



## Bitcast Brief 023

### ◆ **Supporting Document**

---

#### **Purpose of the Video**

Explain Yuma Consensus in clear, everyday terms so that viewers walk away understanding why it's different and why it matters.

The aim is not to cover every technical detail, but to give a high-level explanation that anyone wanting to learn more about Bittensor can grasp.

---

#### **Example Titles for Videos:**

“Yuma Consensus & Decentralized AI: How Bittensor Beats Centralized Labs”

“Why Yuma Consensus Makes Bittensor the Future of Decentralized AI”

“How Yuma Consensus Empowers Bittensors Decentralized AI to Outperform Centralized Labs”

“Yuma Consensus: The Key to Secure, Fair, Decentralized AI”

## 🌱 What is Yuma Consensus?

YC can be best understood by analogy to Bitcoin. On the Bitcoin network, miners are rewarded for producing blocks. The process of producing blocks involves using large amounts of compute in order to find a valid hash meeting the criterion for block production. By extension, the Bitcoin network incentivizes miners to bring lots of hash power (compute) to the network.

Yuma Consensus extends this concept to incentivize behavior more complex than producing sha256 hashes. Instead, Yuma is able to reward miners for producing any work of measurable quality. This includes storage, AI model training, or AI inference, as well as a host of other processes.

- Bittensor's mechanism that decides how validators agree on miner rewards.
- Think of it as the "fair scoring system" of the network.

---

## 🌍 High level process

The high level design of Yuma Consensus works as follows: Validators, which are peers on the network, interact with and score a set of miners (secondary labeled peers) to determine their skill or productive value to the network.

Subnet owners determine the rules for scoring miners. Validators then perform this evaluation and score the quality of the miner's work (a value between 0 and 1). YC then aggregates these scores and allocates rewards for miners and validators accordingly.

Examples: Validators may -

- Send inference requests to the miners (SN 64)
- Measure the speed they found proteins submitted to storage (SN68)
- Read the code they submitted to github (SN62)
- Evaluate the models they trained (SN56)
- Evaluate the gradients they produced in the last window of time (SN3)
- Check if they have a GPU available for rent (SN64).

After running this process of "checking" or "weighting" the miners based on some subjective measure of value, the validators tell the Bittensor blockchain how good each miner is.

This is where Yuma Consensus comes into play. **YC is the algorithm which aggregates the scores produced by each of these validators and turns them into payouts (emission) for the miners based on their performance scores.** You can think of YC like a function which accepts the knowledge (weights) from the network validators and turns it into emission for miners.

YC allows Bittensor to be able to incentivize the coming to agreement of validators attempting to assess miners when there would otherwise be no incentive to do so. In particular, there could be reasons to intentionally not reach consensus and to vote for their own miners or do other malicious things with their power of judging miners, but YC is a powerful tool in the prevention of this behavior.

At a high level, **YC's internal algorithm pays validators for being in agreement with the lion share of other validators, and punishes them for being outside consensus.** As validators diverge from this consensus, their ability to pay miners decreases AND they receive less payment themselves.

In addition to the above, YC also rewards discovery. This means that if validator A is the first to discover a new miner doing great work, and the rest of the network sees the great work this miner is doing and agrees with validator A, then validator A will receive increased rewards.

- Each subnet designs its own mechanism for scoring miners
- Each validator runs the mechanism and provides a score for each miner
- The Yuma Consensus programmatically checks each validator's score against the scoring of all of the other validators running on each subnet.

---

### **How does YC deal with the diversity of validator weights?**

Before rewards are emitted to miners, Yuma Consensus compares all validator scores.

If there's a discrepancy, the system doesn't ignore the outlier. **First it applies a process called clipping** - a mathematical alteration where specific validator scores may be reduced if they are too far out of the consensus (the precise parameters here can vary from one subnet to another, based on the initial choices made by the subnet owner).

YC then calculates a **stake-weighted average across all of the clipped validator scores.** This ensures every validator's opinion is included, but those with more stake have proportionally more influence.

---

### **Is a validator penalized if they are not in consensus?**

Validators are rewarded by the network for measuring the quality of miner work.

This work is important because these scores are used to determine the incentives paid out to miners. As such, it is important for validators to achieve consensus.

This means that if a validator's score does not fall into consensus with the other validators their rewards are reduced (potentially all the way to 0).

Therefore, they are incentivized to do a good job running the scoring mechanism.

---

### Why it matters

- YC allows validators to come to consensus on the quality of miner weights to in turn incentivize miners to produce better and better outputs to get rewards.
- Keeps the system **fair**: no single validator controls the scoring.
- Ensures **decentralization**: decisions are spread across many validators.
- Provides **security**: reduces manipulation or bias.
- **No central scoring method**: Yuma enables multiple validators to secure the network whilst minimising the impact of disagreement between them.

This system prevents validators from manipulating results. For example, if a validator were to create their own miner and attempt to direct rewards to themselves, their fraudulent scoring would not align with the majority of honest validators. As a result, it would fail to reach consensus.

---

### The difference from centralized AI

- **Open participation** - Anyone with a GPU or model can plug in and start earning. Unlike closed AI labs, Bittensor is open to the best talent worldwide — **no CVs, no gatekeepers, no elite university filter.**
- **Permissionless scaling** - The network expands organically as more miners join. No central bottleneck, no approval needed — growth is built into the protocol.
- **Unstoppable Continuity**, No lab shutdown, boardroom vote, or regulator can “turn off” Bittensor. Yuma Consensus keeps the network running even if validators or miners drop out.
- **No single point of failure** - Instead of a central IT team controlling inputs, hundreds of miners secure the network. Decentralization ensures resilience, redundancy, and trust.
- **Incentive alignment** - Yuma ensures that **good outputs are consistently rewarded**. This makes decentralized networks self-optimizing.

- i. Validators who mis-rank or act maliciously are economically penalized through clipping and slashing.
  - ii. Centralized AI relies on trust in internal ethics boards with no enforcement mechanism.
- **Stake-weighted trust** - Validators with more stake have more influence, aligning long-term incentives with network health.
  - **Composability** - Subnets can specialize (compute, inference, storage, search, content, etc.) while all settle rewards on the same TAO ledger.
  - **Rewards flow to miners who consistently prove value** - not to whoever controls the most GPUs or has the deepest funding.
  - **No one can blacklist a participant** for political, financial, or competitive reasons. As long as you contribute quality outputs, you earn emissions.
- 

### **Big picture**

- Yuma Consensus is the foundation that allows Bittensor to scale trustlessly.
  - It's what makes decentralized AI possible - without it a single validator could control the scoring
  - This is how decentralized AI can scale in a trustless system and challenge centralized AI competitors
- 

### **Resources**

- Yuma Consensus documentation: <https://docs.learnbittensor.org/yuma-consensus/>