LLM pretraining: The Use-Case Blockchain Has Been Waiting For? *

Macrocosmos[1], Taoverse[1], Const[2], and Datura[3]

[1]www.macrocosmos.ai
[2]www.bittensor.com
[3]www.datura.ai

August 2024

Abstract

The creation of the state-of-the-art in Large Language Models (LLM)'s has to date been solely the province of large, well funded technology companies. In this white paper, we showcase our initial findings, demonstrating that it is possible to train such models in a decentralised manner, using the Bittensor Ecosystem, and argue that the creation of intelligence is the true use case that blockchains have been searching for. Subnet 9, the pretraining subnet within Bittensor, has developed 700M (million) and 7B (billion) parameter models that can outperform comparable industry-leading models such as *gpt2-large* and *falcon-7b*, respectively. After presenting the subnet's architecture and justifying design choices, we share our roadmap for future developments, which outlines our plans to further extend this novel proof-of-concept system.

*Alt. title: Incentives Are All You Need

Foreword

Blockchain technology has proven its ability to marshal decentralized computing power to address a specific challenge, such as facilitating transparent and tamper-proof voting in a decentralized voting system [1]. It is a foundational technology powering new models of economic value.

In this paper, we introduce new research showing how decentralized incentive systems, powered by blockchain technologies, can be used to tackle the biggest computational challenge of our age: developing State-of-the-Art (SOTA) open-source large language models (LLMs) in a transparent way. We demonstrate how multiple actors can be aligned by economic incentives through a Bittensor Subnet - in this case Subnet 9, its flagship pretraining subnet. Furthermore, they can collectively produce new pretraining models that outperform state-backed contemporaries such as *falcon-7b*, when benchmarked using modern web-scale datasets. This is a crucial contribution to Bittensor's mission of creating a collectively-run intelligence market that continually produces newly trained models.

With this paper, and our wider work on the pretraining subnet, we want to provide a blueprint for how a decentralized network can be harnessed and directed at the biggest computing challenges we face.

Contents

1 Introduction

1.1 Context

Generative AI is still in its infancy. We can see it's technically feasible, commercially viable and increasingly powerful. But key questions still remain: How can end users access generative AI at scale? Who should be building and governing LLMs? And how do we incentivise the creation of SOTA models that are constantly improving?

Underpinning the answer to these questions is the ownership structure of both machine learning models and the datasets they are trained on. Broadly, these can be conceptualized as:

- **Closed-source**: Referring to proprietary datasets and AI models that companies are developing internally and, in the case of models, distributing as a "black box" through a user-facing portal or API. Decisions about datasets and model design are made privately, shaped by the company's strategic priorities.

- **Open-source**: These are models and datasets that are freely available for developers to build with. Models such as Llama and Mistral are typically referred to as open source, although questions remain about how open these models are.

The core thesis of Bittensor is that there is a third way to build AI models: using a secure, distributed network to incentivise rapid open-source model improvement.

The Bitcoin blockchain provides a useful analogy to Bittensor. Since launch, Bitcoin has become the world's largest supercomputer [1]. It assembles computing power on a scale that is unmatched by any company, research body, or government agency. It then applies that computing power solely to the task of securing a proof-of-work blockchain and rewards participants according to their contribution.

At the highest level, Bittensor is a framework for distributed teams to work in parallel to marry the computational power of bitcoin, with the computational problem of the creation of machine intelligence, in the form of digital commodities [3]. It allows different groups to make contributions, have their effort verified by validators and then be rewarded in TAO, the Bittensor-native cryptocurrency. These participants include:

- **Miners** who provide the computational resources as well as leveraging their unique skill sets.

- **Validators**, who evaluate and rank the work done by miners and reward

them based on their contributions.

- **Subnet Owners**, who define both the competition and the evaluation criteria and as such determine the commodities which are produced by their subnet.

In some subnets, including pretraining, the competition rewards participants in a winner-takes-all fashion. In others, miners receive a percentage of total rewards based on their contributions. Within Bittensor, we have the ability to direct the computing power of all network participants towards solving specific problems, from protein folding [4] to LLM pretraining [5] and fine-tuning [6]. By combining open-source datasets and software development with the incentive structure and authentication of a blockchain, it is possible to coordinate computing power at scale, and develop AI commodities in a transparent, fair, and rewarding manner.

1.2 Why a Pretraining Subnet Belongs on Bittensor

Pretraining incurs very high costs due to its enormous energy consumption and reliance on specialized hardware and infrastructure. The computing power required for a state-of-the-art LLM doubles every 10 months. While some of this increasing compute is off-set by greater efficiency (due to both modern techniques and Moore's Law), it still translates into high costs for model development. Leading closed-source models don't publish the exact training cost of their models, but research from OpenAI suggested model development already costs around $100,000,000 and will reach $500,000,000 by 2030. Separately, OpenAI CEO Sam Altman has said that GPT-4 cost "more than a hundred million dollars" to train [7].

As the cost of model training increases exponentially, only the very largest companies can access and pay for pretraining and fine-tuning on the scale needed for SOTA models. The pretraining subnet on Bittensor offers an alternative: a unique set of incentives for pretraining models which, as we detail in section 4, incentivise miners to produce high quality models, given a set of constraints such as model size. Within Bittensor, pretraining is a logical use case for the ecosystem, as these pretrained models can then power other intelligence-based subnets. It is a pillar of the Bittensor ecosystem where marginal improvements to pretraining can have significant downstream benefits for other Bittensor-based projects.

Pretraining is also the most computationally intense stage of AI model development. Established AI firms are spending huge amounts to assemble the computing power they need to pretrain models - Meta's Llama3.1 8B took 1.46 million H100 GPU hours to train , while the larger 405B model took a staggering 30.84 million H100 GPU hours [8].

This is therefore one of the clearest use cases for developing a decentral-

ized alternative. Building on Bittensor allows us to subsidize that training cost while providing a platform for skilled practitioners to monetize their expertise and bring value to the wider ecosystem in the form of open models. This is why developing an effective pretraining subnet on Bittensor has been a priority, providing the foundation that other subnets and model builders can fine-tune for specific use cases. Figure 1 shows the empirical trends for AI energy consumption, overlaid with the same figures for cryptocurrency mining. Current trends are extrapolated to predict growth from 2026 to 2030. This supports our argument that using cryptocurrency mining as a vehicle for AI development is an excellent opportunity for procuring the necessary scale of compute for the next generation of AI systems.

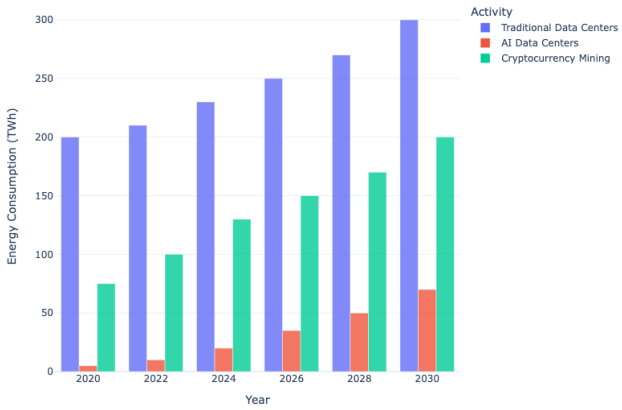


Figure 1: Energy consumption of cryptocurrencies and data centers. Projected growth is shown for 2026 and beyond.

1.3 Decentralized Training vs Decentralized Competition

The concept of decentralized training has become increasingly popular, and can be easily confused with the competitions that currently run within the pretraining subnet and Bittensor more broadly. Pluralis Research defines decentralized training as "the creation of foundation models over loosely-connected, heterogeneous swarms of consumer-grade computers" [9]. This modular approach would enable models to grow beyond the constraints of any individual miner. In simple terms, decentralized training can in theory produce models that are far larger than the memory capacity of individual miners by "spreading" the model layers across the network.

At the time of writing, there is an ongoing debate about whether such an architecture for model development is feasible in practice and what form

that could take. Today, the pretraining subnet harnesses a crowd of competing miners each delivering a complete pretrained model, which can then be built upon by others as the competition evolves. We discuss the roadmap for our pretraining subnet in section 5 which outlines a potential approach to swarm learning.

2 Subnet Architecture

2.1 Overview

At a very high level, participants in the pretraining subnet perform one of the following tasks:

- **Miners** individually pretrain large language models and upload them to a public hub (Hugging Face). Each model is owned by a single miner with a unique identifier (UID) and this association is written to the blockchain via a commit hash and verified by the validators (section 3.3.1).

- **Validators** independently download and evaluate the performance of each model against a randomly selected batch of a public web-scale evaluation dataset and assign a weight to each miner (section 2.3). They periodically submit their weights to the Bittensor chain, where the weights are combined using Yuma Consensus [3] and used to determine the TAO emissions. Yuma Consensus combines proof-of-work and proof-of-stake mechanisms to incentivise miners and validators respectively. Validators and miners are both rewarded with newly-minted TAO, based on their contributions.

In the following sections, we give a formal and simplified description of the subnet's components. For the sake of clarity, we will make the following assumptions:

1. Each miner has a unique UID, and therefore has trained and uploaded a single model.

2. Only one validator participates in the evaluation.

2.2 Miners

We consider a competition with n miners, each identified by a unique identifier UID_i, with $i \in \{0, ..., n-1\}$. Each participant trains an LLM f_i using their respective dataset(s) D_i of choice. For performance reasons, miners will most likely train on the same dataset they will be evaluated on. When miners are confident that they have produced the winning model in the competition, they upload the model to Hugging Face at a specific block b.

2.3 Validators

Validators download the set of pretrained LLMs $M = \{f_0, ..., f_{n-1}\}$, and evaluate them on a common evaluation dataset $\mathcal{D}_{eval}$.

2.3.1 The Evaluation Dataset

Given that miners have access to the validator's code, and could even be running a validator themselves, having a fixed and predefined evaluation dataset would inevitably result in the general adoption of trivial memorization strategies, rendering the subnet useless. To address this issue, we create $\mathcal{D}_{eval}$ at evaluation time by randomly sampling a subset of a very large and high quality dataset such as FineWeb [10]. Once $\mathcal{D}_{eval}$ has been created and properly tokenized, it is partitioned into a set $D = \{d_1, ..., d_m\}$ consisting of m batches d of equal size. The evaluation process consists of three steps:

- Computing the losses (per batch)
- Determining the winners
- Calculating the rewards

2.3.2 Computing Losses

The loss function used in this evaluation is perplexity. Before defining perplexity, we introduce the cross-entropy loss function, which is used to calculate perplexity.

Cross-Entropy Loss ($\mathcal{H}$): For a given model f and a batch $d = \{(x_t, y_t)\}_{t=1}^{T}$, the cross-entropy loss is defined as:

$$\mathcal{H}(f, d) = -\frac{1}{T}\sum_{t=1}^{T} \log P(y_t|x_t, f) \tag{1}$$

where T is the number of samples per batch, x_t are inputs, y_t are the corresponding target outputs and $P(y_t|x_t, f)$ is the predicted probability of the target y_t given the input x_t and the model f.

Perplexity Loss ($\mathcal{L}$): The perplexity loss is the exponentiation of the cross-entropy loss, which provides a more interpretable measure of model performance. It is defined as:

$$\mathcal{L}(f, d) = e^{-\frac{1}{T}\sum_{t=1}^{T} \log P(y_t|x_t, f)} \tag{2}$$

Perplexity is an intuitive objective function because it measures how well a model predicts a sequence of words, with lower values indicating better pre-

dictions. It's like assessing how "*surprised*" the model is by the actual outcome – the less surprised it is, the better it understands the data.

For the remainder of the paper, we will refer to $\mathcal{L}$ simply as loss. For each model f_i, the losses on the evaluation batches are computed as follows:

- **Initial Verification**: First, we ensure the model generates reasonable outputs. This is done by checking the model's outputs on the first two batches d_1 and d_2. Specifically, we generate a fixed number of tokens for each input and evaluate the diversity and repetitiveness of these outputs. If the model's outputs are too similar to each other or internally repetitive, which means they perform no better than randomly guessing words, it is assigned an infinite loss for all batches.

- **Loss calculation**: For the models that pass the initial verification, we compute the perplexities $\mathcal{L}(f_i, d_j) \equiv \mathcal{L}_{ij}$ on each batch d_j.

2.3.3 Determining Pairwise Winners

A pairwise win comparison is defined between any two models f_i and f_j. For each batch d_k, a win is determined using the function:

$$\text{isWin}(\mathcal{L}_{ik}, \mathcal{L}_{jk}, b_i, b_j) = \begin{cases} \mathcal{L}_{ik} < \mathcal{L}_{jk} & : i = j \\ (1-\epsilon)\mathcal{L}_{ik} < \mathcal{L}_{jk} & : \text{otherwise} \end{cases} \tag{3}$$

with i, j integers such that $i \leq j$, b_i and b_j are the blocks at which models f_i and f_j were uploaded to Hugging Face, respectively. Finally, ϵ is a positive constant factor which acts the increase rewards for earlier models.

Note that ϵ is used to ensure that a new model is considered better than an older model if and only if, the difference in their losses is greater by at least ϵ. This is necessary given that all models are publicly available in Hugging Face, and without imposing a minimum improvement threshold, downloading the top performing model and minimally tweaking it would be enough to game the scoring strategy.

2.3.4 Determining the Final Scores

The total number of wins W_i for each model f_i on $\mathcal{D}_{eval}$ are computed by iterating over all batches as follows:

$$W_i = \sum_{j \neq i} \sum_{k=1}^{m} \text{isWin}(\mathcal{L}_{ik}, \mathcal{L}_{jk}, b_i, b_j) \tag{4}$$

and the win rate R_i is defined as

$$R_i = \frac{W_i}{(n-1) \cdot m} \tag{5}$$

where $(n-1) \cdot m$ is the total number of pairwise comparisons each model participates in.

2.3.5 Setting Weights

Win rates are converted into weights using the softmax function, which introduces a probabilistic approach to model selection and gives higher weights to better-performing models. The softmax function is defined as

$$W_i = \frac{e^{\frac{R_i}{T}}}{\sum_{k=1}^{m} e^{\frac{R_j}{T}}} \tag{6}$$

where T is the temperature, which controls how much of the weight the top miners recieve. We use $T = 0.01$ which gives 96% of the weight to the best model.

These weights are then updated using a moving average method, which balances the influence of previous weights with the newly computed ones to ensure both stability and adaptability. Finally, models are prioritized for the next evaluation round based on their updated weights and win rates. Models with significant weights are given higher priority, followed by those with higher win rates, ensuring that the most promising models are evaluated more frequently.

3 Design Choices

Our approach is to impose as few constraints on miners as possible, in order to give them the greatest possible freedom to explore the model landscape. However, we have also made multiple design choices to align Subnet 9's goals and the work produced by its participants. Our three guiding questions are:

- How do we identify and compensate individual contributions?
- How do we incentivise continuous improvement?
- How do we mitigate risks?

These fundamental questions allow us to focus on key outcomes as we hone the subnet's design. Each of these questions ultimately leads us to maximize the competitive element of the subnet architecture, while reducing the risk posed by bad actors.

3.1 Evaluation

LLM benchmarks (such as HellaSwag [11] and MMLU-Pro [12]) provide a standardized measure of model capabilities and are commonly used by practitioners to provide an objective comparison between models. Despite their utility, benchmarks can be a double edged sword. In their pursuit of impressive results, researchers may prioritize performance on these metrics over broader research objectives (and even overall usefulness), as outperforming benchmarks is currently the most recognized method for demonstrating the superiority of an AI model. This phenomenon is encapsulated by Goodhart's Law, which states, *"When a measure becomes a target, it ceases to be a good measure."* This is particularly true when it comes to models trained by miners trying to maximize their rewards. Rewarding miners for performing well on known benchmarks could lead to overfitting on them, limiting their usefulness as an evaluation method.

By choosing a state-of-the-art loss function such as perplexity (used in GPT and Llama families of models, among others), and a SOTA dataset such as FineWeb Edu [14], we incentivize miners to perform well on those, laying the grounds to keep up with the latest closed-source models.

Miners know that they will be evaluated on a random subset of FineWeb Edu. Despite this knowledge, a miner would find it difficult to overfit on such a massive dataset. The vastness and diversity of the FineWeb dataset make it impractical to memorize or overfit specific patterns or examples. The random subset evaluation also means that the miners would struggle to predict which portions of the dataset they will be tested on, ensuring that their models need to generalize well across the entire dataset rather than performing well on a few specific sections. Furthermore, continuous updates and expansions of the evaluation dataset make it a moving target, further complicating any attempts at overfitting.

3.2 Model Size

Models submitted to the pretraining subnet have to meet specific sizes and architectures. Sizes have steadily increased over time - for the current competition, we cap the maximum model size at approximately 7B (billion parameters). Previous generations of competition on the subnet have run at smaller parameter limits: 186M and 700M. Moreover, we will be expanding to a multi-competition subnet, hosting concurrent competitions at 700M, 3B, 7B and 14B [1]. By introducing concurrent competitions of different model sizes, we hope to provide an on-ramp for teams to validate and refine their approaches in a lower-risk environment. An additional benefit to this design is that it enables us to analyze

[1] On August 12th 2024, the subnet was expanded to support concurrent competitions at sizes 700M, 3B and 7B, and an additional 14B competition will be introduced on September 10th 2024. We leave the reporting of these results to a future work.

the scaling laws of the models trained on our subnet, which will be used to plan much larger competitions.

While model size positively correlates with performance, limiting model size has a number of benefits. For miners, larger models require significantly more tokens to train, which translates to higher training times and costs. Skilled miners might struggle to compete if the entry costs keep increasing. Different use cases might require models of different sizes - to run on a laptop or phone, for example. Fixing model size allows us to focus on optimizing performance without simply relying on an increasing number of parameters.

3.3 Quantifying and Rewarding Contributions

3.3.1 Establishing Ownership

As a foundational requirement, we need to ensure that every participant on the subnet is identifiable and there is no ambiguity over model ownership. To achieve this, every miner and validator within the subnet has its own UID. Miners pretrain models and upload those models to their unique Hugging Face repository. Miners commit metadata to the blockchain which identifies the unique model's repository as well as the hash for the model itself.

Hugging Face gives us a neutral 3rd-party platform for hosting models and verifying they're unique. Once uploaded, the weights of every model are visible to everyone miner. This creates the incentive for different teams of miners to constantly improve on what is currently the best performing model on the subnet. It also prevents participants from keeping their models private – *only by making their models publicly available can they be eligible to receive rewards.*

3.3.2 Reward Distribution: Winner-takes-all

In subnet design, form should follow function. Competitions are not standardized but shaped according to user needs, performance requirements, and miner behavior. On some subnets like Subnet 1, high bandwidth requirements mean that second-best options are useful substitutes during periods of high demand. The incentive mechanism therefore rewards multiple miners.

On the pretraining subnet, we have different constraints. For pretraining, we want a small community of committed miners who are focused on constantly improving the subnet's best LLM. This is the case because demand for AI systems is highly concentrated towards the current leading models. In this use case, intense competition is more valuable than diversification. Therefore compensation for miners is distributed on a winner-takes-all basis. The top-performing model receives most (95%+) of the miners' emissions on the subnet, while every other model receives almost nothing.

The consequence of this winner-takes-all compensation structure is that it

encourages greater professionalism among miners. At this stage, the subnet is developed enough to ensure that participants aren't simply running a model on their laptop and finding that it's competitive. They are instead committing, skill, resources and computing power to develop incrementally better pretraining models, in order to reach the top of the leaderboard and maintain their position.

3.3.3 Ensuring Continuous and Meaningful Improvement

As mentioned in section 2.3.3 new models have to cross the "Epsilon Threshold"; a minimum improvement on the loss function required to beat an earlier model on the leaderboard. At the time of writing, this threshold is set to $\epsilon = 0.5\%$.

With epsilon, we're looking to satisfy two competing criteria. The first is that the threshold has to be high enough to ensure stability on the subnet - we don't have constant changes to the top-performing model being driven by minuscule changes in model performance. 0.5% is a high enough threshold to ensure a new model has significantly improved over the existing best-performing model.

The second criteria is that the improvement threshold has to be low enough that miners are incentivised to train many increasingly powerful models. The ideal strategy for a miner on Subnet 9 looking to maximize earnings is to have a submitted model at the top of the leaderboard and multiple unsubmitted models at progressively higher performance levels. For the subnet as a whole, as long as two miners are following the same strategy, the subnet will see a steeply decreasing loss curve. The consequences of this are discussed further in section 3.4.2.

3.4 Limitations, Risks and Exploits

In this section we outline the potential limitations of a system such as ours, and how our design addresses them.

3.4.1 Data Limitations

There are several considerations that must be taken to ensure that the dataset is not a limiting factor for the production of high quality models, which are:

- **The throttling effect of sub-optimal datasets**: We've seen that the underlying dataset on which models are pretrained had a greater-than-expected impact on performance. Originally, we specified that models should be trained on the Falcon Refined Web dataset [13], but it became clear this dataset was throttling miner performance due to its lower quality data. We have shifted the validator dataset for the subnet to FineWeb Edu in order to address this issue, and we will continue to explore the highest quality datasets available.

- **Dataset bias**: Relying on FineWeb Edu introduces a new risk: miners have no incentive to use multiple training datasets, but solely focus on performing well on this benchmark dataset. While this is optimal behavior for the miners, it may limit the performance of the pretrained models. Expanding benchmarking tests, as outlined in section 5, will address this.

- **Avoiding overfitting**: As mentioned in section 3.1, miners know the dataset from which the evaluation subset comes from. To any machine learning expert, this would be a major red-flag for overfitting. However, our benchmarking shows that models perform well on common benchmarks that the miners are not training on. That demonstrates to us that our datasets are large enough that miners aren't overfitting their models to the dataset.

3.4.2 Model Hoarding

It can sometimes be the case that the top miner already has an even better pretrained model, but the subnet's current incentive does not encourage them to publish it. This misalignment between the miner's optimal strategy and the performance of the subnet as a whole reveals a design flaw, which the rebasing strategy (section 3.4.3) fails to correct by itself. We are implementing a decaying version of the epsilon threshold, which will directly address this issue.[2]

3.4.3 Embracing Openness

By design, the pretraining subnet is designed to encourage model sharing between miners. Once a better model is added to the subnet, it is publicly available for every other miner to use as a starting point for further training. We call this design "rebasing", in analogy to version control systems. This creates the incentive for teams of miners to constantly build upon the current best performing model and therefore delivers long-term performance improvement to the subnet, while minimizing the risk of one miner locking in an insurmountable advantage with a proprietary model. Secondly, it lowers the barrier to entry as participants are not required to train a model from scratch unless they prefer to.

There are some strategies that may be employed which can prevent collaboration between teams. These strategies typically preserve the inference capabilities of model while interfering with the performance of training when it is resumed. Two known examples, which involve the obfuscation of model weights, are

1. **Vanishing gradients**: Weights that precede normalization layers may be rescaled so that they are orders of magnitude smaller (and at the limit

[2] On August 8th 2024 an additional competition, named 7B*, was introduced which is identical to the 7B competition except for a lower epsilon threshold of $\epsilon = 0.1\%$. This experiment will be used elucidate the optimal value by means of a head-to-head comparison of results.

of the floating point precision). This would not impact inference by much, but gradients on such small weights would be vanishingly small, which would inhibit continued training.

2. **Exploding gradients**: Weights in the final layers of the model that are associated with rarely activated neurons can be magnified substantially. This will incur a small degradation of the model performance during inference, but will poison the network when training is attempted due to the backpropagation of very large gradients.

Understanding and ultimately preventing these counter-productive strategies is a key focus for us and we employ monitoring and red-teaming exercises to detect such models.

3.4.4 Preventing Cabals and Exploits

Every subnet on Bittensor faces the risk of cabals, and the specific nature of cabalistic risk depends on the idiosyncrasies of the subnet's design. That's why each subnet must develop its own approach to prevention. On Subnet 9, we have designed the subnet to minimize the potential for collusion between teams of miners. The winner-takes-all compensation model incentivizes competition and reduces the risk of teams colluding. There is no economy of scale for individual actors who run multiple miners, as only a single model can win at once. Miners who are at the top of the leaderboard get no further benefit from working with other teams when they are already receiving all the rewards of the competition. Only by improving upon the best model can a second team receive rewards.

Likewise, there is a perennial risk for every Bittensor subnet that individual miners will attempt to exploit the incentive mechanism. On Subnet 9, we manage this risk with the "Epsilon Threshold" approach (section 2.3.3), so we only reward valuable contributions, rather than random modifications to a model's weights that only achieve minor improvements in performance.

3.5 Transparency, Security and Ethics

With the pretraining subnet, we are building the foundations of a platform which will underpin years, if not decades, of model development. To deliver that potential, we've carefully considered the security and ethical consequences of our work on Bittensor. In terms of security, we ensure that miners only upload weights rather than code, which limits the risk posed by rogue miners. To address concerns around data privacy, we made the design decision to only rely on free, publicly available data sources for model pretraining.

Central to our approach to building the pretraining subnet is to share with the community as much performance data as possible through our dashboards [16]. These dashboards allow us to share information such as the network loss in real time.

4 Results

Since launch, we've continuously benchmarked performance on the pretraining subnet against comparable models. To date, the results have confirmed our starting hypothesis: models trained within Bittensor have been able to match or out-perform models at a comparable model size. In this section we outline the key results in two competitions; 700M and 7B. Figure 2 show the improvement of model losses in Subnet 9 over time for these competitions.

In tables 1 and 2 we compare perplexity losses of all models across 3 different benchmarking datasets; Wikitext103 [**?**], Falcon Refined Web and FineWeb Edu.

4.1 Competing at 700M Parameters

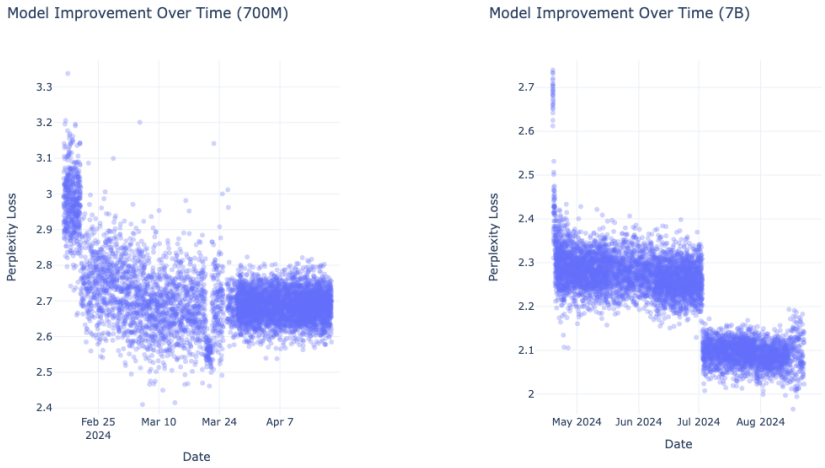
Post-launch, the first major competition we ran on Subnet 9 was at 700 million parameters, comparable to *gpt2-large*. Our goal was to demonstrate that models built on the subnet could compete with a household name like *gpt2-large*, which was state of the art in its time. Table 1 shows how the top-performing model developed in the 700M competition benchmarks against comparable models.

Model	Size	$\mathcal{L}^a$	$\mathcal{L}^b$	$\mathcal{L}^c$
gpt2	124M	30.13	35.25	29.63
gpt2-large	774M	19.50	23.89	19.5
phi-2	2.8B	9.79	15.19	12.09
Jonathan18/net9_miner3	769M	15.89	15.57	15.21

Table 1: Benchmarking results for different ≈700M models on 3 datasets. Results are given as perplexities on (a) Wikitext103, (b) Falcon Refined Web and (c) FineWeb Edu. The model *Jonathan18/net9_miner3* was produced by subnet 9.

The top performing model in the 700M competition (*Jonathan18/net9_miner3*) far exceeded the performance of *gpt2-large*, across every benchmark test we ran. The biggest difference in performance was on the Falcon Refined Web dataset (15.57 vs 23.875), which at the time was the dataset used for validating models in subnet 9. However, even on the other benchmarks we track, the top-performing model produced by our pretraining subnet outperformed GPT-2-Large.

Moreover, the top-performing model in the 700M competition recorded performance closer to that of Phi-2, an LLM developed by Microsoft, scoring 15.57 on Falcon Refined Web, vs. 15.1896 for Phi-2. Given that Phi-2 is almost 4x larger than the maximum size allowed for competition, it's a remarkable result.



(a) 700M competition

(b) 7B competition

Figure 2: Loss curves ($\mathcal{L}$) for (a) 700M parameter and (b) 7B parameter competitions. Decreasing trends of loss signify improvements of the models over time. The sudden decrease in loss of the 7B competition was due to the use of an improved dataset (Fine Web)

4.2 Competing at 7B Parameters

Following the results from the early competition on the pretraining subnet, it was clear that the subnet could foster the development of new models that could outperform household names like GPT 2. Our next goal was to scale up the size of the competition. 700M is obviously small compared to current models. So as a second stage, we increased 10-fold the possible parameters to 7 billion. Table 2 compares the performance of the top model produced by subnet 9 with equivalent models; *mistral-7b*, and *falcon-7b*.

At 7 billion parameters, the top performing model on the subnet (*jw-hf-test/jw2*) outperformed *falcon-7b* on the FineWeb Edu dataset and the loss continues to improve. We also note that scores for both Wikitext103 and Falcon Refined Web are approaching SOTA. This is a major milestone for the Bittensor ecosystem, as we've shown an open-source model built in a decentralized ecosystem can out-perform a state-backed project like Falcon.

Model	Size	$\mathcal{L}^a$	$\mathcal{L}^b$	$\mathcal{L}^c$
falcon-7b	6.9B	6.563	11.063	9.938
Mistral-7B-v0.1	7.2B	4.93	9.113	7.178
jw-hf-test/jw2	6.9B	7.084	13.648	8.593

Table 2: Benchmarking results for different ≈7B models on 3 datasets. Results are given as perplexities on (a) Wikitext103, (b) Falcon Refined Web and (c) FineWeb Edu. The model *jw-hf-test/jw2* was produced by subnet 9.

5 Roadmap

5.1 What is the future for the pretraining subnet?

We are still in the early stages of Subnet 9 and the capabilities that come with an open source, incentivized model for building pretraining models. Having demonstrated our initial hypothesis by showing that a subnet can be used to develop a pretraining model, our focus is now to further improve and scale the subnet. These include:

1. **Multiple concurrent competitions**: Other subnets run multiple competitions, allowing individual miners to compete on specific tasks. This allows more miners to earn TAO on the subnet, while also creating more value by means of a larger range of models. On August 12th 2024, the subnet was expanded to support concurrent competitions at sizes 700M, 3B and 7B, and an additional 14B competition will be introduced before the end of August 2024. Results for these ongoing competitions can be found on Macrocosmos' subnet 9 dashboard [**?**]. We are also investigating use cases for pretraining multi modal and omni models, which is an exciting area of research in modern AI. Lastly, we plan to introduce further competitions for larger models in the coming months so that subnet 9 can continue to push the frontiers of decentralized pretraining.

2. **Incentive mechanism improvements**: We will continue to devise new experiments to analyze and refine the incentive mechanism design in order to maximize the value created by our subnet. On August 8th 2024 an additional competition, named 7B*, was introduced which is identical to the 7B competition except for a lower epsilon threshold of $\epsilon = 0.1\%$. This experiment will be used elucidate the optimal value by means of a head-to-head comparison of results. Live results can be found in Macrocosmos' subnet 9 dashboard [**?**]. Beyond this, we are experimenting with various formulations of "dynamic epsilon", which decays the ϵ over time in order to ensure that competitions do not stagnate based on sub-optimal ϵ values. Our first version of "dynamic epsilon" will be released before the end of August 2024. We will also investigate how we can safely relax key constraints on the competitions such as architectures and tokenizers, so

that competitions can be less restricted and more productive. Our goal is to resolve any remaining open questions on the optimality of the present system design by running open experiments with the subnet 9 mining community.

3. **Improved evaluation of models**: To date, we've measured models on their natural language generation. Existing model benchmarking increasingly uses a suite of different tasks such as coding and math problems to assess performance. Implementing a wider set of benchmarks is a natural next step in subnet 9's evolution. Furthermore, we are developing novel synthetic benchmarking datasets that will enable us to scrutinize model performance in an objective way, building on our work on other subnets within Bittensor.

4. **Directability and productization**: The broader goal for the pretraining subnet is to reach a point where start-ups and small businesses can overcome the expense and technical challenges of training their own LLMs by using the pretraining subnet, as well as a broader suite of directed Bittensor subnets. We are reaching a point where the pretraining subnet is becoming the foundation for a wide range of use cases, both internal to Bittensor and within the broader AI marketplace. Already, the pretraining subnet produces base models for Bittensor's flagship fine-tuning subnet. By building this market around pretraining, we'll be able to deliver the economic potential of pretraining on Bittensor.

5. **Fully decentralized training**: As mentioned in 1.3, the pretraining subnet has been designed to host decentralized pretraining competitions. To date, we have demonstrated that this framework works in practice. Experiments are already underway at Macrocosmos which explore how to create a subnet that isn't limited by the capabilities of the best individual miner. In order to tackle larger challenges in the future, we plan to evolve the pretraining subnet towards a decentralized training model where miners are collaborating on model development, rather than each developing their own separate model.

6 Conclusion

This whitepaper demonstrates that it is possible to develop pretraining models on Bittensor that match or exceed the capabilities of equivalent SOTA models. By combining an open-source model development with a blockchain-based incentive mechanism designed to encourage competition, the pretraining subnet shows what Bittensor is capable of delivering. Our key findings are:

- At 700 million parameters, top-performing models on Bittensor notably outperformed equivalent models like *gpt2-large* and achieved results comparable to models multiple times larger.

- At 7 billion parameters, we have seen that they are equally competitive, matching state-sponsored models like *falcon-7b* across every benchmark test that was used.

These preliminary results serve as our proof-of-concept for a pretraining subnet, and are the product of a living experiment which began in November 2023. The present work is foundational research that demonstrates how an incentive structure can be used to leverage decentralized computing power and drive real improvements in pretraining model performance. The subnet is ultimately designed to encourage exponential improvement in model performance through a winner-takes-all competition. We're still at the start of that exponential improvement. Our future roadmap will build on these successes, and will learn from our mistakes.

The appropriate framing of these results is to compare them to equivalent-sized projects. This is because the tech giants who are currently producing industry-leading models are allocating resources on the order of billions of dollars and thousands of world-class researchers and developers. Meanwhile, Bittensor is powered by a small group of passionate, highly motivated individuals. The sum total of all results which we present in this whitepaper were funded by the Bittensor token TAO and it's community, by means of miner emissions, who received estimated lifetime earnings of circa $5M for their contributions. At Bittensor continues to grow, it will further attract top talent who want to monetize their expertise and build an open future for AI. For these reasons, we believe the future of pretraining in Bittensor is bright.

"Never doubt that a small group of thoughtful, committed citizens can change the world; indeed, it's the only thing that ever has." – Margaret Mead

7 Acknowledgements

The authors of this paper would like to thank the Bittensor community for supporting this foundational work on incentive design, and for their valuable feedback and insights.

In particular, we acknowledge that the present results would not have been possible were it not for a group of highly skilled miners who continue to push the frontiers of decentralized pretraining.

Lastly, the authors would like to thank the subnet 9 team (past and present); Const, Fish, Sid, Rustic, Alan, Rodrigo, Will and Steffen.

8 About Macrocosmos

Macrocosmos is an open-source AI research lab, building on Bittensor.

By creating competitive, secure, validated marketplaces for AI development, we unlock the potential of distributed networks to serve broader interests beyond the walled garden - create powerful, decentralized, and equitable intelligence for all.

Bittensor is the largest open source blockchain network for training and building machine learning models. Harnessing distributed compute at scale, Bittensor incentivises and rewards participants, creating an ecosystem of skilled teams competing to improve model performance. In turn, this drives efficiency up, and the cost-of-compute down. Its open-source architecture gives AI innovators what they need to thrive: transparency, ownership and accountability.

At Macrocosmos, we build and operate five subnets on Bittensor. We're honing the environment on which AI innovators can build new applications, by designing incentive mechanisms and reward structures that make world-class compute accessible and affordable. We see ourselves as an incubator for the Bittensor platform: conducting groundbreaking research, testing new concepts, and continuously driving innovation.

Each subnet - language models, pretraining, fine-tuning, data-scraping and protein folding - is a network incentivising the development of machine learning models. Together, they can enhance performance across the whole Bittensor ecosystem, creating a platform for new, open source AI applications at scale.

References

[1] https://bitcoin.org/bitcoin.pdf

[2] https://www.blockchain.com/explorer/charts/hash-rate

[3] https://bittensor.com/whitepaper

[4] www.macrocosmos.ai/sn25

[5] www.macrocosmos.ai/sn9

[6] www.macrocosmos.ai/sn37

[7] https://www.wired.com/story/openai-ceo-sam-altman-the-age-of-giant-ai-models-is-already-over/

[8] https://huggingface.co/meta-llama/Meta-Llama-3.1-8B-Instruct

[9] https://pluralisresearch.substack.com/p/decentralized-ai-looms?utm$_campaign = postutm_medium = webtriedRedirect = true$

[10] https://huggingface.co/spaces/HuggingFaceFW/blogpost-fineweb-v1

[11] https://paperswithcode.com/dataset/hellaswag

[12] https://arxiv.org/abs/2406.01574

[13] https://huggingface.co/datasets/tiiuae/falcon-refinedweb

[14] https://huggingface.co/datasets/HuggingFaceFW/fineweb-edu-score-2

[15] https://arxiv.org/pdf/2004.08900

[16] https://www.macrocosmos.ai/sn9/dashboard