Does Distributed Training Undermine Compute Governance?

Robi Rahman [1]

Abstract

Compute governance proposals often rely on the assumption that frontier AI training requires large, detectable computing clusters. However, recent advances in distributed training algorithms could allow developers to conduct frontier-scale training on distributed agglomerations of hardware, rather than needing large datacenter facilities. Developers who prefer not to be constrained by regulations may structure their hardware in a manner that evades the registration and monitoring requirements associated with compute governance. Therefore, regulations must be designed to detect and prevent illicit distributed training operations. This paper evaluates the feasibility of such evasion and outlines recommended countermeasures, including whistleblowing, chip tracking, forensic accounting, and memory and compute thresholds for clusters.

1. Introduction

Many existing and proposed compute governance measures rely on the governing body's awareness of large agglomerations of computing hardware, known as compute clusters or supercomputers (Pilz et al., 2025). Training frontier models requires very large amounts of computing power, and frontier model developers prefer to concentrate the requisite hardware into large clusters. This configuration enables all the constituent chips to transfer data to the others over high-bandwidth interconnect. It also allows personnel to perform maintenance on all the hardware in a single location, and enables chips to share memory, which improves resilience to hardware malfunctions and reduces the requisite frequency of checkpointing (Erben & Erdil, 2024).

Many regulations based on compute have been proposed, and several have been enacted. US Executive Order 14110, in effect from October 2023 to January 2025, previously required developers conducting training runs larger than 10^{26} FLOP (or larger than 10^{23} FLOP if training primarily on biological sequence data) to report them to the government and disclose their cybersecurity and risk management practices (White House, 2023). Articles 51 and 55 of the EU AI Act require general-purpose AI models trained with more than 10^{25} FLOP to undergo safety evaluations and red-teaming (European Parliament and Council, 2024). California SB 53 regulates developers who train models with more than 10^{26} FLOP, requiring transparency and incident reporting, prohibiting false or misleading statements about catastrophic risk posed by the models, and establishing penalties of up to $1 million for violations (State of California, 2025). Scher et al. (2025) propose an international ban on model training above 10^{24} FLOP and fine-tuning above 10^{23} FLOP. Baker et al. (2025) recommend measures to verify compliance with requirements such as compute usage limits, auditing of large training runs, and prohibitions on certain categories of workloads.

Enforcement of these regulations would be rendered ineffective if the regulated developers could hide their computing hardware. Frontier-scale datacenters cannot be hidden because their power consumption and physical footprints are too large, so they can be tracked with power grid monitoring and satellite imagery (Epoch AI, 2026b). However, recent advancements in distributed computing methods for ML model training have increased the feasibility of harnessing diffuse collections of hardware to perform large-scale training runs. The *DiLoCo* (distributed low-communication training) family of algorithms compresses the gradients that are transferred between processors during training, theoretically allowing large-scale model training to be done with less than 100 Mbps of bandwidth, thousands of times less than the 400+ Gbps connections that are standard in modern datacenters (IEEE, 2017). If a frontier-scale cluster were split into nodes[1] smaller than the regulator's monitoring threshold, it would be undetectable by thermal and electrical means, obligating enforcement to rely on network monitoring and in-person observations. Furthermore, once a node is identified, it is not certain that the regulator would be able to prove that it is part of a distributed training operation.

[1]Machine Intelligence Research Institute. Correspondence to: Robi Rahman <robi@intelligence.org>.

[1]A node refers to a single, self-contained unit of computing hardware that is part of a larger, geographically dispersed network. In the scenarios modeled, we investigate the feasibility of training with nodes smaller than the hardware reporting threshold proposed in Scher et al. (2025).

This work develops a comprehensive efficiency model for distributed training based on the state of the art as of 2026, evaluates the feasibility of large-scale training runs using a variety of types and sizes of processors, and assesses the efficacy of countermeasures for detecting regulatory evasion. We recommend that regulations governing models be accompanied by registration requirements for clusters exceeding compute throughput and/or accelerator memory thresholds, in order to prevent evasion. An interactive simulator is published online[2] to assist policymakers and governance researchers in both understanding the problem and designing standards that are robust to such evasion.

2. Background and Related Work

Stich (2019) introduced *local SGD*, a variant of stochastic gradient descent that maintains copies of a machine learning model on multiple different processors, training each copy independently for H steps before averaging parameters, which reduces communication by a factor of H. Douillard et al. (2023) introduced *DiLoCo*, a low-communication variant of local SGD where each replica[3] computes pseudo-gradients on its copy of the weights and data for H steps, then the pseudo-gradients are averaged across the replicas in an all-reduce operation. Douillard et al. (2025) further optimized the algorithm to yield *Streaming DiLoCo*, wherein subsets of the model parameters are synchronized in sequence, rather than all at once, and workers continue training during the parameter synchronization. By overlapping communication with compute, the system becomes compute-bound when H, the number of local steps between synchronizations, is sufficiently large. These advances progressively reduced the bandwidth required for distributed training. This research is extended in Decoupled DiLoCo (Douillard et al., 2026), an asynchronous variant that tolerates partial hardware failures and heterogeneous chips within a single run.

Erdil & Schneider-Joseph (2024) theoretically model the effectiveness and limitations of large-scale model training subject to data movement bottlenecks, and Erdil & Besiroglu (2024) published an interactive simulator illustrating the model's prediction of training performance depending on the user's input. This work builds on theirs to simulate the case of multiple sites, and accommodates lower bandwidth and greater latency.

Kryś et al. (2025) provided a taxonomy of governance risks from distributed and decentralized training, including proliferation of dangerous capabilities, decreased shutdownability of misaligned or misused models, and lack of oversight. However, they did not assess the feasibility of frontier-scale distributed training with small nodes, nor evaluate the effectiveness of countermeasures, which are analyzed below.

Scher et al. (2025) proposed an international agreement that, if enacted, would restrict AI training in countries party to the agreement. The proposal would ban model pre-training over 10^{24} FLOP and fine-tuning over 10^{23} FLOP, with model training between 10^{22} FLOP and 10^{24} FLOP allowed but monitored. It would also require all clusters of chips with more computing power than the equivalent of 16 H100 GPUs to be registered and monitored. The proposal explicitly intends to make it difficult to perform distributed training across multiple sub-threshold sets of chips. However, recent advancements in distributed training algorithms enable exactly this type of violation: models larger than their banned threshold can be trained in a relatively short timeframe using nodes with processing rates of 16 H100-equivalents or less, as demonstrated herein. This paper proposes an amendment to Scher et al. (2025)'s cluster registration requirements that makes governance evasion much more difficult and detectable.

Sevilla & Troynikov (2025) assessed the feasibility of synchronizing frontier-scale datacenters to create even larger supercomputers, in order to enable larger training runs than those that can be done using only one datacenter. They concluded that this is technically feasible and would allow developers to bypass power grid limitations, illustrating an example of 23 large datacenters connected with 4,800 km of fiber optic cables and a total power draw of 10 GW. Finally, Sevilla (2025) conducted a literature review of distributed training techniques and collected a dataset of models trained using these techniques, then used the data to forecast future scaling of distributed training runs. This paper extends Sevilla and Troynikov's work to cases of smaller nodes, more replicas, more local steps, lower bandwidth, and greater latency.

3. Methodology

3.1. Threat Model

Suppose a governing body regulates and requires reporting of model training above a compute threshold, and reporting of compute clusters above a performance threshold. Then a developer can avoid complying with the model development regulations by using clusters of hardware below the performance threshold and not reporting their model. The scenarios analyzed here involve a well-resourced AI developer attempting this strategy.

We investigate the feasibility of developing models similar to Llama 3.1-405B (Meta AI, 2024) (the largest pub-

[2]Simulator available at https://intelligence.org/research/distributed-training-simulator; source code available at https://github.com/robirahman/miri-decentralized-training-report.

[3]A replica refers to a node, or pipeline-parallel set of nodes, containing a copy of model weights for DiLoCo training.

lished open-source model by training compute (Rahman et al., 2025; Epoch AI, 2026a), and above most regulatory compute thresholds) under very difficult conditions: consumer-grade internet with 100 Mbps bandwidth and 100 ms latency—thousands of times slower than inter-server connections in datacenters—and a limit of only 16 H100-equivalents of compute per node, as in Scher et al. (2025), the most restrictive compute governance proposal in the literature. Furthermore, we assume the governing body is actively searching for any signs of illicit distributed training, by monitoring internet traffic, power and water usage, satellite imagery, and thermal radiation, as well as conducting physical inspections of suspected undisclosed computing facilities. We assume the time available for training any model is limited to two years or less; our conclusions are robust to alternative assumptions; see Appendix A for justification and analysis.

Pretraining is the most communication-heavy and compute-intensive phase of model development; fine-tuning and RL are less difficult to adapt to a distributed hardware configuration, so we conservatively focus on pretraining because it is the most challenging phase for the evader (Kryś et al., 2025). If the evader succeeds at pretraining a large model, it is relatively easy to fine-tune it to improve its reasoning, coding, and other capabilities.

3.2. Hardware Selection

The evader selects the most useful hardware that can be purchased in sufficient quantities and performs the specified training workload under the above constraints, in particular, the limit of 16 H100-equivalent compute per node. In this case, the optimal hardware is often the NVIDIA A100 80GB, which is inexpensive, exists in large quantities, and has a very large amount of accelerator memory relative to its computing performance. The NVIDIA GH200 is also an effective choice, with superior FLOPS/$, support for FP8 numeric format, and up to 144 GB of high-bandwidth memory per card. Other hardware options, including Google and Huawei chips, are compared in Appendix B. Recent work (Ryabinin et al., 2023; Douillard et al., 2026) demonstrates that distributed training can mix hardware types within a training run without degrading ML performance, so in practice, evaders are not constrained to a single hardware type; however, in the scenarios modeled, heterogeneous hardware would not outperform uniform hardware.

3.3. Efficiency Model

The most important metrics calculated by the simulator for any training configuration are C_{local}, the local-equivalent compute, and C_{quality}, the quality-adjusted compute:

$$C_{\text{local}} = C_{\text{throughput}} \times \eta, \tag{1}$$

where $C_{\text{throughput}}$ is the nominal compute throughput, the total number of operations done by the processors and applied to model training,[4] and η is the distributed training inefficiency factor, a ratio between 0 and 1 that accounts for the relative inferiority of distributing hardware across multiple nodes and communicating over relatively slow internet connections, rather than aggregating the hardware in a single cluster where data can be transferred at hundreds or thousands of gigabits per second. Local-equivalent compute is calculated as nominal compute throughput times the distributed training inefficiency factor.

$$C_{\text{quality}} = C_{\text{local}} \times \chi. \tag{2}$$

C_{quality} is quality-adjusted, local-equivalent FLOPs—that is, the amount of centralized, optimally allocated compute that would produce the same quality model. χ is a penalty applied to models that are over- or under-trained relative to Chinchilla optimality, explained below.

The inefficiency factor η consists of three separate factors, or four if pipeline parallelism is enabled:

$$\eta = \eta_H \times \eta_{\text{comp}} \times \eta_{\text{rep}} \times \eta_{\text{act}}. \tag{3}$$

Sync interval penalty (η_H) is the decrease in efficiency from having replicas do many inner steps before synchronizing with each other, resulting in replicas drifting apart and computing stale gradients. If $H = 1$, then the distributed training operation is simply doing data-parallel training with no local SGD, so there is no penalty. For higher values of H, η_H decreases logarithmically, with scaling coefficients calibrated to experiments published by Stich (2019), Douillard et al. (2023), and Charles et al. (2025). This is a small factor in η, causing around 2–10% loss in effective compute for the training runs modeled in this work. It is modeled as $\eta_H = 1 - \alpha \log_{10}(H)$, with α decreasing with model size, per Charles et al. (2025). For example, a model with 250B parameters has $\alpha \approx 0.05$; at $H = 50$, $\eta_H \approx 0.91$.

Compression quality (η_{comp}) accounts for the loss in compute efficiency from compressing gradients to save on bandwidth. Compressing the gradients reduces the amount of communication required but makes the transmitted gradient slightly inaccurate. This can be mitigated by error feedback methods, such as error feedback accumulation, wherein the difference between the gradient calculated locally and the gradient sent with compression is stored and added onto the next update. The SparseLoCo algorithm, implemented in Covenant-72B, is a prominent example of this. Based on Covenant-72B's published results (Lidin et al., 2026), the simulator's default compression setting is $150\times$, with a corresponding $\eta_{\text{comp}} = 0.99$.

[4] Compute throughput is calculated using the standard formula $C = 6ND$.

Replica divergence (η_{rep}) is the loss in compute efficiency from averaging gradients collected from replicas trained on different subsets of data and exploring different areas of the loss landscape. In general, averaging gradients from many replicas is not optimal: for example, if two replicas descend down two parallel valleys in the loss landscape with a hill between them, the average of their individual gradients may place the models on the hill. Based on Charles et al. (2025), this penalty scales inversely with model size: the more high-dimensional the loss surface, the more similar the gradient paths experienced by all replicas. This is the dominant factor in η for most training runs modeled here, with η_{rep} ranging between 0.15 and 0.90.

3.4. Chinchilla Quality Adjustment

Distributed training can sometimes face other constraints not present in single-cluster training—most crucially, limited accelerator memory—and may therefore structure models suboptimally relative to what would be done locally. We define quality-adjusted compute C_{quality} as the local-equivalent compute times a Chinchilla-suboptimality penalty (χ), accounting for the fact that a Chinchilla-optimal model would achieve equal quality with a smaller training compute budget (Hoffmann et al., 2022; Besiroglu et al., 2024).

Since single nodes have much less memory available than the hardware in total, the maximum model size that can fit into memory during training or inference is much smaller than if training were done in a single cluster. This causes distributed training operations to tend to over-train their models. Some amount of overtraining is desirable (Erdil, 2024), but too much results in lower training compute efficiency.

3.5. Pipeline-parallel DiLoCo

The activation quality penalty (η_{act}) is the loss in compute efficiency from compressing activations from the model's hidden layers when using pipeline parallelism to shard the model across nodes.

At large node counts, flat DiLoCo severely overtrains small models. Therefore, it becomes worthwhile to use pipeline parallelism across nodes, using groups of nodes to hold a larger model than can fit on any one of them by itself. This incurs an efficiency penalty because activations must be compressed and transmitted between nodes over the internet, but at very large compute scales, this produces a greater amount of Chinchilla-optimal-equivalent compute. To simulate the effects of activation compression and pipeline bubbles, the simulator introduces the factor η_{act} within η when the hardware configuration scenario is set to use PP groups. Otherwise, this factor is not applied, or equivalently, $\eta_{\text{act}} = 1$ in the equation for η above.

4. Results

The simulator's predicted feasibility of training models at larger scales, based on the scaling behavior derived from the published ML literature, is shown in Table 1. The following compute thresholds could be exceeded by distributed training operations, even under the hardware restrictions proposed in Scher et al. (2025), within the allotted time limit.

Based on the compute thresholds of 10^{24} FLOP in Scher et al. (2025), 10^{25} FLOP in the EU AI Act, and 10^{26} FLOP in California SB 53, these proposed or existing regulations could be evaded using \$1.6M, \$31M, and \$3.8B, respectively, of hardware arranged in clusters smaller than any proposed registration requirement. This poses a severe challenge for regulation and governance, which cannot be applied to models and hardware when the governing body cannot reliably know that they exist. This also makes it difficult to stop the proliferation of dangerous capabilities that emerge at this scale. 101 sub-monitoring nodes reach the local-equivalent compute of GPT-4, and 625 nodes reach the training compute of Llama 3.1-405B.

Above that level, hardware requirements increase sharply due to decreasing efficiency at high replica counts, though the increase is much more modest if larger nodes can be used.

Appendix C illustrates the cost to achieve 10^{25} FLOP of local-equivalent compute with different amounts of available network bandwidth. At 100 Mbps bandwidth, it costs about 3× more to do training distributed relative to centralized, but it is not a substantial barrier to feasibility. Latency has no significant effect: transmission times at DiLoCo's sync volumes are greater than round-trip time by several orders of magnitude.

Target	Node Config	Nodes	Mode	Model	H	η	C_{local}	χ	C_{quality}	OT	Cost
10^{24}	16× H100 FP8	2	Flat	91B	18	0.7957	1.3×10^{24}	0.9796	1.3×10^{24}	1.3×	\$1.6M
DeepSeek-V3	16× GH200 FP8	7	Flat	160B	19	0.5973	3.4×10^{24}	0.9635	3.3×10^{24}	1.4×	\$6.3M
10^{25}	16× GH200 FP8	34	Flat	160B	19	0.3698	1.0×10^{25}	0.6250	6.4×10^{24}	7.0×	\$30.7M
GPT-4	16× GH200 FP8	101	Hier (8 × 12)	160B	19	0.2580	2.1×10^{25}	0.3689	7.8×10^{24}	21×	\$91.3M
Llama 3.1-405B	50× A100 80GB	625	Hier (10 × 62)	250B	19	0.1524	3.8×10^{25}	0.3214	1.2×10^{25}	26×	\$441.3M
GPT-5	16× H100 FP8	2,880	PP (2 × 1440)	180B	3	0.4244	6.6×10^{25}	0.2901	1.9×10^{25}	31×	\$2.32B
10^{26}	16× H100 FP8	4,706	PP (2 × 2353)	180B	3	0.3934	1.0×10^{26}	0.2123	2.1×10^{25}	51×	\$3.80B

Table 1. Minimum-cost distributed hardware configurations achieving the specified levels of training compute. Columns show the hardware configuration (from node types in Appendix B), model size, inner step count H, compute efficiency η, Chinchilla quality-adjustment factor χ, overtraining ratio (OT) relative to Chinchilla-optimal, and upfront hardware cost. Costs are computed as (price per chip)×(chips)×(chip-to-server factor 1.64×)×(server-to-cluster factor 1.23×), adapted from Cottier et al. (2024). Training modes include flat, hierarchical (Hier), and pipeline-parallel (PP) DiLoCo variants. Training compute of DeepSeek-V3 (3.3×10^{24}) and Llama 3.1-405B (3.8×10^{25}) reported by DeepSeek-AI (2024) and Grattafiori et al. (2024); GPT-4 (2.1×10^{25}) and GPT-5 (6.6×10^{25}) estimated by Epoch AI (2026a).

5. Discussion

5.1. Assumptions

The simulator estimates evader capability using techniques demonstrated in published ML experiments, and extrapolates their performance to larger scales where the developer uses more parameters and compute. The extrapolations are typically pessimistic, assuming worse performance beyond the maximum scale where they have been tested. A full catalog of parameter values is provided in the accompanying documentation and the project repository. Actual distributed training performance may improve, especially with the application of new techniques. Distributed training is an active area of research, with development ongoing in major labs such as Google (Douillard et al., 2026).

The simulator's most recent calibration test was Covenant-72B (Lidin et al., 2026), released after the simulator's initial development. Covenant confirmed compute-bound operation (94.5% time spent computing even without Streaming DiLoCo), and achieved $146\times$ compression with negligible quality loss—nearly an order of magnitude above the simulator's original $16\times$ default. We updated the default to $150\times$ to reflect this result. Covenant matched Llama-2-70B quality with substantially fewer training tokens, validating the simulator's predictions at the 72B scale and exceeding several of its conservative assumptions.

In the opposite direction, three of our modeling choices are generous to the evader. First, we assume the evader has access to enough high-quality pretraining data to train at 10^{26} FLOP and beyond without reusing data across multiple epochs. At the upper end of the configurations we model, this implies tens of trillions of unique tokens, a corpus larger than what the open-web frontier has so far been able to assemble. A data-constrained evader would need to repeat data, which incurs diminishing returns not modeled by the standard Chinchilla scaling law and would moderately reduce model quality at fixed compute. Second, the time-limit derivation assumes the evader is willing to train for years under treaty-restricted growth rates. The treaty's enforcement mechanisms—whistleblower programs, challenge inspections, financial intelligence—create cumulative detection probability that scales with operational duration, pushing evaders toward shorter runs at the cost of total compute. Even at a six-month duration rather than two years, however, the configurations analyzed exceed the relevant compute thresholds by wide margins, so this consideration moderately affects the quality of illicitly developed prohibited models, but not the overall feasibility of developing them. Third, we assume that all workers finish processing their batch and transmit their update at the same time. In reality, random hardware failures and network conditions cause some to report late or drop out, but these effects can be mitigated by techniques demonstrated in past work and do not affect the feasibility of distributed training under realistic network conditions and hardware reliability rates.

Evidence for each assumed value from published ML literature is provided in the accompanying simulator documentation (see Appendix D). Discussion of uncertainties in scaling and how they affect the simulator's predictions is provided in Appendix E, and a review of techniques accounting for stragglers and unreliable workers is provided in Appendix F.

5.2. Evader Strategy

When targeting high quantities of quality-adjusted compute C_{quality}, the simulator penalizes configurations that train small models, because these are overtrained relative to Chinchilla-optimal, and a developer could theoretically create a model with the same loss using a smaller compute budget. However, in practice, developers prefer their models to be moderately overtrained. This is because the amount of inference compute used over the lifetime of any model is generally within one order of magnitude of the upfront training compute (Erdil, 2024), and developers can spend some multiple of extra inference compute to get the same effect as a multiple more training compute (Villalobos & Atkinson, 2023)—if inference compute were much smaller than training compute, the developer would train a smaller model and scale up the compute used during each inference, achieving a cost saving while maintaining the same output quality. Therefore, to optimize the utility of the model over its deployed lifespan while minimizing compute spending, developers tend to deliberately over-train models. Therefore, the simulator's C_{quality} metric is excessively pessimistic for modest overtraining ratios.

In the current world, hardware and software are rapidly improving and AI investments are growing sharply. On the other hand, if AI development were tightly restricted as proposed by Scher et al. (2025), slower growth rates would mean existing hardware, software, and budgets remain at the frontier for longer, and developers could spend a much longer time training a state-of-the-art model. There are two other effects on training time from a highly restrictive governance scenario. First, ongoing enforcement and intelligence operations, as well as whistleblower incentives, are more effective against longer operations, so evaders would prefer to train faster in order to minimize the timeframe in which they can be caught. On the other hand, for any given amount of compute, a bigger operation that trains faster with more hardware and greater throughput will have more sites, visibility, and financial costs than another operation that reaches the same compute more slowly using less hardware. This effect encourages developers to train over longer periods.

The overall effect is that if developers are trying to evade governance, some of them, including those training the most capable models, would likely attempt long-running training operations, up to the time limits identified in Sevilla et al. (2022).

5.3. Countermeasures

Several countermeasures are potentially relevant and could help prevent developers from evading AI governance using distributed training. The most effective measures include chip tracking, whistleblowing programs, memory and compute limits on unregistered hardware, and traditional intelligence and law enforcement techniques. Others, such as capping bandwidth available from AI computing sites and monitoring internet traffic, are likely ineffective because they are too easily evaded by developers with sufficient determination and technical capabilities. A detailed assessment of the efficacy of various countermeasures is provided in Appendix G.

5.4. The Simulator

We release the simulator, with open-source Python backend and an interactive web interface, designed to allow governance researchers to evaluate evasion scenarios under their own assumptions. Configurable parameters include hardware assumptions (GPU type, node count, local MFU), network conditions (bandwidth, latency), DiLoCo configuration (inner steps, compression ratio, and flat, hierarchical, or pipeline-parallel mode), training duration, and model size. The simulator distinguishes raw local-equivalent compute (C_{local}) from quality-adjusted compute (C_{quality}), and supports optimistic, expected, and conservative scenarios for compression-quality factors so that users can examine the sensitivity of any particular conclusion to extrapolation uncertainty. The codebase, scaling-law calibration, literature review of past distributed training experiments, and relevant assumptions are documented in the project repository.

6. Conclusion

Distributed training can produce models above every existing or proposed compute threshold, and as capable as current frontier models, even when using hardware *below* every proposed monitoring threshold. Configurations evaluated herein would violate Scher et al. (2025) for under $2M, the EU AI Act for $31M, and SB 53 for $3.8B worth of unmonitored hardware. This leaves an enforcement gap in current governance methods, if developers become determined to evade them. Currently, the increased costs are economically unpalatable, but if developers encounter regulatory bottlenecks in the future, they may be willing to pay the price. Given the pace of ongoing research, the enforcement gap identified here is actively growing, and methods for detecting it will become even more important.

Existing technical countermeasures are insufficient. However, a combination of compute- and memory-based chip tracking substantially raises evasion costs. Combining chip registries with whistleblower programs, challenge inspections, and conventional intelligence operations reestablishes monitorability of distributed training operations and allows AI governance to continue without loopholes.

Impact Statement

This paper presents work whose goal is to advance the field of technical governance of AI. The work identifies potential failures of governance and investigates solutions in order to mitigate both existential and prosaic risks.

LLM Usage Statement

No AI-generated text is included in the final draft of this paper. LLMs were used for assistance in identifying relevant literature, fitting simulator parameters to published experiments, implementing the simulator scripts, creating the web simulator frontend, fact-checking claims made in this document, and converting it to LaTeX.

Acknowledgements

Thanks to Aaron Scher for extensive feedback, to Jaime Sevilla for past reviews of distributed training that inspired the simulator concept, and to Alex Beck for editing assistance.

A. Training Time Limit

We assume that development for any model is subject to a practical time limit of 740 days, based on the theoretical model published by Epoch in Sevilla et al. (2022) and using model inputs motivated by the compute governance proposals in Scher et al. (2025).

Sevilla et al. (2022)'s model gives an upper bound for training time when accounting for hardware improvements, algorithmic and software improvements, and rising AI investments. The key insight is that, when hardware is improving over time, an overly long training run will be overtaken by a shorter training run that starts later but uses newer, faster hardware. Similarly, if funding is increasing over time, a long training run with a small budget will be overtaken by a shorter training run that starts later but has a larger budget. The formula for maximum training time is then:

$$T = \frac{1}{g_H + g_S + g_I}, \tag{4}$$

where g_H is the rate of improvement of hardware efficiency, g_S is the rate of improvement of software, and g_I is the growth rate of AI development spending.

In the current, unregulated environment, growth rates are $1.37\times$ per year for AI chip FLOPS/$ (Epoch AI, 2026c), $3\times$ per year in pretraining algorithmic efficiency (Ho et al., 2024), and $3.5\times$/year in frontier model training costs (Epoch AI, 2026d), resulting in a training time bound of approximately 4.5 months.

Based on correspondence with the authors of Scher et al. (2025), we model the evader's time constraint based on growth rates of 6% per year for hardware, 50% per year for software, and 3% per year for investment. Their governance proposal is intended to completely stop the advancement of AI algorithms and hardware, but in practice, a partial reduction is more realistic. This slower progress would allow time for AI alignment researchers to improve model safety before the development of transformative, superhuman artificial intelligence, but it also incentivizes evaders to spend more time doing illicit model development. These input values represent a reduction of over 75% in the rate of hardware and software progress, and cut investment scaling all the way down to the background level of economic growth, yielding a time limit of 740 days or approximately two years.

With intermediate levels of regulations, the training time used by evaders would correspondingly fall in between these values. Using the slower growth rates and longer time limit is potentially generous to the evader, but even training times at the lower end of this range are long enough for the evader to successfully train models at near-frontier scale, as illustrated in the Results section above. Implications of long training runs on detectability of the evader by the governing body are addressed in the Discussion section.

B. Hardware Configurations

The simulator offers a variety of node configurations so that researchers can estimate the efficiency and cost of training models on different types of hardware. Presets include standard racks or pods produced by the hardware manufacturers, as well as nodes sized at or below the 16 H100-equivalent hardware reporting threshold proposed in Scher et al. (2025).

All performance figures are in PFLOPS and all memory figures are in GB.

B.1. Standard Commercial Pods

Table 2. NVIDIA preset pods.

Preset	Chips	Perf. (FP16)	Memory
GH200 NVL32	32	31.68	4,608
GB200 NVL72	72	162.0	13,824
DGX H100 SuperPOD	256	253.44	20,480
DGX A100 SuperPOD	1,120	349.44	89,600

Table 3. Huawei preset pod.

Preset	Chips	Perf. (FP16)	Memory
CloudMatrix 384 (910C)	384	230.4	49,152

Table 4. Google preset pods.

Preset	Chips	Perf. (BF16)	Memory
TPU v4 pod	4,096	1,126.4	131,072
TPU v5e pod	256	50.43	4,096
TPU v5p pod	8,960	4,112.64	851,200
TPU v6e pod	256	235.01	8,192

B.2. Nodes Under the 16 H100-Equivalent Threshold

Table 5. Sub-threshold node configurations, 16-bit arithmetic.

Hardware	Chips	Perf. (16-bit)	H100-eq.	Memory	Cost/chip (USD)
50× A100 80GB	50	15.600	15.76	4,000	$7,000
49× Ascend 910B	49	15.680	15.84	3,136	$16,000
26× Ascend 910C	26	15.600	15.76	3,328	$26,000
57× TPU v4	57	15.675	15.83	1,824	$12,000
80× TPU v5e	80	15.760	15.92	1,280	$6,000
34× TPU v5p	34	15.606	15.76	3,230	$20,000
17× TPU v6e	17	15.606	15.76	544	$25,000

Table 6. Sub-threshold node configurations with 8-bit FP support.

Hardware	Chips	Perf. (8-bit)	Perf. (16-bit)	Memory	Cost/chip (USD)
16× H100 SXM	16	31.68	15.84	1,280	$25,000
16× GH200	16	31.68	15.84	2,304	$28,000
17× TPU v6e FP8	17	31.21	15.61	544	$25,000

C. Bandwidth Sensitivity

Table 7 illustrates the cost to achieve 10^{25} FLOP of local-equivalent compute with different amounts of available net-

Table 7. Minimum-cost distributed hardware configurations that achieve 10^{25} FLOP of training compute with different amounts of available bandwidth. As of April 2026, Chinese average bandwidth is 207 Mbps down and 47 Mbps up; US average is 310 Mbps down and 57 Mbps up (Speedtest by Ookla, 2026).

Bandwidth	Node Config	Nodes	Mode	Model	H	η	C_{local}	χ	C_{quality}	OT	Cost
10 Mbps	16× GH200 FP8	3,082	PP (2×1541)	310B	4	0.4234	1.0×10^{25}	0.9465	9.5×10^{24}	1.6×	$2.79B
30 Mbps	50× A100 80GB	168	Hier (10×16)	250B	61	0.1493	1.0×10^{25}	0.6213	6.2×10^{24}	7.0×	$118.6M
China avg	16× GH200 FP8	38	Flat	160B	25	0.3266	1.0×10^{25}	0.5969	6.0×10^{24}	7.8×	$34.3M
US avg	16× GH200 FP8	34	Flat	160B	20	0.3639	1.0×10^{25}	0.6250	6.3×10^{24}	7.0×	$30.7M
100 Mbps	16× GH200 FP8	34	Flat	160B	19	0.3698	1.0×10^{25}	0.6250	6.4×10^{24}	7.0×	$30.7M
300 Mbps	16× H100 FP8	23	Flat	91B	7	0.5391	1.0×10^{25}	0.4507	4.5×10^{24}	15×	$18.6M
1 Gbps	16× H100 FP8	16	Flat	91B	2	0.8160	1.1×10^{25}	0.5370	5.7×10^{24}	10×	$12.9M

work bandwidth. At 100 Mbps bandwidth, it costs about 3× more to do training distributed relative to centralized, but distribution is not a substantial barrier to feasibility. Latency has no significant effect: transmission times at DiLoCo's sync volumes are greater than round-trip time by several orders of magnitude.

D. Simulator and Modeling Documentation

A bibliography of past distributed training research, and comprehensive documentation of the simulator's mechanics, are available on GitHub.[5] The bibliography includes published ML experiments whose data are used to estimate each parameter, and the documentation explains how parameters and outputs are estimated in the simulator backend.

E. Principal Uncertainties

Two structural uncertainties limit our ability to make precise quantitative claims at the upper end of the modeled range. The first is the empirical scaling of distributed training itself. The largest published DiLoCo-based run is Covenant-72B at 72.7B parameters and 4.8×10^{23}—roughly an order of magnitude fewer parameters, and two orders of magnitude less compute, than configurations we expect to be feasible in practice. The simulator's central efficiency factors—the sync-interval penalty coefficient, the replica divergence exponent, the per-boundary activation compression quality factor—are fit to experiments at $\leq$ 16B parameters with ≤ 16 replicas, and our conclusions extrapolate these by a factor of 10–300× in model size and up to 250× in replica count. PP-Group DiLoCo is particularly sensitive: at 8 pipeline stages, based on conservative values for loss degradation from activation compression, it compounds to $\eta_{\text{act}} \approx 0.56$, reducing C_{quality} at 2,000 nodes by 49% relative to the best-guess values estimated from existing experiments. The qualitative finding that distributed training crosses the compute thresholds in existing and proposed compute governance is robust, but there is substantial uncertainty in the precise values when the scale exceeds 10^{26} FLOP.

The second uncertainty relates to the parameters of the Chinchilla scaling law. Our quality-adjusted compute figures depend on the conversion from raw loss degradation to FLOP-equivalent penalty, which is governed by the scaling exponents published in Hoffmann et al. (2022), fit to models with up to 280B parameters and 6×10^{23} of training compute. Crucially, this uncertainty is not directionally conservative: depending on the scaling behavior above this scale, distributed training could be less or more capable than our C_{quality} figures suggest and evasion may be less or more effective. Hardware costs, node counts, and qualitative feasibility assessments are unaffected because they depend on local-equivalent compute throughput rather than on the loss-to-FLOP conversion.

F. Hardware Failures and Straggler Mitigation

The scenarios illustrated in this paper assume that all workers complete their batches at the same time. In reality, processors have some rate of random failure and not all traffic arrives simultaneously. If the entire distributed training network has to wait for every replica to calculate and send in its update, as in a naive single-program, multiple-data implementation, much time can be wasted by a few late or faulty workers. Several techniques exist to mitigate the effects of straggler nodes, such as aggregating and broadcasting the overall gradient update once some fraction, for example 90%, of replicas have returned their pseudo-gradients. The late updates can be added to the next batch. SWARM demonstrates distributed training with poorly connected, unreliable devices, adaptively rebalancing groups to bypass slowdowns from lost nodes (Ryabinin et al., 2023). Decoupled DiLoCo runs learners independently and aggregates once a quorum has been reached, weighting the individual updates based on the number of tokens processed and number of steps completed by each worker (Douillard et al., 2026).

Ultimately, stragglers do not affect the conclusions presented below at realistic rates of hardware failures

[5] Project repository: https://github.com/robirahman/miri-decentralized-training-report.

(Grattafiori et al., 2024) and under typical network conditions. For example, Meta reported 419 hardware failures while training Llama 3.1-405B on 16,384 H100 GPUs over a period of 54 days, which is one hardware failure per 50,000 GPU-hours (Grattafiori et al., 2024). Different assumptions such as unreliable hardware or networking, and mitigations such as aggregating upon reaching a threshold of workers, can be explored using the simulator.

G. Countermeasure Analysis

This section explores potential countermeasures to prevent governance evasion through distributed training, starting with the least viable options and moving on to the most effective measures.

G.1. Bandwidth Caps

As illustrated by Table 7, reducing the evader's bandwidth would substantially hinder the training throughput and increase the cost of evasion, but it is unfortunately infeasible to prevent evaders from getting fast internet connections. When each node contains hundreds of thousands of dollars worth of hardware, the cost of above-average internet is negligible by comparison, and there are too many broadband customers to audit them all. Nodes could be interspersed with residential facilities, and restricting consumer bandwidth to far below current levels would be an extreme, politically unpalatable measure. DiLoCo remains workable even at lower throughput levels, using greater pseudo-gradient compression and inner step counts.

G.2. Internet Traffic Monitoring

Today, the vast majority of internet traffic is encrypted, so any efforts at detecting distributed training by monitoring internet traffic would have to rely on analysis of the volume and routing, not on the content. DiLoCo-based training achieves $> 10^{25}$ FLOP with ≤ 16 H100-eq per node and less than 40 Mbps of traffic. This is substantially less than the average US household internet speed, 310 Mbps down and 57 Mbps up (Speedtest by Ookla, 2026). If the sites have flexibility to go up to the average connection speed, the traffic can be disguised into almost any irregular pattern. Traffic shaping methods include jitter, variable-rate streaming, and intermittent pauses. All of these can be done while keeping the training hardware compute-bound and therefore having no detrimental effects on performance. And by routing through VPNs and parameter servers, they can increase the difficulty of tracing individual nodes to the rest of the training network. More research is needed to determine if any training detection methods are feasible, but the requirements of this problem appear to favor evaders.

G.3. Chip Tracking

A chip registry with mandatory reporting of possession, transfer, and decommissioning was proposed by Fist & Grunewald (2023) in the context of export controls and echoed in Scher et al. (2025) in the context of international cooperation on AI regulation. This would be a potent countermeasure against illicit AI development in general, but is especially effective against distributed training. Since a distributed configuration is less efficient than a centralized cluster, it requires more total hardware, so a registry that tracks a large fraction of existing compute is hard to evade in this way. Baker et al. (2025) extend this framing, combining compute accounting (registry-based ownership tracking) with hardware-enabled mechanisms (HEMs).

Two HEM proposals are particularly relevant. *Delay-based location verification*, proposed by Brass & Aarne (2024), uses cryptographic challenge-response protocols between chips and landmark servers to bound each chip's physical location via speed-of-light delay. If deployed at manufacture on a sufficiently high percentage of AI chips, location verification would preclude distributed training. If the chips reported their locations, the governing body would be able to track down the nodes, and if a large quantity of chips went dark to avoid reporting, this would alert the authorities to an evasion attempt. An evader could attempt to use unregistered or untracked chips, but this would require large-scale smuggling or fabrication. *Offline licensing* (Kulp et al., 2024) requires chips to hold time-limited licenses, periodically renewed with the governing authority, to operate. The companion *Fixed set* mechanism restricts high-bandwidth communication to pre-authorized pods of chips, directly blocking the inter-node communication DiLoCo depends on. However, these mechanisms do not exist on current hardware and cannot be retrofitted without certain preexisting firmware, so they cannot catch evaders using many types of already-deployed chips, a gap that persists for the useful lifetime of the current installed base. The effectiveness of a future registry is also diminished by developments that allow distributed training networks to effectively utilize heterogeneous hardware (Ryabinin et al., 2023; Douillard et al., 2026), which would allow evaders to illicitly supplement new, tracked hardware with older, unregistered hardware to augment the throughput of a training run, pushing it above a compute threshold without the regulator's awareness.

G.4. Whistleblowing Programs

Whistleblower programs are a simple, low-burden countermeasure that is well-suited to the distributed training threat model, because distributed evasion is more personnel-intensive than centralized training. An evader operating hundreds or thousands of nodes to train a frontier-scale model needs a much larger team to install and operate the net-

Table 8. Countermeasures and their assessed effectiveness if implemented, along with qualitative societal burden on parties not conducting distributed training.

Countermeasure	vs. Non-State	vs. State	Burden	Recommended?
Bandwidth caps	High	Low	Extreme	No
ISP traffic monitoring	Very low	Very low	Medium	No
Chip registry with location verification	High	Medium	Low	Yes
Whistleblower programs	High	Low	Very low	Yes
Memory threshold (1,280 GB)	Medium	Medium	Low	Yes
Challenge inspections	Medium	Medium	Low	Yes

work than to build one cluster in a datacenter. Procurement, installation, and ongoing maintenance across hundreds of locations requires a substantial workforce, and every person in that workforce is a potential reporter. Unlike centralized training, which can in principle be conducted by a small team inside a single secure facility, a distributed operation has an attack surface proportional to its node count.

Baker et al. (2025) identify whistleblower programs as one of the six independent verification layers needed for robust AI agreement enforcement, and Scher et al. (2025) explicitly include whistleblowing in their enforcement regime alongside supply-chain tracking and challenge inspections. The design template is well-established: the SEC Whistleblower Program, created under the Dodd-Frank Act, offers reporters 10–30% of any resulting penalty along with anti-retaliation protections, and has recovered billions of dollars since its inception. The bipartisan Stop Stealing Our Chips Act (SSOCA), introduced in both chambers of Congress in 2025 and passed by the Senate as of 2026 (Rocco, 2026), would apply this model directly to AI chip export violations at the Bureau of Industry and Security, with financial incentives and anonymous reporting portals (Grunewald, 2025). A parallel program covering unauthorized training runs, not only chip diversion, would extend this same mechanism to close the governance loophole modeled in this paper.

Whistleblower programs have two properties that make them particularly valuable in combination with the other countermeasures discussed above. First, they require no new hardware or chip design changes and therefore work immediately against the current installed base, unlike location verification and offline licensing. Second, their effectiveness scales with the evader's operational complexity, so memory caps and chip registries that force evaders to operate more nodes also amplify whistleblowing's reach. The main limitation is jurisdictional: programs administered by a single state cannot easily reach insiders at state-actor operations abroad, which is why whistleblowing must be combined with the other layers rather than relied on in isolation.

G.5. Memory Limits

Every cluster-reporting threshold proposed or enacted to date defines cluster size in compute terms alone. Executive Order 14110 and its implementing BIS proposed rule (2023–2024) required reporting of computing clusters with aggregate performance exceeding 10^{20} OP/s connected at over 300 Gbit/s (White House, 2023). Scher et al. (2025) would require registration of any Covered Chip Cluster (CCC) with aggregate effective computing capacity above 16 H100-equivalents (15,840 TFLOP/s), a threshold three orders of magnitude lower than EO 14110 but likewise defined solely by compute throughput.

This shared framing creates an exploitable asymmetry: chips with large accelerator memory relative to their compute, such as the A100 80GB, can hold substantially larger models per node while remaining below any compute-only threshold. A sub-threshold node of 50 A100 80GBs, for example, contains 4,000 GB of HBM—more than three times the memory of the reference 16-H100 node, while having a similar processing rate. As demonstrated in Table 1, such nodes are favorable choices for evaders, precisely because a high memory-to-compute ratio minimizes the communication and overtraining penalties that dominate distributed training efficiency. We therefore recommend that Scher et al. (2025) revise their CCC definition to incorporate an accelerator memory threshold of 1,280 GB HBM, matching the memory of 16 H100s. Clusters should be covered if they exceed the capacity of 16 H100s in either memory or computational throughput.

This amendment forces the evader into one of two inefficient regimes. They can either shrink the model size to fit within the memory of each node—which, if trained to very large compute budgets, results in severe overtraining and a significant quality degradation ($\chi < 1$) relative to a Chinchilla-optimal model—or they can shard models across nodes using pipeline parallelism, which triggers activation-compression inefficiencies ($\eta_{\text{act}} < 1$) and substantially increases the number of nodes required. Under the configurations in Table 1, imposing the memory limit reduces the maximum unregistered per-node model from 240B to

roughly 91B parameters, forces the evader to operate approximately five times as many nodes for equivalent model quality, and raises the cost of illicitly developing a 10^{25} FLOP model by roughly 50%.

The amendment imposes minimal collateral burden. Standard research workstations (4–8 GPUs), DGX-class servers (640 GB), cloud inference instances, and rendering farms using GDDR rather than HBM memory all fall well below the threshold. Only clusters equipped with 4,000 GB of HBM or more—usually 17 or more specialized AI chips—require registration, which are precisely the configurations that are of concern for AI governance. Notably, the memory threshold does not make evasion impossible; it makes evasion *expensive and operationally complex*, which amplifies the effectiveness of the other countermeasures discussed above: a larger node count means more procurement, more personnel, more facilities, and therefore more surface for detection by chip tracking, whistleblower reports, and conventional intelligence operations.

G.6. Conventional Intelligence Operations

The countermeasures discussed above—chip tracking, whistleblowing, and memory-threshold registration—are technical and regulatory instruments specific to compute governance. They are complemented by the same conventional intelligence tools that states routinely apply to other forms of illicit economic activity, and which Baker et al. (2025) include as their sixth verification layer alongside the more AI-specific mechanisms.

Financial and procurement intelligence are very well-suited to uncovering distributed training. The configurations in Table 1 require millions to billions of dollars of AI hardware plus corresponding expenditure on facilities and personnel—roughly $441M to illicitly develop a model at the scale of Llama 3.1-405B under Scher et al. (2025)'s constraints, and several billion at the scale of 10^{26} FLOP. Procurement of this magnitude is easily detectable when done by private entities in market economies. Just as financial structuring, breaking transactions into sub-threshold amounts to evade Bank Secrecy Act reporting, is detectable by anti-money-laundering enforcement, the governing body could audit organizations suspected of making undisclosed compute-related purchases.

Challenge inspections are on-site inspections conducted on short notice, and can be performed at facilities with known or suspected relevance to international nonproliferation treaties. They were conducted bilaterally 18 times per year by the United States and Russia from 2011 to 2022 on each other's nuclear facilities, establishing a clear precedent for their usage in international AI governance (Wasil et al., 2024). They are named by Scher et al. (2025) as a component of governance verification and are particularly effective against distributed training because discovery and physical inspection of even a single node can yield evidence that exposes the broader operation. However, asynchronous training architectures such as Douillard et al. (2026) tolerate the loss of individual nodes or groups without interrupting the run, so inspections should be coordinated across the suspected network rather than executed piecemeal. Human intelligence and national technical means round out the stack. None of these tools individually defeat a determined state-scale evader, but combined with the preceding methods, they restore the ability to monitor and reveal covert adversaries.

References

Baker, M. et al. Verifying international agreements on AI: Six layers of verification for rules on large-scale AI development and deployment, 2025. URL https://arxiv.org/abs/2507.15916.

Besiroglu, T. et al. Chinchilla scaling: A replication attempt, 2024. URL https://arxiv.org/abs/2404.10102.

Brass, A. and Aarne, O. Location verification for AI chips, 2024. URL https://www.iaps.ai/research/location-verification-for-ai-chips.

Charles, Z. et al. Communication-efficient language model training scales reliably and robustly: Scaling laws for DiLoCo, 2025. URL https://arxiv.org/html/2503.09799v1.

Cottier, B. et al. The rising costs of training frontier AI models, 2024. URL https://arxiv.org/abs/2405.21015.

DeepSeek-AI. DeepSeek-V3 technical report, 2024. URL https://arxiv.org/abs/2412.19437.

Douillard, A. et al. DiLoCo: Distributed low-communication training of language models, 2023. URL https://arxiv.org/abs/2311.08105.

Douillard, A. et al. Streaming DiLoCo with overlapping communication: Towards a distributed free lunch, 2025. URL https://arxiv.org/abs/2501.18512.

Douillard, A. et al. Decoupled DiLoCo for resilient distributed pre-training, 2026. URL https://arxiv.org/abs/2604.21428.

Epoch AI. Data on AI models, 2026a. URL https://epoch.ai/data/ai-models. Accessed 21 Apr 2026.

Epoch AI. Frontier data centers, 2026b. URL https://epoch.ai/data/data-centers/.

Epoch AI. Trends in artificial intelligence: Hardware, 2026c. URL https://epoch.ai/trends#hardware.

Epoch AI. Trends in artificial intelligence: Training runs, 2026d. URL https://epoch.ai/trends#training-runs.

Erben, A. and Erdil, E. Hardware failures won't limit AI scaling, 2024. URL https://epoch.ai/blog/hardware-failures-wont-limit-ai-scaling/.

Erdil, E. Optimally allocating compute between inference and training, 2024. URL https://epoch.ai/blog/optimally-allocating-compute-between-inference-and-training.

Erdil, E. and Besiroglu, T. Introducing the distributed training interactive simulator, 2024. URL https://epoch.ai/blog/introducing-the-distributed-training-interactive-simulator/.

Erdil, E. and Schneider-Joseph, D. Data movement limits to frontier model training, 2024. URL https://arxiv.org/abs/2411.01137.

European Parliament and Council. Regulation (EU) 2024/1689 (artificial intelligence act), 2024. URL https://eur-lex.europa.eu/eli/reg/2024/1689/oj. 2024 O.J. (L series, 12 July 2024).

Fist, T. and Grunewald, E. Preventing AI chip smuggling to china, 2023. URL https://www.cnas.org/publications/reports/preventing-ai-chip-smuggling-to-china.

Grattafiori, A. et al. The Llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Grunewald, E. A whistleblower incentive program to enforce U.S. export controls, 2025. URL https://www.lawfaremedia.org/article/a-whistleblower-incentive-program-to-enforce-u.s.-export-controls.

Ho, A. et al. Algorithmic progress in language models, 2024. URL https://arxiv.org/abs/2403.05812.

Hoffmann, J. et al. Training compute-optimal large language models, 2022. URL https://arxiv.org/abs/2203.15556.

IEEE. IEEE standard for ethernet amendment 10: Media access control parameters, physical layers, and management parameters for 200 gb/s and 400 gb/s operation, 2017. URL https://standards.ieee.org/ieee/802.3bs/6748/.

Kryś, J. et al. Distributed and decentralised training: Technical governance challenges in a shifting AI landscape, 2025. URL https://arxiv.org/abs/2507.07765.

Kulp, G. et al. Hardware-enabled governance mechanisms, 2024. URL https://www.rand.org/pubs/working_papers/WRA3056-1.html.

Lidin, J. et al. Covenant-72B: Pre-training a 72b LLM with trustless peers over-the-internet, 2026. URL https://arxiv.org/abs/2603.08163.

Meta AI. Introducing Llama 3.1, 2024. URL https://ai.meta.com/blog/meta-llama-3-1/.

Pilz, K. et al. Trends in AI supercomputers, 2025. URL https://arxiv.org/abs/2504.16026.

Rahman, R. et al. Over 30 AI models have been trained at the scale of GPT-4, 2025. URL https://epoch.ai/data-insights/models-over-1e25-flop. Accessed 21 Apr 2026.

Rocco, J. Senate passes bipartisan stop stealing our chips act, 2026. URL https://ari.us/senate-passes-bipartisan-stop-stealing-our-chips-act/.

Ryabinin, M. et al. SWARM parallelism: Training large models can be surprisingly communication-efficient, 2023. URL https://arxiv.org/abs/2301.11913.

Scher, A. et al. An international agreement to prevent the premature creation of artificial superintelligence, 2025. URL https://arxiv.org/abs/2511.10783.

Sevilla, J. How far can decentralized training over the internet scale?, 2025. URL https://epoch.ai/gradient-updates/how-far-can-decentralized-training-over-the-internet-scale/.

Sevilla, J. and Troynikov, A. Could decentralized training solve AI's power problem?, 2025. URL https://epoch.ai/blog/could-decentralized-training-solve-ais-power-problem/.

Sevilla, J. et al. The longest training run, 2022. URL https://epoch.ai/blog/the-longest-training-run/.

Speedtest by Ookla. Speedtest global index, 2026. URL https://www.speedtest.net/global-index.

State of California. Senate bill 53: Transparency in frontier artificial intelligence act, 2025. 2025–2026 Reg. Sess., enacted Sept. 29, 2025, codified at Cal. Bus. & Prof. Code §§ 22757 et seq.

Stich, S. U. Local SGD converges fast and communicates little, 2019. URL https://arxiv.org/abs/1805.09767.

Villalobos, P. and Atkinson, D. Trading off compute in training and inference, 2023. URL https://epoch.ai/blog/trading-off-compute-in-training-and-inference.

Wasil, A. R. et al. Verification methods for international AI agreements, 2024. URL https://arxiv.org/abs/2408.16074.

White House. Executive order 14110: Safe, secure, and trustworthy development and use of artificial intelligence, 2023. URL https://www.federalregister.gov/documents/2023/11/01/2023-24283/.