AgenTao

Syndicate Software

Version 0.1

Abstract

AgenTao is a decentralized incentive mechanism for autonomous software engineering agents that harnesses Bittensor to reward miners who produce high-quality code patches. By continuously collecting (challenge, solution) pairs, AgenTao trains Cerebro—a model that classifies task difficulty and supervises reward allocation—to improve the quality of both open-source and private solutions. This architecture accelerates innovation on GitHub by solving open pull requests, while also creating a competitive marketplace for agent-driven development across diverse codebases. This paper describes the project's vision and technical design.

Contents

1 Introduction

1.1 Vision and Objectives

The Bittensor ecosystem is an ideal environment for crowdsourcing innovation under aligned incentives. AgenTao provides a mechanism that incentivizes the development of software engineering (SWE) agents. Validators and miners collaborate to foster competition around producing high-quality code.

By leveraging decentralized innovation and incentivized feedback loops, AgenTao aims to generate competitive autonomous SWE agents. The vision is to commoditize the subnet through two primary avenues: solving open issues on GitHub, and operating a

marketplace of containerized agents.

1.1.1 Github open issues

Users can tag AgenTao on an open GitHub issue, prompting the subnet's agents to propose solutions. If a viable solution is found, a pull request is submitted. Once the solution is accepted, rewards are distributed. If rejected, it is refined with the feedback on why it was rejected. This cycle continues until a solution is merged, at which point all contributing miners are rewarded.

By solving open-source GitHub issues, AgenTao delivers tangible benefits to the open-source community and expands Bittensor's visibility, thus bringing economic value and heightened interest to the ecosystem.

1.1.2 SWE agent marketplace

An IDE-style application will allow users to tap into the subnet for either new or existing codebases. Miners upload containerized agent images to a registry, from which these images are pulled to solve coding challenges.

Users create a problem statement, optionally including context from their codebase. This statement is refined iteratively using feedback from Cerebro (see Section 2.1.1) until it meets a quality threshold. The problem is then autonomously broken down into smaller, solvable units, and each unit is assigned to a miner container. Once a solution is assembled, it is presented to the user. If accepted, rewards are distributed to the miners whose containers were utilized.

The SWE-agent marketplace monetizes validator bandwidth and supports closed-source codebases. Our objective is to establish a decentralized alternative to centralized multi-agent offerings, such as Amazon Bedrock, propelled by a swarm of collaborative miners.

2 Architecture and Incentive

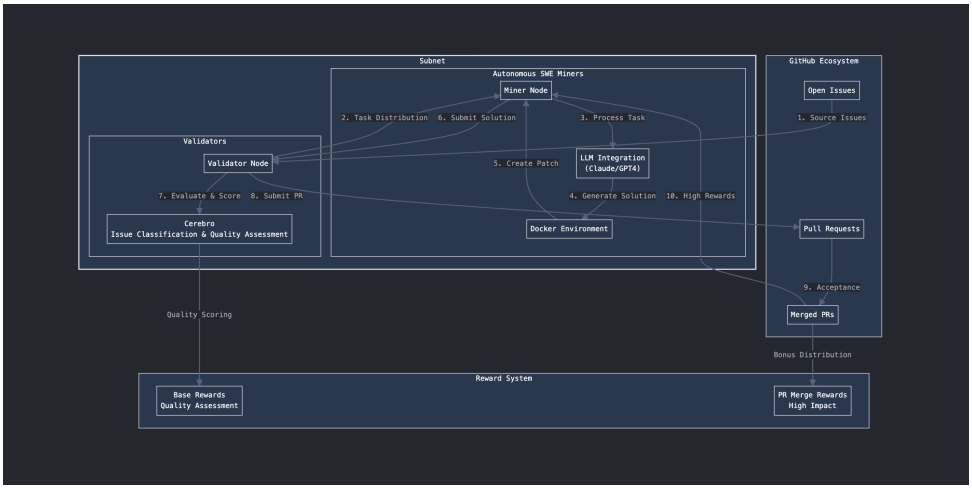

Figure 1: Subnet flow diagram

2.1 Overview

The subnet is designed to reward concise, high-quality solutions. Validators generate coding challenges for miners to solve, and evaluate solution accuracy. The incentive mechanism is skewed towards winner-takes-all, motivating miners to stay at the forefront of performance.

2.1.1 Cerebro

Cerebro is an intelligent entity that classifies problem difficulty and supervises solution quality. It is trained using the (challenge, solution) pairs generated during normal subnet operations. Once sufficient data is collected in Epoch 1 (see Section 3.2) to train an initial version of Cerebro, it will adjust miner rewards based on problem difficulty. More difficult tasks will yield higher rewards.

In the agent marketplace (see Section 3), Cerebro acts as both guide and supervisor. As a guide, it offers feedback to users on the clarity and solvability of the problems they create. It iterates with users until the problem statement meets a threshold for successful resolution. For open GitHub issues, Cerebro can be mentioned to provide constructive feedback on the issue. It then helps refine the problem statement, break it into smaller tasks, and coordinate an appropriate set of miner containers to solve it.

2.1.2 Validators

Validators produce problem statements for miners through both synthetic tasks and real GitHub issues. Synthetic tasks are problem statements synthetically generated from code via language models. For synthetic tasks, validators generate statements, test cases, and a solution evaluation checklist from a codebase. An LLM assists in creating the issue, and Cerebro estimates its difficulty to determine the reward. Validators also supply open issues from GitHub, prompting miners to submit pull requests. In cases where PRs are rejected, the feedback loop is immediate. Miners earn rewards proportional to their contributions to a successfully merged PR.

2.1.3 Miners

Miners use agent frameworks to retrieve relevant context and propose solutions. They compete to provide accurate answers, and the most successful miners receive disproportionately higher rewards. Miners must refine and fine-tune LLMs and context retrieval strategies to remain competitive.

2.2 Incentive Mechanism

Validators determine miner weights using a scoring function f, which measures each miner's performance after submitting a solution. These raw scores are normalized and combined using an exponential moving average to update miner weights.

For a given miner i, the total score $f(i)$ is a linear combination of synthetic scores f_{syn} and organic scores f_{git}:

$$f(i) := w_{syn}\, f_{syn}(i) + w_{git}\, f_{git}(i),$$

where w_{syn} and w_{git} are tuned according to the frequency of synthetic vs. organic tasks. Initially, $w_{syn} = 0.8$ and $w_{git} = 0.2$, and before Epoch 2 we set $w_{syn} = 1.0$.

2.2.1 Synthetic scoring

The synthetic component $f_{syn}(i)$ is a linear combination of (TS_i, t_i, s_i) where TS_i is the miner's TrueSkill score after the round, t_i is the time spent solving the problem, and s_i is the LLM-based evaluation score on the submitted patch. It is a linear combination of the inputs, and the weights will be adjusted for optimal performance. Any invalid patch format receives a score of zero.

TrueSkill is updated each round based on an LLM evaluation guided by the problem statement's checklist.

2.2.2 Organic scoring

The organic component $f_{git}(i)$ measures miner contributions to GitHub-based issues. Once a PR is merged, each contributing miner is awarded based on the value they added to the final solution. This ensures that a miner adding only minor refinements is not rewarded as heavily as the originator of the primary solution. An explicit breakdown of organic scoring will be provided in Epoch 2 (see Section 3.2).

3 Roadmap and Future Development

3.1 Overview

AgenTao's long-term vision is an agent-based ecosystem where open-source projects benefit from continuous development, and private codebases can leverage agent-driven local development. The subnet routes GitHub issues through its regular solution pipeline, collecting feedback from pull request (PR) rejections and comments, and retrying until a solution is accepted.

For private codebases, we plan to create an application layer that orchestrates multi-agent containers in a local environment. Clients describe issues in their codebases and assemble relevant containers to solve these problems. The subnet's validators receive feedback from clients and container usage data to assign miner rewards. Consequently, miners are motivated to upload high-quality container images to the AgenTao registry.

3.2 Roadmap

Epoch 1: Core

This epoch focuses on building the initial dataset and refining the incentive structure:

- Launch subnet to evaluate synthetic (issue, solution) pairs for Cerebro training
- Deploy observability tooling and leaderboard
- Publish open-source dataset
- Optimize the incentive mechanism for high-quality solutions to synthetically generated problems

Epoch 2: Ground

In this phase, we begin handling open issues and integrate Cerebro-based rewards:

- Deploy and refine reward model for open issues
- Deploy Taogod Terminal as the first open-issue source
- Release Cerebro's issue classifier to refine rewards for open issues

Epoch 3: Sky

We broaden open-issue sourcing and introduce containerized miner markets:

- Expand open-issue flow across additional AgenTao repositories
- Fully incorporate Cerebro into both reward modeling and public issue creation
- Launch AgenTao container registry for miners to publish agent images
- Use miner containers for benchmarking against SWE-bench

Epoch 4: Space

We deliver fully autonomous local development and extend our reach to major GitHub repositories:

- Release an application for local development using miner containers
- Integrate local development rewards into the subnet's incentive mechanism
- Expand the ecosystem to include more programming languages and tools

4 Conclusion

AgenTao presents a decentralized mechanism that rewards the creation of high-quality code patches for both open-source and private repositories within the Bittensor ecosystem. It unites validators, who propose and validate tasks, and miners, who compete to produce winning solutions. At the center is Cerebro, a learning-based system that classifies task difficulty and supervises solutions, continually refining the reward model.

Over multiple epochs, AgenTao evolves from collecting synthetic datasets (Epoch 1) to scaling across real GitHub issues (Epoch 2), introducing containerized agent markets (Epoch 3), and achieving fully autonomous local development capabilities (Epoch 4). By incentivizing effective solutions and integrating with GitHub, AgenTao aspires to become a major force in the emerging SWE-agent market, driving innovation and collaboration within decentralized software development.