermore, a comparison of experimental results with different privacy budgets (i.e., varying values of ϵ) shows that the stronger the level of required privacy protection (i.e., the smaller the value of budget ϵ), the more pronounced the advantage in model accuracy with our dynamic noise strategy. In particular, when setting a small $\epsilon = 0.3$ which implies a strong privacy guarantee, our Dyn-D^{2}P achieves a 25% higher model accuracy than Const-D^{2}P employing constant noise strategies. These results verify the superiority of our dynamic noise approach.

For the shallow CNN training task, we present the experimental result of final model accuracy in Table 2, and provide the plots of training loss/testing accuracy in terms of iteration in Figure 3 with a privacy budget of $\epsilon = 1$ (the plots with other values of ϵ can be found in Appendix G in our full version [Zhu *et al.*, 2025]). The takeaways from the experimental results are similar to previous experiments on the ResNet-18 training task, and our proposed Dyn-D^{2}P performs much better than the baseline algorithm Const-D^{2}P while maintaining the same level of privacy protection, which again highlights the superiority of our dynamic noise approach. In particular, under a strong level of required privacy guarantee with budget $\epsilon = 0.3$, our Dyn-D^{2}P achieves a 39% higher model accuracy compared to Const-D^{2}P.

5.3 Sensitivity to Hyper-parameters

In this part, we test the robustness of our algorithm (c.f., Algorithm 1) to different values of hyper-parameters ρ_c and ρ_μ. We use grid search to demonstrate the impact of these two parameters on the final model accuracy. It follows from the results as shown in Figure 4 that, our algorithm can almost maintain the final model accuracy and consistently improve model accuracy compared to Const-D^{2}P across a wide range of ρ_c and ρ_μ, which implies that our proposed algorithm is robust to the value of hyper-parameter ρ_c and ρ_μ.

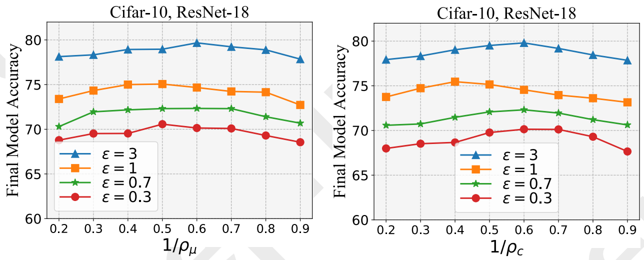
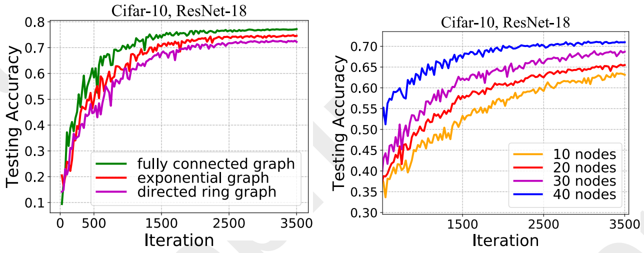


Figure 4: Illustration of the robustness of Dyn-D^{2}P against the values of ρ_μ and ρ_c under different privacy budget ϵ.



(a) $(1, 10^{-4})$-DP guarantee (b) $(1, 10^{-4})$-DP guarantee

Figure 5: Comparison of convergence performance (model utility) for Dyn-D^{2}P over (a) different graphs consisting of 20 nodes and (b) an exponential graph with different number of nodes, respectively.

5.4 Performance over Different Graphs and Node Numbers

First, we implement Dyn-D^{2}P over a static directed ring graph, a time-varying directed exponential graph, and a fully connected graph, where the q value w.r.t. these three graphs decreases in sequence. The experimental result shown in Figure 5(a) demonstrates that the model accuracy (utility) increases in sequence across these graphs. Then, we implement Dyn-D^{2}P on a time-varying directed exponential graph consisting of different numbers of nodes (refer to Appendix F.2 in our full version [Zhu *et al.*, 2025] for more details on the setup). It can be observed from Figure 5(b) that, increasing the number of nodes improves the model accuracy (utility). Note that both of these above observations align with the theoretical insights outlined in Corollary 1 and Remark 2.

6 Conclusion

In this work, we proposed a differentially private decentralized learning method Dyn-D^{2}P, for non-convex optimization problems, which dynamically adjusts gradient clipping bounds and noise levels across the update. The proposed dynamic noise strategy allows us to enhance the model accuracy while maintaining the level of privacy guarantee. Extensive experiments show that our Dyn-D^{2}P outperforms the existing counterparts with fixed-level noises, especially under strong privacy levels. Our analysis shows that the utility bound of Dyn-D^{2}P exhibits an explicit dependency on the network-related parameter and enjoys a scaling factor of $1/\sqrt{n}$, up to a bias error term induced by gradient clipping. To our knowledge, we provide the first theoretical utility analysis for fully decentralized non-convex stochastic optimization with dynamic gradient clipping bounds and noise levels, highlighting the utility-privacy trade-off. We will focus on eliminating the bias term induced by gradient clipping in the future.

Acknowledgments

This work is supported in parts by the National Key R&D Program of China under Grant No. 2022YFB3102100, and in parts by National Natural Science Foundation of China under Grants 62373323, 62088101, U244120033, 62402256.

References

[Abadi *et al.*, 2016] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[Andrew *et al.*, 2021] Galen Andrew, Om Thakkar, Brendan McMahan, and Swaroop Ramaswamy. Differentially private learning with adaptive clipping. *Advances in Neural Information Processing Systems*, 34:17455–17466, 2021.

[Assran *et al.*, 2019] Mahmoud Assran, Nicolas Loizou, Nicolas Ballas, and Mike Rabbat. Stochastic gradient push for distributed deep learning. In *International Conference on Machine Learning*, pages 344–353. PMLR, 2019.

[Bu *et al.*, 2020] Zhiqi Bu, Jinshuo Dong, Qi Long, and Weijie J Su. Deep learning with gaussian differential privacy. *arXiv preprint arXiv:1911.11607*, 2020.

[Chen *et al.*, 2020] Xiangyi Chen, Steven Z Wu, and Mingyi Hong. Understanding gradient clipping in private sgd: A geometric perspective. *Advances in Neural Information Processing Systems*, 33:13773–13782, 2020.

[Cheng *et al.*, 2018] Hsin-Pai Cheng, Patrick Yu, Haojing Hu, Feng Yan, Shiyu Li, Hai Li, and Yiran Chen. LEASGD: an efficient and privacy-preserving decentralized algorithm for distributed learning. *arXiv preprint arXiv:1811.11124*, 2018.

[Cheng *et al.*, 2019] Hsin-Pai Cheng, Patrick Yu, Haojing Hu, Syed Zawad, Feng Yan, Shiyu Li, Hai Li, and Yiran Chen. Towards decentralized deep learning with differential privacy. In *International Conference on Cloud Computing*, pages 130–145. Springer, 2019.

[Dong *et al.*, 2022] Jinshuo Dong, Aaron Roth, and Weijie J Su. Gaussian differential privacy. *Journal of the Royal Statistical Society Series B*, 84(1):3–37, 2022.

[Du *et al.*, 2021] Jian Du, Song Li, Xiangyi Chen, Siheng Chen, and Mingyi Hong. Dynamic differential-privacy preserving sgd. In *International Conference on Machine Learning*. PMLR, 2021.

[Dwork *et al.*, 2006] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Annual international conference on the theory and applications of cryptographic techniques*, pages 486–503. Springer, 2006.

[Dwork *et al.*, 2014] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[Fu *et al.*, 2022] Jie Fu, Zhili Chen, and Xiao Han. Adap DP-FL: Differentially private federated learning with adaptive noise. In *2022 IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 656–663. IEEE, 2022.

[Ge *et al.*, 2015] Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points—online stochastic gradient for tensor decomposition. In *Conference on Learning Theory*, pages 797–842. PMLR, 2015.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[Iyengar *et al.*, 2019] Roger Iyengar, Joseph P Near, Dawn Song, Om Thakkar, Abhradeep Thakurta, and Lun Wang. Towards practical differentially private convex optimization. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 299–316. IEEE, 2019.

[Kempe *et al.*, 2003] David Kempe, Alin Dobra, and Johannes Gehrke. Gossip-based computation of aggregate information. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, pages 482–491. IEEE, 2003.

[Krizhevsky, 2009] Alex Krizhevsky. Learning multiple layers of features from tiny images. *Master's thesis, University of Toronto*, 2009.

[Langer *et al.*, 2020] Matthias Langer, Zhen He, Wenny Rahayu, and Yanbo Xue. Distributed training of deep learning models: A taxonomic perspective. *IEEE Transactions on Parallel and Distributed Systems*, 31(12):2802–2818, 2020.

[Li and Chi, 2025] Boyue Li and Yuejie Chi. Convergence and privacy of decentralized nonconvex optimization with gradient clipping and communication compression. *IEEE Journal of Selected Topics in Signal Processing*, 2025.

[Li *et al.*, 2019] Yanan Li, Shusen Yang, Xuebin Ren, and Cong Zhao. Asynchronous federated learning with differential privacy for edge intelligence. *arXiv preprint arXiv:1912.07902*, 2019.

[Li *et al.*, 2020] Jeffrey Li, Mikhail Khodak, Sebastian Caldas, and Ameet Talwalkar. Differentially private meta-learning. In *8th International Conference on Learning Representations, ICLR 2020*, 2020.

[Li *et al.*, 2022] Zhize Li, Haoyu Zhao, Boyue Li, and Yuejie Chi. SoteriaFL: A unified framework for private federated learning with communication compression. *Advances in Neural Information Processing Systems*, 35:4285–4300, 2022.

[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.

[Liu *et al.*, 2022] Tianyu Liu, Boya Di, Bin Wang, and Lingyang Song. Loss-privacy tradeoff in federated edge learning. *IEEE Journal of Selected Topics in Signal Processing*, 16(3):546–558, 2022.

[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.

[McMahan *et al.*, 2018] H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. In *International Conference on Learning Representations*, 2018.

[Mironov, 2017] Ilya Mironov. Rényi differential privacy. In *2017 IEEE 30th computer security foundations symposium (CSF)*, pages 263–275. IEEE, 2017.

[Paszke *et al.*, 2017] Adam Paszke, Sam Gross, Soumith Chintala, and Gregory Chanan. Pytorch: Tensors and dynamic neural networks in python with strong gpu acceleration. *PyTorch: Tensors and dynamic neural networks in Python with strong GPU acceleration*, 6(3):67, 2017.

[Shin *et al.*, 2018] Hyejin Shin, Sungwook Kim, Junbum Shin, and Xiaokui Xiao. Privacy enhanced matrix factorization for recommendation with local differential privacy. *IEEE Transactions on Knowledge and Data Engineering*, 30(9):1770–1782, 2018.

[Tang *et al.*, 2018] Hanlin Tang, Xiangru Lian, Ming Yan, Ce Zhang, and Ji Liu. D^2: Decentralized training over decentralized data. In *International Conference on Machine Learning*, pages 4848–4856. PMLR, 2018.

[Wang and Nedic, 2024] Yongqiang Wang and Angelia Nedic. Tailoring gradient methods for differentially private distributed optimization. *IEEE Transactions on Automatic Control*, 69(2):872–887, 2024.

[Wang *et al.*, 2017] Di Wang, Minwei Ye, and Jinhui Xu. Differentially private empirical risk minimization revisited: Faster and more general. *Advances in Neural Information Processing Systems*, 30, 2017.

[Wang *et al.*, 2019] Zhibo Wang, Mengkai Song, Zhifei Zhang, Yang Song, Qian Wang, and Hairong Qi. Beyond inferring class representatives: User-level privacy leakage from federated learning. In *IEEE INFOCOM 2019-IEEE conference on computer communications*, pages 2512–2520. IEEE, 2019.

[Wang *et al.*, 2020] Di Wang, Hanshen Xiao, Srinivas Devadas, and Jinhui Xu. On differentially private stochastic convex optimization with heavy-tailed data. In *International Conference on Machine Learning*, pages 10081–10091. PMLR, 2020.

[Wang *et al.*, 2023] Lingxiao Wang, Bargav Jayaraman, David Evans, and Quanquan Gu. Efficient privacy-preserving stochastic nonconvex optimization. In *Uncertainty in Artificial Intelligence*, pages 2203–2213. PMLR, 2023.

[Wei and Liu, 2021] Wenqi Wei and Ling Liu. Gradient leakage attack resilient deep learning. *IEEE Transactions on Information Forensics and Security*, 17:303–316, 2021.

[Wei *et al.*, 2020] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H Yang, Farhad Farokhi, Shi Jin, Tony QS Quek, and H Vincent Poor. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security*, 15:3454–3469, 2020.

[Wei *et al.*, 2021a] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Hang Su, Bo Zhang, and H Vincent Poor. User-level privacy-preserving federated learning: Analysis and performance optimization. *IEEE Transactions on Mobile Computing*, 21(9):3388–3401, 2021.

[Wei *et al.*, 2021b] Kang Wei, Jun Li, Chuan Ma, Ming Ding, Cailian Chen, Shi Jin, Zhu Han, and H Vincent Poor. Low-latency federated learning over wireless channels with differential privacy. *IEEE Journal on Selected Areas in Communications*, 40(1):290–307, 2021.

[Wei *et al.*, 2023] Wenqi Wei, Ling Liu, Jingya Zhou, Ka-Ho Chow, and Yanzhao Wu. Securing distributed sgd against gradient leakage threats. *IEEE Transactions on Parallel and Distributed Systems*, 2023.

[Wu *et al.*, 2020] Nan Wu, Farhad Farokhi, David Smith, and Mohamed Ali Kaafar. The value of collaboration in convex machine learning with differential privacy. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 304–317. IEEE, 2020.

[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.

[Xu *et al.*, 2022] Jie Xu, Wei Zhang, and Fei Wang. A(DP)^{2}SGD: Asynchronous decentralized parallel stochastic gradient descent with differential privacy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):8036–8047, 2022.

[Yu *et al.*, 2021] Dongxiao Yu, Zongrui Zou, Shuzhen Chen, Youming Tao, Bing Tian, Weifeng Lv, and Xiuzhen Cheng. Decentralized parallel sgd with privacy preservation in vehicular networks. *IEEE Transactions on Vehicular Technology*, 70(6):5211–5220, 2021.

[Zeng *et al.*, 2021] Yiming Zeng, Yixuan Lin, Yuanyuan Yang, and Ji Liu. Differentially private federated temporal difference learning. *IEEE Transactions on Parallel and Distributed Systems*, 33(11):2714–2726, 2021.

[Zhang *et al.*, 2017] Jiaqi Zhang, Kai Zheng, Wenlong Mou, and Liwei Wang. Efficient private erm for smooth objectives. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 3922–3928, 2017.

[Zhou *et al.*, 2023] Yipeng Zhou, Xuezheng Liu, Yao Fu, Di Wu, Jessie Hui Wang, and Shui Yu. Optimizing the numbers of queries and replies in convex federated learning with differential privacy. *IEEE Transactions on Dependable and Secure Computing*, 2023.

[Zhu *et al.*, 2019] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. *Advances in neural information processing systems*, 32, 2019.

[Zhu *et al.*, 2024] Zehan Zhu, Ye Tian, Yan Huang, Jinming Xu, and Shibo He. R-FAST: Robust fully-asynchronous stochastic gradient tracking over general topology. *IEEE Transactions on Signal and Information Processing over Networks*, 10:665–678, 2024.

[Zhu *et al.*, 2025] Zehan Zhu, Yan Huang, Xin Wang, Shouling Ji, and Jinming Xu. Dyn-D^{2}P: Dynamic differentially private decentralized learning with provable utility guarantee. *arXiv preprint arXiv:2505.06651*, 2025.

[Zinkevich *et al.*, 2010] Martin Zinkevich, Markus Weimer, Lihong Li, and Alex Smola. Parallelized stochastic gradient descent. *Advances in neural information processing systems*, 23, 2010.