Imagining a Democratic, Affordable Future of Foundation Models: A Decentralised Avenue

Fengxiang He[0000−0001−5584−2385], Lihao Nan[0009−0009−4073−9532], and Tongtian Zhu[0000−0002−5503−9290]

Abstract Foundation models show astonishing performance for a variety of tasks while requiring extremely huge amounts of computing resources in both training and inference. Such costs are beyond the affordability of most users; consequently, foundation models are dominantly occupied by tech giants. To pursue an affordable and democratic future of foundation models, there is growing interest in examining decentralised learning approaches. This chapter provides a thorough review of the current decentralised solutions and offers insights into prospective strategies to overcome the existing barriers. We also describe our insights in facilitating decentralised learning by blockchain, as well as challenges and future work. In our vision, decentralised learning will energise the foundation model economy, but is still obstructed by major challenges such as establishing robust incentive mechanisms and developing training strategies suitable for heterogeneous environments.

1 Introduction

Recently, foundation models [8] (e.g., T5 [61], GPT-3 [13], PaLM [17], OPT [93], GPT-4 [1], Llama-2 [75], Mistral [35] and DALL·E 3 [5]) have made groundbreaking advancements in understanding and generating natural language and images, driven by substantial increases in model size and training data size [13]. Notably, GPT-3 consists of over 100 billion parameters and leverage immensely large datasets for training [13]. This expansion necessitates extremely high demand of CPU, memory,

Fengxiang He
University of Edinburgh, e-mail: F.He@ed.ac.uk

Lihao Nan
Microsoft, e-mail: lihnan@microsoft.com

Tongtian Zhu
Zhejiang University, e-mail: raiden@zju.edu.cn

and GPU hardware. Furthermore, the operational costs associated with running these foundation models at such a massive scale have escalated substantially. For instance, OpenAI reportedly incurs a daily expenditure of $700,000 to maintain ChatGPT [84], despite the fact that training the GPT-3 foundation model alone cost over $5 million [84]. Consequently, only a small set of large corporations with sufficient data and computation resources control the access to the best AI models.

To democratise these advanced technologies for a broader user base, decentralized learning emerges as a promising strategy. The general idea of decentralized learning is to crowdsource the training of machine learning models with thousands of regular volunteers provided by decentralised volunteers via a peer-to-peer (P2P) network. Concretely, one could partition a large model (e.g., a neural network) into thousands of parallel segments and let each volunteer manage one of the segments. The advantages are mainly threefold:

- **Cost amortization**. As more volunteers contribute their computational resources, the costs of computational tasks are spread over a larger number of participants. This reduces the individual cost burden, making participation in the network more affordable;
- **Autonomy**. Decentralization creates an environment where autonomous and democratic participation is naturally encouraged.
- **Fault tolerance**. As the volunteer network grows, the resilience of the system to node and communication failures strengthens, enhancing its overall robustness.

Despite these advantages, training foundation models in a fully decentralised manner presents unique obstacles. The primary challenges include managing and potentially incentivizing intricate coordination among a massive number of heterogeneous volunteers (i.e., with data, computational power, and model heterogeneity) with inconsistent network connectivity. This leads us to an important question:

? Question

How can we effectively coordinate the decentralised training of foundation models with heterogeneous volunteers under inconsistent network connectivity?

To answer this pivotal question, this chapter provides a thorough review of the existing solutions and envisions future strategies aimed at advancing towards a democratic and affordable future of foundation models. Section 2 introduces the basic concepts of deep learning and foundation models, while Section 3 discusses decentralised machine learning strategies and their cutting-edge extensions for training foundation models, highlighting the benefits of communication efficiency and cost-sharing mechanisms in decentralised approaches. Sections 3.1 and 3.2 detail the core principles, motivations, and algorithmic development of decentralised training methods. Section 3.3 explores the specific challenges associated with scaling decentralised techniques to support the training of foundation models, and summarizes current advancements. The chapter further assesses the advantages of integrating Blockchain technology within decentralised learning systems in Section 4.

2 Deep Learning and Foundation Models

Deep learning emerges as a transformative force in artificial intelligence, fundamentally reshaping our understanding and potential within the field. Drawing inspiration from the neural networks of the human brain, deep learning models can learn from large datasets to identify underlying patterns and generalize, that is, to make accurate predictions on unseen data. They have proven to be exceptionally adept across various domains, including language processing [22, 13] to vision [39, 31], outperforming traditional machine learning models.

Foundation models, exemplified by groundbreaking language models like OpenAI's ChatGPT, represent another leap forward. These models are trained on extensive datasets, which lays the groundwork for their remarkable ability to adapt to specialized tasks through fine-tuning. ChatGPT, in particular, also extends beyond standard fine-tuning by incorporating reinforcement learning from human feedback (RLHF) [18], a promising way to align foundation models with human intents. The performance of foundation models can be further elevated by employing techniques such as prompt tuning [13] and in-context learning [82], which refine its ability to interpret and respond to prompts in a context-aware manner Furthermore, techniques like LoRA [32] enable more resource-efficient fine-tuning by integrating low-rank layers into the original model, thereby avoiding the retraining of entire parameters.

Despite these technical achievements, foundation models like GPT-3.5 also present substantial challenges, particularly in terms of the economic investment required for their deployment and development. GPT-3.5, with its extensive ability to generate human-like text, answer complex questions, and craft creative content, comes at the cost of significant hardware and computational demands. These demands render the deployment and training of such models economically infeasible for many individuals and academic institutions. These barriers stand as significant impediments to democratizing access to state-of-the-art models, potentially stifling scientific advancement.

3 Decentralised Machine Learning

This aforementioned challenges necessitates the development of sophisticated distributed learning paradigms. In this section, we introduce decentralised machine learning, which integrates the idea of volunteer computing into distributed machine learning. We start from the formal definition of distributed machine learning, then introduce the algorithmic aspects of decentralised learning, and move on to summarize the latest research on decentralised training of foundation models.

Algorithm 1 Parallel SGD [21, 43]

Worker $j = 1, \dots, m$ (in parallel):
Receive $\theta^0 = 0$ from server
for step $t = 1$ to T **do**
  Sample training batch $\{z_{j,i}\}_{i=1}^{|\mu_j^t|}$ from local training dataset
  Compute gradient $g_j^t := \frac{1}{|\mu_j^t|} \sum_{i=1}^{|\mu_j^t|} \nabla L(\theta^t, z_{j,i})$ ▹ mini-batch gradient computation
  Push g_j^t to server
  Receive θ^{t+1} from server
Server:
for $t = 1$ to T **do**
  Aggregate $g^t := \frac{1}{m} \sum_{j=1}^{m} g_j^t$ ▹ global gradient aggregation
  Set learning rate as η^t, update $\theta^{t+1} := \theta^t - \eta^t g^t$ ▹ global weight update

3.1 Distributed Machine Learning

Notations. We denote $\mathcal{X} \subseteq \mathbb{R}^{d_x}$ and $\mathcal{Y} \subseteq \mathbb{R}$ as the input and output domains, respectively. The training set is denoted as $\mu = \{z_1, \dots, z_N\}$, where each $z_\zeta = (x_\zeta, y_\zeta)$, for $\zeta = 1, \dots, N$, is sampled independent and identically distributed (i.i.d.) from an unknown data distribution $\mathcal{D}$ defined on $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$. The goal of supervised learning is to learn a predictor (or hypothesis) $g(\theta; \cdot)$, parameterized by $\theta \in \mathbb{R}^d$ of an arbitrary finite dimension d, to approximate the mapping between the input variable $x \in \mathcal{X}$ and the output variable $y \in \mathcal{Y}$, based on the training set μ. The cost function $c : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}^+$ is used to evaluate the prediction performance of hypothesis g. The loss of a hypothesis g with respect to (w.r.t.) the example $z_\zeta = (x_\zeta, y_\zeta)$ is defined as $L(\theta; z_\zeta) = c(g(\theta; x_\zeta), y_\zeta)$, which quantifies the performance of the model parameterized by θ. The empirical risk of θ, which is the target of optimisation, is thus defined as follows:

$$L_\theta^\mu = \frac{1}{N} \sum_{\zeta=1}^{N} L(\theta; z_\zeta). \tag{1}$$

Distributed learning. Traditional distributed learning considers optimising the empirical risk jointly with multiple workers [66]. In this framework, each worker, for $j = 1, \dots, m$, can access $|\mu_j|$ i.i.d. local training examples $\mu_j = \{z_{j,1}, \dots, z_{j,|\mu_j|}\}$. The global empirical risk of θ then becomes

$$L_\theta^\mu = \frac{1}{m} \sum_{j=1}^{m} L_\theta^{\mu_j} = \frac{1}{m} \sum_{j=1}^{m} \frac{1}{|\mu_j|} \sum_{\zeta=1}^{|\mu_j|} L(\theta; z_{j,\zeta}), \tag{2}$$

where $L_\theta^{\mu_j} = \frac{1}{|\mu_j|} \sum_{\zeta=1}^{|\mu_j|} L(\theta; z_{j,\zeta})$ denotes the local empirical risk on the j-th worker and $\mu_j = \{z_{j,\zeta}\}_{\zeta=1}^{|\mu_j|}$ represents the local training dataset. The optimisation of equation (2) is also a distributed consensus problem [12].

Algorithm 2 FedAvg [53]

Client $j = 1, \ldots, m$ (in parallel):
Receive $\theta^0 = 0$ from server
for round $t = 1$ to T **do**
  for local step $l = 1$ to E **do**
    Sample local training batch $\{z_{j,i}\}_{i=1}^{|\mu_j^{t,l}|}$ from local training dataset
    Compute gradient $g_j^{t,l} := \frac{1}{|\mu_j^{t,l}|} \sum_{i=1}^{|\mu_j^{t,l}|} \nabla L(\theta^{t,l}, z_{j,i})$ ▹ local gradient computation
    Set local learning rate as $\eta_j^{t,l}$, compute $\theta_j^{t,l+1} := \theta_j^{t,l} - \eta_j^t g_j^{t,l}$ ▹ local weight update
  Push $\theta^{t,E}$ to server
  Receive θ^{t+1} from server
Server:
for $t = 1$ to T **do**
  Aggregate $\theta^{t+1} := \frac{1}{m} \sum_{j=1}^{m} \theta_j^{t,E}$ ▹ global weight aggregation

Due to the increasing scale of training data and the model complexity, various parallelisation strategies have been proposed for effective large-scale distributed learning. The primary strategies include data parallelism, pipeline parallelism, and tensor parallelism.

Data parallelism. In machine learning scenarios where fixed training datasets are available, the aforementioned distributed strategy corresponds to randomly partitioning the whole training samples among the multiple machines. Each machine is thus endowed with a subset of samples that are i.i.d. drawn from the same source distribution. Extending the principle, data parallelism [45] shares the workload by distributing a large mini-batch across multiple devices. Each device trains its own local model replica independently. Ensuring consistency across these local models necessitates synchronization of gradients information via a parameter server [71, 21, 43] or through an AllReduce operation [60, 45]. Algorithm 1 is an example of implementing stochastic gradient descent with a parameter server. In each step, each worker computes their gradients locally, then the server aggregates these gradients to update the shared parameter. Federated Learning (FL) [53, 9, 47] is another famous example of data parallelism. FL considers the problem of collaborative learning with heterogeneous edge devices, where each device j keep its own local data. A typical FL approach is FedAvg (see Algorithm 2), where clients update locally with E steps, and then the server averages these local updated models to refine the global model. The updated global model is then redistributed to the clients, facilitating subsequent local training cycles. This decentralised approach leverages the computational resources of the edge devices and also respects user privacy by keeping sensitive data locally. Despite the advantages offered by data parallelism, its scalability is mainly constrained by the following factors: the significant communication overhead required for model parameter synchronization which could limit scalability [42, 47, 14], the reduction of GPU utilization as the per-GPU batch size becomes too small [57], and the memory constraints that render loading an entire large model onto a single GPU impractical.

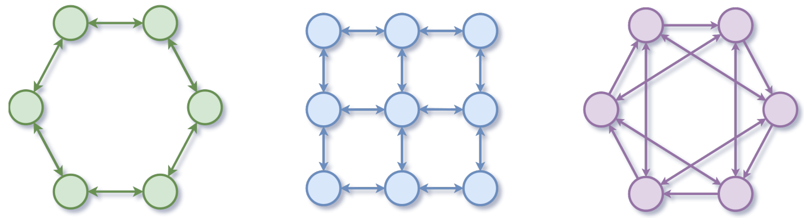

Fig. 1 An illustration of peer-to-peer communication topology in decentralised learning.

Pipeline parallelism. Addressing the challenge of individual GPU memory, pipeline parallelism partitions a model into finer slices at the layer-level, each processed on separate devices in a sequential fashion [33, 24, 55]. The downside of this approach is that the sequential layer processing creates dependencies that can limit scaling efficiency. These dependencies often result in potential idle time, known as "bubble time", where some devices wait for others to complete their tasks before proceeding [56, 57].

Tensor parallelism. In tensor model parallelism [70], matrix multiplications within each individual layer are split over multiple devices. This form of parallelism is especially fitting for super-large models, and requires access to high communication-bandwidth environments for efficiently handling the intensive data exchange [57]. However, in cases when intra-group communication is not fast enough, tensor model parallelism may exhibit subpar performance. Therefore, tensor parallelism is typically applied within a single physical server, in conjunction with complementary parallelisation strategies [34, 57].

3.2 Decentralised Machine Learning

To mitigate the communication bottleneck of server-based distributed machine learning, decentralised learning emerges as a powerful alternative. Employing a peer-to-peer approach, decentralised training harnesses the power of locally connected computing resources, effectively distributing the workload without the need for a central coordinating server [81, 11, 92, 4, 52].

The conceptual foundations of decentralised training algorithms are rooted in the early and influential work of [77], [76] and [59]. These studies provide the groundwork for the development of algorithms such as Decentralised Parallel Stochastic Gradient Descent (D-SGD) [48, 38], which integrates the principles of decentralisation with gradient-based optimisation (see Algorithm 3). In the vanilla Adapt-While-Communicate (AWC) version of D-PSGD [59, 48], each worker updates its own model locally and incorporating weights from peers. During weight exchange, a "sender" shares its locally trained model with its neighbors, and a "receiver" integrates these models into its local model. This peer-to-peer communication is through a gossip protocol orchestrated by a mixing matrix $P = [P_{j,k}] \in \mathbb{R}^{m \times m}$, which char-

Algorithm 3 Decentralised Parallel SGD [48, 38]

Input: Given communication graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and mixing matrix $P = [P_{j,k}] \in \mathbb{R}^{m \times m}$
Worker $j = 1, \ldots, m$ (in parallel):
1: Initialize $\theta_j^0 = 0$
2: **for** step $t = 1$ to T **do**
3: Sample training batch $\{z_{j,i}\}_{i=1}^{|\mu_j^t|}$ from local training dataset
4: **for all** neighbors $k : \{j, k\} \in \mathcal{E}$ **do**
5: Compute $\theta_j^{t+\frac{1}{2}} = \sum_{k=1}^{m} P_{j,k} \theta_k^t$ ▷ gossip weight aggregation
6: Compute gradient $g_j^t := \frac{1}{|\mu_j^t|} \sum_{i=1}^{|\mu_j^t|} \nabla L(\theta_j^t, z_{j,i})$ ▷ mini-batch gradient computation
7: Compute $\theta_j^{t+1} = \theta_j^{t+\frac{1}{2}} - g_j^t$ ▷ local weight update

acterizes the connectivity of the underlying communication topology $\mathcal{G}$ [96]. The central goal of D-PSGD is to establish a consensus model by optimising the empirical risk $\frac{1}{m} \sum_{j=1}^{m} \frac{1}{|\mu_j|} \sum_{\zeta=1}^{|\mu_j|} L(\theta; z_{j,\zeta})$ (see equation (2)) cooperatively through m locally-connected workers.

Theoretical research has shown that large-scale models can effectively converge with D-PSGD [50, 68], with asymptotic linear speedup in convergence rate similar to centralised parallel SGD (C-SGD) [20, 44]. Recent studies [95] have further linked D-PSGD to a centralised generalisation-enhancing algorithm called Sharpness-Aware Minimization (SAM), suggesting that decentralised learning may offer additional generalisation benefits compared to server-based learning paradigms.

The development of decentralised algorithms has been characterized by their flexibility in adapting to complex environments. Notably, decentralised algorithms have been adapted to various contexts, including time-varying topologies [58, 51, 38, 91], asynchronous settings [49, 88, 54, 10], personalized settings [46], data-heterogeneous scenarios [72, 78, 41] and Byzantine-robust versions [90, 25]. Decentralised optimisation problems have been further extended beyond standard single-level minimization problems, including compositional [27], minimax [87, 94, 15], and bi-level [89, 26, 16] optimisation problems. Despite these advancements, existing decentralised training approaches predominantly focus on data parallelism, which alone could be inadequate for foundation models whose parameter sets are too large to be accommodated by a single device.

3.3 Decentralised Training and Inference of Foundation Models

Foundation models have reaped substantial rewards from the expansion of training data and model complexity, in accordance with the principles of scaling laws [63, 37]. However, this trend towards larger data size and models has outstripped the evolution of hardware, which trails behind the escalating requirements for computing power and memory. As a result, training and deploying modern foundation models not only requires advanced GPUs, but often necessitates specialized High-Performance

Table 1 Review of Methods, Framework and Platforms for Decentralised Training and Inference of Foundation Models.

Methods	Description
Learning@home [65]	A decentralised mixture-of-experts (MoE) training paradigm for massive, poorly connected networks
DeDLOC [23]	A decentralised data-parallel framework using adaptive averaging strategy for collaborative training under diverse internet speeds and connectivity challenges
DT-FM [92]	A decentralised pipeline parallel method for training GPT-style foundation models, employing a specialized algorithm for “tasklet” allocation over heterogeneous and lower-bandwidth networks.
SWARM Parallelism [64]	A parallel training strategy for training billions of parameters across unreliable, heterogeneous devices with slow connectivity
FusionAI [73]	A distributed system supporting dynamic join and quit policy for training large language models with underutilized consumer-grade GPUs
Petal [11]	A decentralised collaborative inference service engine for cost sharing
HexGen [36]	A decentralised inference method supporting asymmetric partitioning of the inference computation by reformulating the scheduling problem as a constrained optimisation problem
AQ-SGD [80]	A decentralised activation compression algorithm for communication-efficient pipeline parallelism training over slow networks
CocktailSGD [79]	A communication-efficient algorithm combining decentralisation, sparsification, and quantization
SAKSHI [6]	A decentralised platform for energy-efficient, trust-free, and incentive-compatible AI service hosting and delivery

Computing (HPC) clusters to handle their substantial computational demands. Sophisticated parallelisation strategies like data, pipeline, and tensor parallelism are widely used, yet they assume the availability of luxury data centers equipped with fast interconnects, which is beyond the budget of many individuals and academic institutions. The immensity of this challenge is exemplified by the requirements of foundation models like GPT-3, which requires 325GB of GPU memory [67] and 3.64K petaflop/s-days for training [13]. Such requirements starkly illustrate the daunting barriers faced by those with limited access to such computational resources.

Fully decentralised training of Foundation Models. Thanks to the advantages in communication efficiency, cost sharing and fault tolerance, decentralised approaches have emerges as promising alternative to train foundation models such as Large Language Models (LLMs). Leveraging the concept of volunteer computing [69, 2, 3, 40], Learning@home [65] and DeDLOC [23] spearhead the collaborative volunteer training of foundation models. Learning@home [65] proposes a promising decentralised mixture-of-experts (MoE) training paradigm to handle massive poorly connected participants with a Decentralised Hash Table (DHT) used to route inputs to the appropriate expert. However, the training and evaluation of Learning@home is confined to relatively smaller datasets. DeDLOC [23] uses a decentralised adaptive averaging strategy that considers the diverse internet speeds and connectivity limitations of volunteers, but still relys on data parallelism. [92] explores the potential of training standard GPT-style foundation models with a new decentralised model parallelism over a heterogeneous and lower-bandwidth interconnected network. The major contribution is a scheduling algorithm allocating computational "tasklets". Subsequent work by SWARM Parallelism [64] leverages fault-tolerant pipelines and dynamically rebalances nodes across stages to train foundation models on heterogeneous devices under slower connectivity. In parallel, [73] proposes FusionAI supporting dynamic join and quit policy for training large language models with underutilized consumer-grade GPUs. Based on swarm parallelism, Petal [11] develops a decentralised pipeline inference framework to amortize inference cost of LLMs. Petal facilitates a collaborative environment wherein users can donate heterogeneous computation resources to perform inference and small-scale fine-tuning collaboratively. A more recent work inference method called HexGen [36] can further allocate the asymmetric inference tasklets among workers by reformulating the scheduling problem as a constrained optimisation problem. At the algorithmic level, CocktailSGD [79] elegantly combines decentralisation, sparsification, and quantization for communication-efficient fine-tuning of foundation models on slow networks. AQ-SGD [80] introduces a decentralised activation compression algorithm for communication-efficient pipeline parallelism training over slow networks. Beyond the framework and algorithmic design, SAKSHI [6] emerges as a new decentralised platform for energy-efficient, trust-free and incentive compatible AI service hosting and delivery.

4 Decentralised Learning on Blockchain

As highlighted in the preceding section, decentralised learning stands out as an attractive strategy for training foundation models, offering notable benefits in terms of communication efficiency and cost-sharing. However, the absence of effective incentive mechanism and reliable security assurances remains a critical hurdle for such systems. Blockchain technology, characterized by its secure, auditable, immutable, incentive-based and decentralised nature, presents a natural auxiliary to decentralised learning that encourages a collaborative environment [30]. In this section, we discuss the potential benefits of integrating Blockchain technology into decentralised learning.

Automation. By combining blockchain technology with smart contracts [74], users can execute verifiable and traceable transactions autonomously, aligning with the self-organizing principles in decentralized learning systems.

Security and integrity. The security and integrity of transactions in a blockchain network are ensured through a verification process. Each account in the blockchain holds a public key and a private key, with the public key available to everyone and the private key only visible to the account owner. When a user, designated as the Request

Table 2 Advantages of Integrating Blockchain with Decentralised Learning

Aspect	Role of Blockchain	Impact on Decentralised Learning
Automation	Smart contracts	Automatically execute decentralised training
Security & **Integrity**	Robust encryption mechanisms; Tamper-resistant ledgers	Secures model/data exchange by ensuring only authorized access, maintaining the immutability of records
Incentivization	Token-based reward mechanism	Encourages active and fair participation in model training

Node, submits a transaction, it uses its private key to create a digital signature for the transaction data. This digital signature is unique to the transaction and the private key of the Request Node. The verification process involves the following steps:

1. **Hash the transaction data**: The transaction data is hashed, creating a fixed-length string of characters that uniquely represents the transaction.
2. **Decrypt the digital signature**: The public key is used to decrypt the digital signature, revealing the original hash of the transaction data.
3. **Compare hashes**: The decrypted hash is compared to the hash of the transaction data. If they match, it means the transaction data has not been tampered with.
4. **Verify ownership**: The network also verifies that the public key used to create the digital signature belongs to the Request Node. If the signature is valid and the public key matches the public key of the Request Node on record, the transaction is considered legitimate.

Blockchain, further integrated with zero-knowledge proofs [29, 7, 28], could offer a robust framework for safeguarding information exchange in decentralized learning, where nodes can confirm the legitimacy of the contributions of others without compromising privacy. In scenarios where there are many heterogeneous participants, the data owner and computation provider can be decoupled. Consider a Request Node falls short for substantial computational tasks. A straightforward solution is to transmit the code and data to an entity with stronger computational power, known as the Compute Node, as illustrated in Figure 2. In the scenario of decentralized training, a natural question is

? Questions

Execution Verification: How to confirm that the Compute Node has actually executed the instruction from the Request Node and not fabricated the results in a fully decentralised learning system?

In a collaborative learning system, the Compute Node can generate a zk-SNARK proof after completing a computation task. This proof validates the correctness of the computation without revealing the details. When a new computational task is

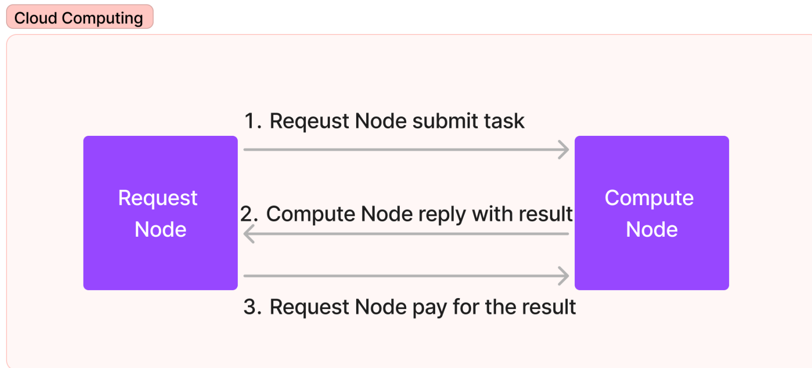

Fig. 2 Procedure for Computation in a Centralised Cloud Computing Environment

submitted to the blockchain, miner nodes can verify this zk-SNARK proof in the same way they would verify a transaction, without access to the specific details of the computation. The strength of this system lies in the inherent immutability of blockchain; once transactions of model updates are recorded on the ledger, they are permanent and cannot be altered, effectively preventing unauthorized modifications. This characteristic is vital for creating a verifiable and trustable collaborative learning environment, particularly in sensitive sectors such as healthcare where data provenance and integrity are critical [81].

Incorporate Incentives. One of the primary goals of a self-organized decentralised learning system is to foster sustainable collaboration among diverse participants, thus necessitating the design of a robust incentive mechanism to effectively motivate contributors; while also preventing unconstructive participation from receiving rewards. In a manner akin to how Filecoin [85] complements IPFS [86] by providing an incentive layer, blockchain could facilitate the establishment for such a mechanism in decentralized learning, distributing tokens or cryptocurrencies as rewards for valuable contributions based on smart contracts. These incentives are crucial for encouraging participants to contribute computational and communication resources or even local data, foundational to establishing a transparent and cost-effective collaborative environment for training foundation models.

Despite the potential benefits, leveraging blockchain for training large-scale foundation models presents challenges.

High communication costs. BAFFLE [62] and DeepChain [83] have leveraged blockchain to mitigate the security and privacy issues in federated learning. However, these works have not considered leveraging Blockchain to train large-scale deep learning models, such as foundation models. The core challenges here lie in the inefficiencies in parallelising the structure of current foundation models and the high communication costs associated with Blockchain [19]. Therefore, designing model-parallel, communication-efficient decentralized algorithms for training foundation models could be a promising future direction.

5 Conclusion

Foundation models are exceptionally effective for a broad range of tasks; however, they require substantial computational resources for both training and inference, making them financially and technically unattainable for most players. As a result, the control over foundation models is predominantly held by tech giants. The pursuit of a democratic and affordable future is in the interest of wide communities. In this chapter, we discuss of a possible avenue via decentralised learning. We provide a comprehensive review of current decentralised learning methods, the open problems and existing technical challenges, and prospective approaches to address them. We envision that decentralised methodologies could energise the economy based upon foundation models; however, progress is still hindered by challenges including establishing robust incentive mechanisms and developing training strategies suitable for heterogeneous environments. In this context, blockchain technologies can play a significant role in facilitating decentralised learning.

References

1. Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
2. David P. Anderson, Jeff Cobb, Eric Korpela, Matt Lebofsky, and Dan Werthimer. Seti@home: an experiment in public-resource computing. *Communications of the ACM*, 45(11):56–61, 2002.
3. D.P. Anderson. Boinc: a system for public-resource computing and storage. In *Fifth IEEE/ACM International Workshop on Grid Computing*, pages 4–10, 2004.
4. Enrique Tomás Martínez Beltrán, Mario Quiles Pérez, Pedro Miguel Sánchez Sánchez, Sergio López Bernal, Gérôme Bovet, Manuel Gil Pérez, Gregorio Martínez Pérez, and Alberto Huertas Celdrán. Decentralized federated learning: Fundamentals, state-of-the-art, frameworks, trends, and challenges. *arXiv preprint arXiv:2211.08413*, 2022.
5. James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf*, 2(3):8, 2023.
6. Suma Bhat, Canhui Chen, Zerui Cheng, Zhixuan Fang, Ashwin Hebbar, Sreeram Kannan, Ranvir Rana, Peiyao Sheng, Himanshu Tyagi, Pramod Viswanath, et al. Sakshi: Decentralized ai platforms. *arXiv preprint arXiv:2307.16562*, 2023.
7. Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, STOC '88, page 103–112. Association for Computing Machinery, 1988.
8. Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.
9. Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloé Kiddon, Jakub Konečný, Stefano Mazzocchi, Brendan McMahan, Timon Van Overveldt, David Petrou, Daniel Ramage, and Jason Roselander. Towards federated learning at scale: System design. In A. Talwalkar, V. Smith, and M. Zaharia, editors, *Proceedings of Machine Learning and Systems*, volume 1, pages 374–388, 2019.

10. Marco Bornstein, Tahseen Rabbani, Evan Z Wang, Amrit Bedi, and Furong Huang. SWIFT: Rapid decentralized federated learning via wait-free model communication. In *The Eleventh International Conference on Learning Representations*, 2023.
11. Alexander Borzunov, Dmitry Baranchuk, Tim Dettmers, Maksim Riabinin, Younes Belkada, Artem Chumachenko, Pavel Samygin, and Colin Raffel. Petals: Collaborative inference and fine-tuning of large models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 558–568. Association for Computational Linguistics, 2023.
12. Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.
13. Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
14. Xuanyu Cao, Tamer Başar, Suhas Diggavi, Yonina C. Eldar, Khaled B. Letaief, H. Vincent Poor, and Junshan Zhang. Communication-efficient distributed learning: An overview. *IEEE Journal on Selected Areas in Communications*, 41(4):851–873, 2023.
15. Lesi Chen, Haishan Ye, and Luo Luo. An efficient stochastic algorithm for decentralized nonconvex-strongly-concave minimax optimization. *International Conference on Artificial Intelligence and Statistics*, 2024.
16. Xuxing Chen, Minhui Huang, Shiqian Ma, and Krishna Balasubramanian. Decentralized stochastic bilevel optimization with improved per-iteration complexity. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pages 4641–4671. PMLR, 2023.
17. Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.
18. Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.
19. Kyle Croman, Christian Decker, Ittay Eyal, Adem Efe Gencer, Ari Juels, Ahmed Kosba, Andrew Miller, Prateek Saxena, Elaine Shi, Emin Gün Sirer, Dawn Song, and Roger Wattenhofer. On scaling decentralized blockchains. In Jeremy Clark, Sarah Meiklejohn, Peter Y.A. Ryan, Dan Wallach, Michael Brenner, and Kurt Rohloff, editors, *Financial Cryptography and Data Security*, pages 106–125. Springer Berlin Heidelberg, 2016.
20. Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Marc'aurelio Ranzato, Andrew Senior, Paul Tucker, Ke Yang, et al. Large scale distributed deep networks. *Advances in neural information processing systems*, 2012.
21. Jeffrey Dean, Greg S. Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Quoc V. Le, Mark Z. Mao, Marc'Aurelio Ranzato, Andrew Senior, Paul Tucker, Ke Yang, and Andrew Y. Ng. Large scale distributed deep networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, page 1223–1231. Curran Associates Inc., 2012.

22. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019.
23. Michael Diskin, Alexey Bukhtiyarov, Max Ryabinin, Lucile Saulnier, Quentin Lhoest, Anton Sinitsin, Dmitry Popov, Dmitriy Pyrkin, Maxim Kashirin, Alexander Borzunov, Albert Villanova del Moral, Denis Mazur, Ilia Kobelev, Yacine Jernite, Thomas Wolf, and Gennady Pekhimenko. Distributed deep learning in open collaborations. In *Advances in Neural Information Processing Systems*, 2021.
24. Shiqing Fan, Yi Rong, Chen Meng, Zongyan Cao, Siyu Wang, Zhen Zheng, Chuan Wu, Guoping Long, Jun Yang, Lixue Xia, Lansong Diao, Xiaoyong Liu, and Wei Lin. Dapple: a pipelined data parallel approach for training large models. In *Proceedings of the 26th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, PPoPP '21, page 431–445. Association for Computing Machinery, 2021.
25. Sadegh Farhadkhani, Rachid Guerraoui, Nirupam Gupta, Lê Nguyên Hoang, Rafael Pinot, and John Stephan. Robust collaborative learning with linear gradient overhead. In *International Conference on Machine Learning*, 2023.
26. Hongchang Gao, Bin Gu, and My T. Thai. On the convergence of distributed stochastic bilevel optimization algorithms over a network. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206, pages 9238–9281. PMLR, 2023.
27. Hongchang Gao and Heng Huang. Fast training method for stochastic compositional optimization problems. *Advances in Neural Information Processing Systems*, 34:25334–25345, 2021.
28. Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity or all languages in np have zero-knowledge proof systems. *Journal of the ACM*, 38(3):690–728, 1991.
29. S Goldwasser, S Micali, and C Rackoff. The knowledge complexity of interactive proof-systems. In *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing*, STOC '85, page 291–304. Association for Computing Machinery, 1985.
30. Justin D Harris and Bo Waggoner. Decentralized and collaborative ai on blockchain. In *2019 IEEE international conference on blockchain (Blockchain)*, pages 368–375. IEEE, 2019.
31. 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 (CVPR)*, June 2016.
32. Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.
33. Yanping Huang, Youlong Cheng, Ankur Bapna, Orhan Firat, Dehao Chen, Mia Chen, HyoukJoong Lee, Jiquan Ngiam, Quoc V Le, Yonghui Wu, et al. Gpipe: Efficient training of giant neural networks using pipeline parallelism. *Advances in neural information processing systems*, 32, 2019.
34. Zhihao Jia, Matei Zaharia, and Alex Aiken. Beyond data and model parallelism for deep neural networks. *Proceedings of Machine Learning and Systems*, 1:1–13, 2019.
35. Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.
36. Youhe Jiang, Ran Yan, Xiaozhe Yao, Beidi Chen, and Binhang Yuan. Hexgen: Generative inference of foundation model over heterogeneous decentralized environment. *arXiv preprint arXiv:2311.11514*, 2023.
37. Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.
38. Anastasia Koloskova, Nicolas Loizou, Sadra Boreiri, Martin Jaggi, and Sebastian Stich. A unified theory of decentralized SGD with changing topology and local updates. In *Proceedings*

of the 37th International Conference on Machine Learning, volume 119 of *Proceedings of Machine Learning Research*, pages 5381–5393. PMLR, 2020.
39. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.
40. Stefan M Larson, Christopher D Snow, Michael Shirts, and Vijay S Pande. Folding@ home and genome@ home: Using distributed computing to tackle previously intractable problems in computational biology. *arXiv preprint arXiv:0901.0866*, 2009.
41. Batiste Le Bars, Aur'elien Bellet, Marc Tommasi, Erick Lavoie, and Anne-Marie Kermarrec. Refined convergence and topology learning for decentralized sgd with heterogeneous data. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206, pages 1672–1702. PMLR, 25–27 Apr 2023.
42. Mu Li, David G. Andersen, Jun Woo Park, Alexander J. Smola, Amr Ahmed, Vanja Josifovski, James Long, Eugene J. Shekita, and Bor-Yiing Su. Scaling distributed machine learning with the parameter server. In *11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 14)*, pages 583–598, 2014.
43. Mu Li, David G. Andersen, Alexander Smola, and Kai Yu. Communication efficient distributed machine learning with the parameter server. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'14, page 19–27. MIT Press, 2014.
44. Mu Li, David G Andersen, Alexander J Smola, and Kai Yu. Communication efficient distributed machine learning with the parameter server. *Advances in Neural Information Processing Systems*, 2014.
45. Shen Li, Yanli Zhao, Rohan Varma, Omkar Salpekar, Pieter Noordhuis, Teng Li, Adam Paszke, Jeff Smith, Brian Vaughan, Pritam Damania, and Soumith Chintala. Pytorch distributed: Experiences on accelerating data parallel training. *Proc. VLDB Endow.*, 13(12):3005–3018, 2020.
46. Shuangtong Li, Tianyi Zhou, Xinmei Tian, and Dacheng Tao. Learning to collaborate in decentralized learning of personalized models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9766–9775, 2022.
47. Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE signal processing magazine*, 37(3):50–60, 2020.
48. 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. In *Advances in Neural Information Processing Systems*, volume 30, 2017.
49. Xiangru Lian, Wei Zhang, Ce Zhang, and Ji Liu. Asynchronous decentralized parallel stochastic gradient descent. In *International Conference on Machine Learning*, 2018.
50. Jie Lu, Choon Yik Tang, Paul R Regier, and Travis D Bow. Gossip algorithms for convex consensus optimization over networks. *IEEE Transactions on Automatic Control*, 2011.
51. Songtao Lu and Chai Wah Wu. Decentralized stochastic non-convex optimization over weakly connected time-varying digraphs. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.
52. Yucheng Lu and Christopher De Sa. Decentralized learning: Theoretical optimality and practical improvements. *Journal of Machine Learning Research*, 24(93):1–62, 2023.
53. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282. PMLR, 2017.
54. Giorgi Nadiradze, Amirmojtaba Sabour, Peter Davies, Shigang Li, and Dan Alistarh. Asynchronous decentralized sgd with quantized and local updates. *Advances in Neural Information Processing Systems*, 2021.
55. Deepak Narayanan, Aaron Harlap, Amar Phanishayee, Vivek Seshadri, Nikhil R Devanur, Gregory R Ganger, Phillip B Gibbons, and Matei Zaharia. Pipedream: Generalized pipeline parallelism for dnn training. In *Proceedings of the 27th ACM Symposium on Operating Systems Principles*, pages 1–15, 2019.

56. Deepak Narayanan, Aaron Harlap, Amar Phanishayee, Vivek Seshadri, Nikhil R. Devanur, Gregory R. Ganger, Phillip B. Gibbons, and Matei Zaharia. Pipedream: generalized pipeline parallelism for dnn training. In *Proceedings of the 27th ACM Symposium on Operating Systems Principles*, SOSP '19, page 1–15. Association for Computing Machinery, 2019.
57. Deepak Narayanan, Mohammad Shoeybi, Jared Casper, Patrick LeGresley, Mostofa Patwary, Vijay Korthikanti, Dmitri Vainbrand, Prethvi Kashinkunti, Julie Bernauer, Bryan Catanzaro, Amar Phanishayee, and Matei Zaharia. Efficient large-scale language model training on gpu clusters using megatron-lm. SC '21. Association for Computing Machinery, 2021.
58. Angelia Nedi'c and Alex Olshevsky. Distributed optimization over time-varying directed graphs. volume 60, pages 601–615. IEEE, 2014.
59. Angelia Nedic and Asuman Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.
60. Pitch Patarasuk and Xin Yuan. Bandwidth optimal all-reduce algorithms for clusters of workstations. *Journal of Parallel and Distributed Computing*, 69(2):117–124, 2009.
61. Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.
62. Paritosh Ramanan and Kiyoshi Nakayama. Baffle : Blockchain based aggregator free federated learning. In *2020 IEEE International Conference on Blockchain (Blockchain)*, pages 72–81, 2020.
63. Jonathan S. Rosenfeld, Amir Rosenfeld, Yonatan Belinkov, and Nir Shavit. A constructive prediction of the generalization error across scales. In *International Conference on Learning Representations*, 2020.
64. Max Ryabinin, Tim Dettmers, Michael Diskin, and Alexander Borzunov. SWARM parallelism: Training large models can be surprisingly communication-efficient. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 29416–29440. PMLR, 2023.
65. Max Ryabinin and Anton Gusev. Towards crowdsourced training of large neural networks using decentralized mixture-of-experts. In *Advances in Neural Information Processing Systems*, volume 33, 2020.
66. Ohad Shamir and Nathan Srebro. Distributed stochastic optimization and learning. In *2014 52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2014.
67. Ying Sheng, Lianmin Zheng, Binhang Yuan, Zhuohan Li, Max Ryabinin, Beidi Chen, Percy Liang, Christopher Re, Ion Stoica, and Ce Zhang. FlexGen: High-throughput generative inference of large language models with a single GPU. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pages 31094–31116. PMLR, 2023.
68. Wei Shi, Qing Ling, Gang Wu, and Wotao Yin. Extra: An exact first-order algorithm for decentralized consensus optimization. *SIAM Journal on Optimization*, 2015.
69. Michael Shirts and Vijay S Pande. Screen savers of the world unite! *Science*, 290(5498):1903–1904, 2000.
70. Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019.
71. Alexander Smola and Shravan Narayanamurthy. An architecture for parallel topic models. *Proc. VLDB Endow.*, 3(1–2):703–710, 2010.
72. Hanlin Tang, Xiangru Lian, Ming Yan, Ce Zhang, and Ji Liu. D2: Decentralized training over decentralized data. In *International Conference on Machine Learning*. PMLR, 2018.
73. Zhenheng Tang, Yuxin Wang, Xin He, Longteng Zhang, Xinglin Pan, Qiang Wang, Rongfei Zeng, Kaiyong Zhao, Shaohuai Shi, Bingsheng He, et al. Fusionai: Decentralized training and deploying llms with massive consumer-level gpus. *arXiv preprint arXiv:2309.01172*, 2023.
74. Palina Tolmach, Yi Li, Shang-Wei Lin, Yang Liu, and Zengxiang Li. A survey of smart contract formal specification and verification. *ACM Computing Surveys*, 54(7), jul 2021.

75. Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.
76. John Tsitsiklis, Dimitri Bertsekas, and Michael Athans. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE transactions on automatic control*, 31(9):803–812, 1986.
77. John Nikolas Tsitsiklis. Problems in decentralized decision making and computation. Technical report, Massachusetts Inst of Tech Cambridge Lab for Information and Decision Systems, 1984.
78. 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.
79. Jue Wang, Yucheng Lu, Binhang Yuan, Beidi Chen, Percy Liang, Christopher De Sa, Christopher Re, and Ce Zhang. CocktailSGD: Fine-tuning foundation models over 500Mbps networks. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 36058–36076. PMLR, 2023.
80. Jue Wang, Binhang Yuan, Luka Rimanic, Yongjun He, Tri Dao, Beidi Chen, Christopher Ré, and Ce Zhang. Fine-tuning language models over slow networks using activation quantization with guarantees. *Advances in Neural Information Processing Systems*, 35:19215–19230, 2022.
81. Stefanie Warnat-Herresthal, Hartmut Schultze, Krishnaprasad Lingadahalli Shastry, Sathyanarayanan Manamohan, Saikat Mukherjee, Vishesh Garg, Ravi Sarveswara, Kristian H"andler, Peter Pickkers, N Ahmad Aziz, et al. Swarm learning for decentralized and confidential clinical machine learning. *Nature*, 2021.
82. Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.
83. Jiasi Weng, Jian Weng, Jilian Zhang, Ming Li, Yue Zhang, and Weiqi Luo. Deepchain: Auditable and privacy-preserving deep learning with blockchain-based incentive. *IEEE Transactions on Dependable and Secure Computing*, 18(5):2438–2455, 2021.
84. Wikipedia contributors. Chatgpt — Wikipedia, the free encyclopedia, 2023. [Online; accessed 29-October-2023].
85. Wikipedia contributors. Filecoin — Wikipedia, the free encyclopedia, 2023. [Online; accessed 10-December-2023].
86. Wikipedia contributors. Interplanetary file system — Wikipedia, the free encyclopedia, 2023. [Online; accessed 10-December-2023].
87. Wenhan Xian, Feihu Huang, Yanfu Zhang, and Heng Huang. A faster decentralized algorithm for nonconvex minimax problems. *Advances in Neural Information Processing Systems*, 34:25865–25877, 2021.
88. 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*, 2021.
89. Shuoguang Yang, Xuezhou Zhang, and Mengdi Wang. Decentralized gossip-based stochastic bilevel optimization over communication networks. *Advances in Neural Information Processing Systems*, 35:238–252, 2022.
90. Zhixiong Yang, Arpita Gang, and Waheed U. Bajwa. Adversary-resilient distributed and decentralized statistical inference and machine learning: An overview of recent advances under the byzantine threat model. *IEEE Signal Processing Magazine*, 37(3):146–159, 2020.
91. Bicheng Ying, Kun Yuan, Yiming Chen, Hanbin Hu, Pan Pan, and Wotao Yin. Exponential graph is provably efficient for decentralized deep training. In *Advances in Neural Information Processing Systems*, 2021.
92. Binhang Yuan, Yongjun He, Jared Quincy Davis, Tianyi Zhang, Tri Dao, Beidi Chen, Percy Liang, Christopher Re, and Ce Zhang. Decentralized training of foundation models in heterogeneous environments. *Advances in Neural Information Processing Systems*, 35, 2022.
93. Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

94. Miaoxi Zhu, Li Shen, Bo Du, and Dacheng Tao. Stability and generalization of the decentralized stochastic gradient descent ascent algorithm. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.
95. Tongtian Zhu, Fengxiang He, Kaixuan Chen, Mingli Song, and Dacheng Tao. Decentralized SGD and average-direction SAM are asymptotically equivalent. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 43005–43036. PMLR, 2023.
96. Tongtian Zhu, Fengxiang He, Lan Zhang, Zhengyang Niu, Mingli Song, and Dacheng Tao. Topology-aware generalization of decentralized SGD. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 27479–27503. PMLR, 2022.