qBitTensor Labs

Open Mining on Quantum Compute (SN48)

Performance, stewardship, and verified compute: how the incentive mechanism works, and why

2026-10-09

Summary

Quantum Compute (Bittensor subnet 48) is opening mining to the whole community. Until now, mining SN48 meant owning, operating, or having a direct relationship with someone with a physical quantum computer. With open mining, any participant can run a miner: the miner rents metered quantum compute through a trusted quantum provider, executes real user circuits, hands the validator a verifiable *proof of compute*, and earns alpha for it.

SN48 alpha becomes the first asset that can only be mined by faithfully computing real and verifiable workloads on real quantum computers. The incentive mechanism that makes this safe rests on three pillars:

1. **Verified compute.** Emission pays for metered, cryptographically attested quantum work and nothing else. Every validator computes the identical payout vector from one shared, frozen snapshot of the platform's books.
2. **Performance.** Reliable miners win more work; unreliable miners are backed off; cheating is caught by proof-of-compute and punished with bans.
3. **Stewardship.** Extraction harms a subnet structurally: emission follows the value of the subnet's alpha, so a miner that simply dumps what it earns erodes the pot for everyone, itself included. The design prices that harm in: work is dispatched in proportion to the alpha a miner holds staked to its hotkey. Holding the asset earns more work, and more work earns more emission; selling down the stake forfeits both. Skin in the game is the admission ticket and the multiplier.

This paper describes each mechanism as implemented, then explains the design reasoning behind it.

The marketplace

SN48 is a two-sided market. On one side, users (researchers, students, enterprises) submit quantum circuits through our partner platform, Open Quantum, an experience deliberately free of crypto friction: it transacts in USD, speaks OpenQASM, and installs with pip. On the other side, miners execute those circuits and return results.

Three roles interact on the subnet:

- **Our partner (Open Quantum)** holds the job queue, meters the cost of every execution in credits (1 credit = $1 of metered compute), and runs the open attestation service that verifies executions.
- **Validators** pull jobs from the platform, dispatch them to miners over Bittensor synapses, track every execution in flight, verify proofs, and set on-chain weights from the shared snapshot.
- **Miners** accept jobs, submit them to a quantum backend through a provider adapter, upload results, and return completion proofs.

The unit of supply is a **backend class**: a named class of execution capability (for example, a specific QPU family). Miners advertise the classes they can serve; validators dispatch each job to a miner advertising the matching class.

Trusted quantum providers

Miners do not execute against arbitrary backends. To open mining to the broader community who don't operate quantum computers, all mining runs against **trusted quantum providers**: quantum platforms holding a certificate from the Open Quantum Certificate Authority, a free, public CA for quantum computing platforms operated by our partner. The trust anchor sits outside the subnet operator: qBitTensor Labs cannot mint trusted providers unilaterally. A certified provider signs every job result it produces with a post-quantum signature (ML-DSA-65, NIST FIPS 204), and the CA publishes the trust bundle and revocation list. That makes the miner–provider relationship trustless where it counts: a validator can verify that a result genuinely came from an authorized quantum computer without trusting the miner and without taking the provider's word for it. The proof travels with the result.

Today, Open Quantum itself is the first provider on that list, but the list is open. Any QPU platform, whether a hardware operator or a quantum cloud, can onboard as a trusted provider by generating a key locally, keeping it offline, and requesting certificate issuance (support@openquantum.com).

Why a platform in the loop?

Quantum compute is not a commodity you can verify by re-running it cheaply: re-execution on a QPU costs real money, and results are probabilistic. A naive "validators re-run everything" design would double the cost of every job and still not produce bitwise-comparable outputs. Instead, SN48 puts the metering, attestation, and bookkeeping in one auditable place and keeps the *judgment* decentralized: every validator independently verifies proofs, tracks its own view of miner behavior, enforces its own bans, and independently computes the weight vector from the same published books. The platform collates; validators decide.

How work is dispatched

Dispatch is a continuous loop that every validator runs independently, on a fixed 30-second cadence.

1. The willingness sweep. Each cycle, the validator queries every serving miner with a lightweight collect synapse. The response carries three things: finished executions (with proofs), heartbeats for work still in flight, and a willingness signal. Miners signal when they are temporarily unable to take new work, and receive none until they clear it.

2. Capability claims. Miners advertise their serviceable backend classes; validators cache these advertisements with a short TTL (10 minutes). Stale or empty advertisements make a miner

ineligible for new work until it advertises again.

3. The stake lottery. Among miners that are serving, willing, not banned, not backed off, above the stake floor, and advertising a matching class, the validator selects one miner at random with probability proportional to its staked alpha, representing a mechanism to incentivize good stewardship. Two gates apply:

- **Stake floor:** a miner must hold at least **5,000 alpha** staked to its hotkey to receive work.
- **Linear weighting:** a miner with twice the stake wins, on average, twice the dispatch.

4. Backoff and bans. A miner that times out twice in a row is skipped for a period of time. A miner caught cheating on proofs, or that lets a job lease expire, is locally banned for 24 hours, with repeat strikes counted. Each validator maintains its own ban list; there is no central ban authority.

Why dispatch works this way

Why a stake floor? Three reasons. It is a commitment bond: a miner with 5,000 alpha at stake has something to lose from being banned, and something to gain from the asset doing well. It is a Sybil tax: registration spam cannot farm dispatch without buying real stake for every hotkey. And it is a quality filter: dispatching paid-for compute jobs to drive-by hotkeys wastes real user jobs on operators with no commitment to completing them.

Why per-validator bans instead of a global blacklist? A central ban authority would be a central point of capture. Local bans mean a cheater loses every validator's dispatch independently. And because every validator runs the same proof checks on the same evidence, a real cheater is banned everywhere within one lease cycle anyway. The redundancy is the security.

Why linear stake, not a curve? Linear weighting is Sybil-neutral by construction. Split a 100,000-alpha position across ten hotkeys and the total expected dispatch is exactly what it was as one hotkey: no gain, no loss. Any concave curve (such as a square root) would pay a bonus for splitting; any convex curve would punish small miners beyond what their stake already implies. Linear is the unique exponent at which identity games are a waste of time, so the mechanism does not need identity policing.

How work is verified

Proof-of-compute. Every private-job execution returns a proof (`pot.job.v1`) signed by the trusted provider's CA-issued certificate, binding the job, the backend, and the result. The validator verifies the signature chain locally against the CA's public trust bundle; the platform then binds the proof server-side, so the same proof can never be replayed for a second execution. The attestation confirms the four things that matter: the right job ran, on the right backend class, with the right parameters, and the returned results are the results the hardware produced.

Zero tolerance at the proof layer. A forged proof, a replayed proof, a backend-class mismatch, or a proof bound to another execution triggers an immediate 24-hour ban from the detecting validator, the job is failed back for reassignment, and strikes accumulate. There is no appeals process at the protocol layer: honest infrastructure failures look like failed jobs (which cost only reliability), never like bad proofs.

Leases. Every dispatched job carries a lease, heartbeated through the sweep. A miner that goes dark holding a job (neither completing, failing, nor heartbeating it) gets the job pulled and reassigned, and eats a 24-hour local ban. The lease state machine distinguishes the validator

that dispatched the work (which may fail the lease on the platform) from observers (which only ban locally), so one flaky network path never kills a live job that another validator can still see progressing.

How weight setting works

Weights are computed per epoch from a single **frozen snapshot** published by the platform. The snapshot contains a metered report of each miner's work. Every validator feeds the same snapshot through the same arithmetic, producing the same weight vector. Each miner's share tracks the verified metered work it delivered. More verified work, larger share.

Stewardship: why a QPU-mined asset is different

SN48 alpha is unlike any other mineable asset: it is the only cryptocurrency in the universe that can only be mined through verified quantum computation. The mechanism is built so that the people earning the asset are structurally aligned with its value:

- **Mining requires holding.** The 5,000-alpha floor means every miner is a holder before it earns its first job.
- **Holding more earns more.** Linear stake-weighted dispatch means a miner's work share tracks its conviction. Accumulating alpha compounds into more work, more verified compute, and more emission.
- **Selling has a built-in opportunity cost.** A miner that sells down its stake keeps every token it sold, but its dispatch share falls in exact proportion. The mechanism never punishes selling; it simply stops subsidizing it.
- **Mining costs real money.** Every job a miner executes consumes paid, metered compute. A miner only comes out ahead if the alpha it earns is worth more than the compute it spent, so every active miner has a direct, ongoing interest in the asset holding its value.

The result is a supply side made of operators who are simultaneously service providers and investors: the configuration most likely to keep a verified-compute asset valuable while the demand side scales.

Getting started

Register a hotkey on netuid 48, stake at least 5,000 alpha to it, configure an account on Open Quantum, or another trusted quantum provider, and run the miner. Operators of physical QPUs can onboard hardware-backed classes through the existing partnership path, and quantum platforms can join the trusted-provider list through the Open Quantum CA (support@openquantum.com for both). The miner operator guide in the subnet repository covers setup end to end.

- **Code:** github.com/qbittensor-labs/quantum-compute
- **Subnet:** qbittensorlabs.com/quantum
- **Partner platform:** openquantum.com
- **Certificate authority:** ca.openquantum.com

Quantum Compute is operated by qBitTensor Labs. Subnet code is public at github.com/qbittensor-labs/quantum-compute.