Aggregation of Inference Results

The final step in our decentralized inference system is the aggregation of results submitted by individual nodes in the inference committee. This crucial step determines the official outcome of the inference task.

To ensure the integrity and accuracy of the aggregated result, our system implements a default majority vote strategy within a smart contract. We also support customized aggregation outside smart contracts, which will be described later in this section.

Here is how the default aggregation strategy works:

- **Majority Vote within Smart Contract:** A smart contract is tasked with aggregating the results. It tallies the submissions and identifies the majority result, meaning the outcome that has been reported by more than half of the nodes in the committee. This majority result is taken to be the correct and official outcome of the inference task, and it is this result that is ultimately communicated to the user or utilized for further on-chain actions.
- **Fault Detection and Punishment:** Any submitted result that does not align with the majority is flagged as faulty. The node responsible for a faulty result is subject to a penalty. This could manifest as a loss of a security deposit (slashing), a reduction in reputation score, or both. The specific nature and severity of the punishment are pre-defined in the smart contract's rules and are automatically enforced. The penalization protocol serves as a deterrent against submitting incorrect or dishonest results, thus motivating nodes to perform their computations diligently and accurately.
- **Finalization and Reward Distribution:** Once the official result is determined and faulty nodes are penalized, the smart contract finalizes the result and triggers the appropriate reward distribution to the nodes that contributed to the majority result. Rewards are allocated as per the predefined incentive structure, balancing the costs incurred by nodes and incentivizing continued honest participation in the system.

This majority vote strategy ensures that the aggregated result reflects the consensus of the committee, thereby reducing the likelihood of erroneous outcomes due to individual node failures or malicious behavior.

It also reinforces the reliability and trustworthiness of the decentralized inference system, as all nodes are accountable for their contributions and the overall process is transparent and verifiable. The smart contract serves as an impartial and incorruptible arbiter that enforces the rules of the aggregation protocol, ensuring that the system remains fair and resilient.

Customized Aggregation

Nesa is designed to offer flexibility in how inference results are aggregated to produce a final output. While the majority vote strategy serves as a robust default aggregation method within the smart contract, certain use cases may demand more tailored approaches. To cater to these needs, we introduce a new concept called “Customized Aggregation” where model owners can specify their own aggregation logic that goes beyond the capabilities of a smart contract. This subsection outlines the components and considerations of the customized aggregation process.

Inference Results Set Retrieval

After all participating nodes have revealed their inference outputs and the smart contract has recorded these revelations, the contract will now provide the raw set of inference outputs without directly applying any aggregation strategy. This raw result set will be available for further processing as defined by the model owner’s custom aggregation logic.

Custom Aggregation as Part of AIVM.

The custom aggregation code is authored by the model owner and is an integral part of the associated AIVM. This code is stored on the blockchain alongside the model parameters and the AIVM configuration file, ensuring that the entire inference and aggregation process is verifiable and transparent. The model owner can then tailor the aggregation process to the unique requirements of the model and the inference objectives.

Diverse Aggregation Methods

Customized aggregation strategies can range from simple methods like majority voting or averaging to more complex techniques such as averaging with outlier removal. The choice of aggregation method is determined by factors such as the nature of the model, the

desired robustness against aberrant results, and the level of consensus needed among nodes.

Result Verification

Ensuring the integrity of the inference results and identifying nodes that may have submitted outputs without performing the actual computation is critical to ensuring the integrity of the inference results.

The AIVM addresses this by implementing various verification methods:

a) Outlier Identification: This involves statistical analysis within the aggregation code to detect and handle outputs that deviate significantly from the consensus or expected range of results.

b) Publicly Checkable Proof: Similar to approaches used in academic works like TownCrier, the model owner can require nodes to produce zero-knowledge proofs or leverage trusted hardware attestations to publicly verify the correctness of their computations.

c) Outlier Quota and Proof Submission: Each node could be assigned an outlier quota, limiting the number of times it can deviate from the consensus results before it is required to submit a proof of computation. If a node exceeds its quota, it must provide such proof to maintain its standing in the network and avoid penalties.

With these customized aggregation strategies, the AIVM provides model owners with a rare level of flexibility to define how results are synthesized to best suit their use cases.

Previous
Free-Riding Prevention

Next
Step-by-Step Process of Decentralized Inference

Last updated 1 month ago