A PRELIMINARY REPORT ON DISTRO

Bowen Peng
Nous Research
bloc@nousresearch.com

Jeffrey Quesnelle
Nous Research
emozilla@nousresearch.com

Dillon Rolnick
Nous Research
dillon@nousresearch.com

Ari Lotter
Nous Research
ari@nousresearch.com

Umer H. Adil
Nous Research
umer@nousresearch.com

Esteban La Rocca
Nous Research
a4nous@pm.me

ABSTRACT

Training large scale neural networks typically involves sharing gradients between all accelerators, which necessitates specialized, high-speed interconnects. To address this, we introduce DisTrO, a family of architecture-agnostic and network-agnostic distributed optimizers that reduces the inter-GPU communication requirements by four to five orders of magnitude without relying on amortized analysis, enabling low-latency training of large neural networks on slow internet bandwidths with heterogeneous networking hardware. In this preliminary report we are excited to show the first and earliest empirical proof that DisTrO-AdamW matches standard AdamW+All-Reduce in convergence rate while massively reducing the required bandwidth during pre-training of a 1.2B LLM. When using Distributed Data Parallelism, DisTrO may enable future large scale foundation model training to bypass the need for high-speed interconnects entirely.

1 Introduction

Large language models (LLMs) and large diffusion models (LDMs) are currently computationally expensive to train due to their high parameter count[1], which requires the use of multiple accelerators (e.g. GPUs, TPUs) to make training tractable. Various methods exist to split the training among these accelerators, such as Distributed Data Parallelism (DDP) [10] and Fully Sharded Data Parallelism (FSDP) [26]. These methods typically maintain the paradigm of synchronously training a single model. To achieve this, though, the full gradients produced by each training step must be synchronized between the accelerators, which requires extremely high bandwidth interconnects between every accelerator. Thus, each training step involves sharing vast amounts of data[2] across perhaps thousands of accelerators.

Given the current bandwidth requirements, GPUs must be physically close to one another and be hardwired together as a single "super cluster" with specialized high-speed interconnects. Building data centers that can accommodate thousands of interconnected GPUs requires massive up-front capital expenditures, recurring costs, and dedicated infrastructure for land, power and cooling. Today, only governments and large tech companies in wealthy countries have the capital and human resources to train LLMs and LDMs.

In this preliminary report, we show the very first empirical proof that not only is it possible to train large neural networks over a highly constrained bandwidth scenario, but that an optimizer can be designed to match AdamW in convergence

[1] As of late 2024, it is not uncommon to talk about models with trillions of parameters.

[2] The size of the gradients shared across all GPUs when using DDP is roughly in the order of the model parameter size multiplied by the number of GPUs where the model is duplicated across.

rate, all while reducing the required inter-GPU communication requirements by four to five orders of magnitude during pre-training and fine-tuning.

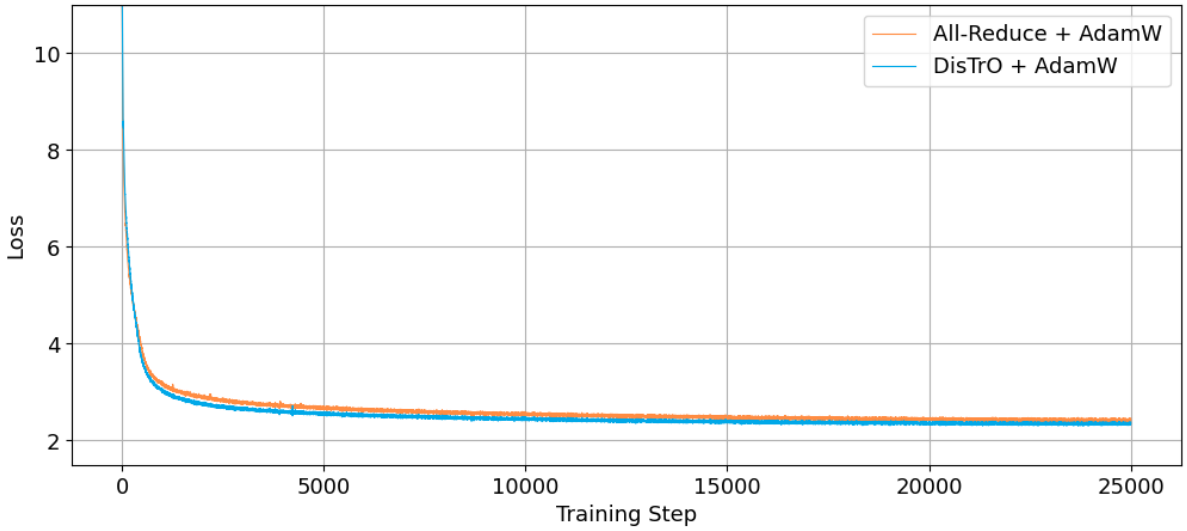
Our Distributed Training Over-the-Internet (DisTrO) family of optimizers massively reduces inter-GPU communication requirements for each training step without using amortization, enabling low latency, efficient and no-compromise pre-training of large neural networks over consumer-grade internet connections using heterogeneous networking hardware. DisTrO is general, scalable and clock-synchronous[3]. In contrast to previous ad-hoc low-communication optimizers (e.g. [12] [11] [20] [16] [8] [3] [25]), DisTrO is also agnostic to the telecommunication network's topology and the neural network architecture while natively supporting distributed data parallel training (DDP) with minimal overhead.

Currently, we do not fully understand the theory behind the unexpected and unreasonable effectiveness of DisTrO, but a more rigorous and fully detailed academic paper is in progress, where we hope to derive a unified theory of distributed training regarding dense[4] neural networks. As we narrow in on the frontier of the absolute minimum necessary communication required to train a model, we see the potential for the formulation of a more generalized information-theoretic description of neural net training.

Additionally, in our commitment to open research, we plan to release both the code and the complete methodology alongside the detailed paper at `https://github.com/NousResearch/DisTrO`.

2 1.2B LLM Pre-training

Figure 1: Training loss comparison between a pre-trained 1.2B LLM using All-Reduce versus DisTrO trained to 105B tokens for 25000 steps.



For training, we used the Nanotron [6] pre-training framework and only operate under the Distributed Data Parallelism strategy, where each GPU has the entire model loaded in VRAM. In this scenario, every GPU can be considered to be an accelerator node. Since it is widely adopted within the open source LLM community, we used the Llama 2 [18] LLM architecture, scaled down to a 1.2B parameter size. Model and training hyper-parameters are given in Table 1. For training data we used the first 105B tokens of a randomly shuffled 10% representative sampling of the Dolma v1.7 [17][5] dataset. For the optimizer, we used AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.95$ and a peak learning rate of 4×10^{-4} decayed with a cosine decay schedule. Weight decay was set to 0.1.

For the DisTrO training run, we swapped out AdamW with DisTrO-AdamW without changing hyper-parameters and disabled every instance of the All-Reduce operation in Nanotron. Unlike previous distributed training methods, DisTrO does not synchronize optimizer states (and has a stateless client variant, but the method tested here is stateful). In

[3] By clock-synchronous, we mean that DisTrO is similar to SGD, Adam, etc., where each training step uses the same arithmetic operations and takes the same amount of time.

[4] In contrast to sparse neural networks such as mixture-of-experts (MoEs).

[5] `https://huggingface.co/datasets/emozilla/dolma-v1_7-305B`

Figure 1 and Table 3, we show empirical proof that DisTrO matches standard training in loss while massively reducing the inter-node bandwidth required.

Table 1: Model hyper-parameters

	1.2B
Layers	16
Hidden Size	2048
Intermediate Size	8192
Attention Heads	8
Key/Value Heads	8
Activation Function	SwiGLU
Vocab Size	32000
Positional Embeddings	RoPE ($\theta = 10000$)
Peak Learning Rate	4×10^{-4}
Decay Learning Rate	4×10^{-5}
Decay Schedule	Cosine
Warmup Steps	400
Total Steps	25000
Sequence Length	2048
Batch Size	2048 (4M tokens)
Total Tokens	104.8576B

For validation, we performed the GPT4All ([2] [19] [23] [13] [1] [15] [14]) zero-shot benchmarks on our trained models. We also compared against a checkpoint[6] of TinyLlama [24] trained on the same number of tokens. We chose TinyLlama as a measure for sanity checking our results given that its architecture and training process was very similar to ours. The results are shown in Table 2 and further confirms that DisTrO matches industry-standard training.

Table 2: Evaluation score comparison between pre-trained LLMs using All-Reduce versus DisTrO trained to 105B tokens, with an intermediate TinyLlama checkpoint used as a sanity check

Model	Model Size	Tokens	ARC-c ↑	ARC-easy ↑	Hellaswag ↑
All-Reduce + AdamW	1.2B	105B	24.5	**47.6**	43.3
DisTrO + AdamW	1.2B	105B	**24.9**	46.2	**47.3**
TinyLlama	1.1B	105B	24.2	44.9	43.6
			OpenBookQA ↑	PIQA ↑	WinoGrande ↑
All-Reduce + AdamW	1.2B	105B	29.0	69.5	51.6
DisTrO + AdamW	1.2B	105B	**30.6**	**71.7**	**54.1**
TinyLlama	1.1B	105B	30.0	67.5	53.7

Table 3: Training performance comparison between a pre-trained 1.2B LLMs using All-Reduce versus DisTrO trained to 105B tokens

Model	GPUs	Wall Time ↓	Receive / Step ↓	Reduction ↑	Final Loss ↓
All-Reduce + AdamW	32xH100	**17.1 Hours**	74.4 GB	1x	2.449
DisTrO + AdamW	32xH100	19.8 Hours	**86.8 MB**	**857x**	**2.373**

3 Discussion

While this initial training run shows a 857x reduction of bandwidth requirements when using DisTrO-AdamW as a drop-in replacement to AdamW+All-Reduce, our preliminary tests indicate that it is possible to get a bandwidth requirements reduction of up to 1000x to 3000x during the pre-training of a 1.2B LLM if hyper-parameters are chosen

[6] `https://huggingface.co/TinyLlama/TinyLlama-1.1B-step-50K-105b`

carefully. For post-training and fine-tuning, we can achieve up to 10000x without any noticeable degradation in loss. More experiments are on-going.

3.1 Data Streaming

In a naive peer-to-peer scenario with 32 nodes, a 857x reduction in bandwidth means that at every step, each node has to transmit on average $2.8\text{MB} \times 31 = 86.8\text{MB}$, and receive the same amount of data, as shown in Table 3. However, if there are dedicated servers for data aggregation, one can get away with only uploading 2.8MB of data per step, while receiving 86.8MB. A hybrid approach might also work. This asymmetry is advantageous since most consumer internet bandwidth is heavily skewed towards bigger download speeds. Assuming a stable internet speed of 100Mbps download and 10Mbps upload, the worst-case latency is only 6.94s for downloading and 2.24s for uploading, creating a total network latency of 6.94s per step. Future research into compression could further improve latency, as we currently do not use any compression and transmit the tensors directly.

3.2 Bandwidth Scaling

We do not yet know whether the ratio of bandwidth reduction scales up, down or stays constant as model size increases. We hypothesize that the bandwidth reduction may indeed increase (i.e. relatively less and less communication is needed as model size grows). We base this on the observation that the 1.2B model seems to be the smallest size that we found that DisTrO works with consistently. Smaller models[7] do not converge as well with DisTrO. Further research into this area should tell us how this bandwidth reduction scales with respect to an increasing model size.

In addition, given that the amount of data between accelerators is decoupled from the model size, a new possible scaling law arises: one where the model size is increased without increasing the communication bandwidth. It would be interesting to see whether a larger model improves loss and learning even when bandwidth stays constant[8]. From our very early experiments, we speculate that this might be the case. If true, it would be possible that we see a future paradigm shift into designing and manufacturing bigger GPUs with larger VRAM and narrower interconnects, where we favor compute-heavy workloads over I/O-heavy operations, since scaling up the bandwidth of GPU interconnections incurs higher costs compared to adding more compute within a GPU.

3.3 Future Applications and Implications

While we have only demonstrated DisTrO's convergence specifically for LLM training, there are strong indications that DisTrO can also be used for training other large foundational models such as Diffusion Models. Lastly, we will consider potential applications of DisTrO and suggest possible implications for the wider AI ecosystem. We caution that such prognostications are largely aspirational and serve merely to spark the imagination of what may be possible.

3.3.1 Federated Learning

Federated learning [12] is a sub-field of machine learning that promises to allow collaborative training of models while keeping each participant's data private and decentralized. This field has seen a recent surge in popularity due to concerns about how data privacy has almost gone extinct in the foundational LLM space, given the training requirements of such models. However, until now, federated learning was held back by a lack of efficient and compromise-free methods of training large models over the limited bandwidth of the internet.

While we do not focus on the data privacy aspect of training, DisTrO imposes no requirements on how the data is processed or assigned to individual GPU nodes, and has a stateless variant (similar to Federated Averaging), thus can be adapted for federated learning in the future. For the first time, we see the practical feasibility of applying federated learning to efficiently train frontier LLMs over the internet.

3.3.2 Decentralized Training

Furthermore, DisTrO enables the creation of a fully decentralized and permissionless network to collaborate and pool resources. Evidence suggests that when implemented, DisTrO is remarkably resilient to a small amount of degraded[9] or dropped nodes during training, and it can be easily made to accommodate new nodes joining.

[7] We have tested 50M and 300M model sizes pre-trained for 10B tokens.

[8] For example, increasing the model size from 1.2B to 12B while keeping the amount of data transmitted at a constant 100MB per step.

[9] A degraded node is a node that returns incorrect information during training due to software or hardware issues.

Cryptographic primitives and incentive structures could be layered on to recruit participants and mitigate risks of non-trusted nodes damaging a run using adversarial attacks. Of course, the introduction of such mechanisms creates overhead on the system, and the balancing of these levers remains a topic for exploration. However, if done correctly, the flexible nature of the design could unlock latent computing resources by minimizing the opportunity cost of contributing to a training run, which in turn, economically incentivizes both institutions and individuals to contribute their compute.

As the DisTrO optimizer promises to activate latent compute resources, it could also activate thought-to-be-obsolete training hardware for training runs (assuming certain advancements happen). As of now, consumer-grade or older generations of accelerators with smaller amounts of memory[10] may be used to train smaller sized models, but they cannot be easily used on larger models. Adapting strategies like FSDP [26], SWARM Parallelism [14] and/or [22] to work in tandem with DisTrO to accommodate hardware with smaller amounts of memory would unlock a swath of untapped resources. This could be one major step forward in creating the first virtual and fully heterogeneous GPU cluster.

3.3.3 Environmental Impacts

A further application of DisTrO at scale may be to relieve some of the issues related to energy consumption, infrastructure costs and land use that massive data centers create. The Llama 3.1 project required the construction of two massive monolithic super-clusters of 24,000 H100 GPUs each [9], and produced around 11,000 metric tons of CO_2 equivalent [4] just for the training process, without accounting for indirect CO_2 equivalent emissions from auxiliary processes. Newer models, even of the same parameter size, will consume much more resources for training due to their ever-growing data corpus [21].

Today, AI related data centers have tested the limits of modern power grids (for example, Ireland's EirGrid has had to reconsider their ability to host new constructions [5]). This energy strain makes renewable power sources less likely to meet a grid's needs, as clean energy can be more susceptible to fluctuations in production. DisTrO can be used to adaptively balance multiple smaller modular data centers that use excess capacity, similar to how Bitcoin mining operations have created infrastructure to make efficient use of surging power grids and underutilized green energy [7]. Tapping into pre-existing infrastructure with a dynamically balanced training technique can lessen the negative impacts of training on the environment.

3.4 Closing Remarks

By breaking down barriers of centralized compute and reducing inter-GPU communication requirements, DisTrO may open up opportunities for widespread participation and collaboration on global AI projects. This shift could not only democratize access to cutting-edge AI technology, but also enhance resiliency by distributing the computational load across multiple nodes, minimizing single points of failure.

As we move forward into this era of decentralized AI, it is crucial to recognize that decentralization is a force for good, promoting transparency, accountability, and ultimately, greater innovation. By harnessing the power of idle compute resources worldwide and fostering open-source collaboration, we can unlock unprecedented advancements in AI research, development, and deployment.

We invite researchers interested in exploring these to areas to join us in our quest.

References

[1] Y. Bisk, R. Zellers, R. Le bras, J. Gao, and Y. Choi. Piqa: Reasoning about physical commonsense in natural language. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7432–7439, Apr. 2020. doi: 10.1609/aaai.v34i05.6239. URL https://ojs.aaai.org/index.php/AAAI/article/view/6239.

[2] P. Clark, I. Cowhey, O. Etzioni, T. Khot, A. Sabharwal, C. Schoenick, and O. Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge, 2018. URL https://arxiv.org/abs/1803.05457.

[3] A. Douillard, Q. Feng, A. A. Rusu, R. Chhaparia, Y. Donchev, A. Kuncoro, M. Ranzato, A. Szlam, and J. Shen. Diloco: Distributed low-communication training of language models, 2023. URL https://arxiv.org/abs/2311.08105.

[4] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, A. Goyal, A. Hartshorn, A. Yang, A. Mitra, A. Sravankumar, A. Korenev, A. Hinsvark, A. Rao, A. Zhang, A. Rodriguez, A. Gregerson, A. Spataru, B. Roziere, B. Biron, B. Tang, B. Chern, C. Caucheteux, C. Nayak,

[10] Such as RTX 3090/4090, A40 and A100-40GB GPUs

C. Bi, C. Marra, C. McConnell, C. Keller, C. Touret, C. Wu, C. Wong, C. C. Ferrer, C. Nikolaidis, D. Allonsius, D. Song, D. Pintz, D. Livshits, D. Esiobu, D. Choudhary, D. Mahajan, D. Garcia-Olano, D. Perino, D. Hupkes, E. Lakomkin, E. AlBadawy, E. Lobanova, E. Dinan, E. M. Smith, F. Radenovic, F. Zhang, G. Synnaeve, G. Lee, G. L. Anderson, G. Nail, G. Mialon, G. Pang, G. Cucurell, H. Nguyen, H. Korevaar, H. Xu, H. Touvron, I. Zarov, I. A. Ibarra, I. Kloumann, I. Misra, I. Evtimov, J. Copet, J. Lee, J. Geffert, J. Vranes, J. Park, J. Mahadeokar, et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

[5] R. Galvin. Data centers are pushing ireland's electric grid to the brink, Dec 2021. URL https://gizmodo.com/data-centers-are-pushing-ireland-s-electric-grid-to-the-1848282390.

[6] Huggingface. nanotron: Minimalistic large language model 3d-parallelism training. URL https://github.com/huggingface/nanotron.

[7] J. I. Ibañez and A. Freier. Bitcoin's carbon footprint revisited: Proof of work mining for renewable energy expansion, 2023. URL https://arxiv.org/abs/2304.04578.

[8] J. Konečný, B. McMahan, and D. Ramage. Federated optimization:distributed optimization beyond the datacenter, 2015. URL https://arxiv.org/abs/1511.03575.

[9] K. Lee, K. Lee, A. Gangidi, and M. Oldham. Building meta's genai infrastructure, May 2024. URL https://engineering.fb.com/2024/03/12/data-center-engineering/building-metas-genai-infrastructure/.

[10] S. Li, Y. Zhao, R. Varma, O. Salpekar, P. Noordhuis, T. Li, A. Paszke, J. Smith, B. Vaughan, P. Damania, and S. Chintala. Pytorch distributed: experiences on accelerating data parallel training. *Proc. VLDB Endow.*, 13(12): 3005–3018, Aug 2020. ISSN 2150-8097.

[11] Y. Lin, S. Han, H. Mao, Y. Wang, and B. Dally. Deep gradient compression: Reducing the communication bandwidth for distributed training. In *International Conference on Learning Representations*, 2018.

[12] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data, 2023. URL https://arxiv.org/abs/1602.05629.

[13] T. Mihaylov, P. Clark, T. Khot, and A. Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1260. URL https://aclanthology.org/D18-1260.

[14] M. Ryabinin, T. Dettmers, M. Diskin, and A. Borzunov. Swarm parallelism: Training large models can be surprisingly communication-efficient, 2023. URL https://arxiv.org/abs/2301.11913.

[15] K. Sakaguchi, R. L. Bras, C. Bhagavatula, and Y. Choi. Winogrande: an adversarial winograd schema challenge at scale. *Commun. ACM*, 64(9):99–106, aug 2021. ISSN 0001-0782. doi: 10.1145/3474381. URL https://doi.org/10.1145/3474381.

[16] S. Shi, X. Pan, Q. Wang, C. Liu, X. Ren, Z. Hu, Y. Yang, B. Li, and X. Chu. Schemoe: An extensible mixture-of-experts distributed training system with tasks scheduling. In *Proceedings of the Nineteenth European Conference on Computer Systems*, EuroSys '24, page 236–249, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400704376. doi: 10.1145/3627703.3650083. URL https://doi.org/10.1145/3627703.3650083.

[17] L. Soldaini, R. Kinney, A. Bhagia, D. Schwenk, D. Atkinson, R. Authur, B. Bogin, K. Chandu, J. Dumas, Y. Elazar, V. Hofmann, A. H. Jha, S. Kumar, L. Lucy, X. Lyu, N. Lambert, I. Magnusson, J. Morrison, N. Muennighoff, A. Naik, C. Nam, M. E. Peters, A. Ravichander, K. Richardson, Z. Shen, E. Strubell, N. Subramani, O. Tafjord, P. Walsh, L. Zettlemoyer, N. A. Smith, H. Hajishirzi, I. Beltagy, D. Groeneveld, J. Dodge, and K. Lo. Dolma: an Open Corpus of Three Trillion Tokens for Language Model Pretraining Research. *arXiv preprint*, 2024.

[18] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, and T. Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

[19] A. Wang, Y. Pruksachatkun, N. Nangia, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing*

Systems, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/4496bf24afe7fab6f046bf4923da8de6-Paper.pdf.

[20] J. Wang, Y. Lu, B. Yuan, B. Chen, P. Liang, C. De Sa, C. Re, and C. Zhang. CocktailSGD: Fine-tuning foundation models over 500Mbps networks. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 36058–36076. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/wang23t.html.

[21] C.-J. Wu, B. Acun, R. Raghavendra, and K. Hazelwood. Beyond efficiency: Scaling ai sustainably, 2024. URL https://arxiv.org/abs/2406.05303.

[22] B. Yuan, Y. He, J. Davis, T. Zhang, T. Dao, B. Chen, P. S. Liang, C. Re, and C. Zhang. Decentralized training of foundation models in heterogeneous environments. *Advances in Neural Information Processing Systems*, 35: 25464–25477, 2022.

[23] R. Zellers, A. Holtzman, Y. Bisk, A. Farhadi, and Y. Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472.

[24] P. Zhang, G. Zeng, T. Wang, and W. Lu. Tinyllama: An open-source small language model, 2024.

[25] J. Zhao, Z. Zhang, B. Chen, Z. Wang, A. Anandkumar, and Y. Tian. Galore: Memory-efficient llm training by gradient low-rank projection, 2024.

[26] Y. Zhao, A. Gu, R. Varma, L. Luo, C.-C. Huang, M. Xu, L. Wright, H. Shojanazeri, M. Ott, S. Shleifer, A. Desmaison, C. Balioglu, B. Nguyen, G. Chauhan, Y. Hao, and S. Li. PyTorch FSDP: Experiences on scaling fully sharded data parallel, 2023. arXiv: 2304.11277.