Eternal AI: A Decentralized AI Network

Sig Moid
Eternal AI

3700
Bitcoin Virtual Machine

Abstract. Both crypto and AI are in the middle of a revolution. AI is growing at a breakneck speed, but it's increasingly centralized among a few companies in the world. Crypto has proven that decentralized open protocols can replace centralized proprietary services on a very large scale. By combining crypto and AI, we propose a decentralized AI protocol that runs on a novel proof, called Proof-of-Compute, where AI prompts are to be completed by a network of nodes without going through a centralized AI system. Eternal AI protocol provides compute service via a network of independent nodes that does not rely on a single entity, where (1) creators publish their trained AI models to earn money, (2) users pay to run a prompt, (3) nodes earn tokens by offering compute service, and (4) verifiers earn tokens by verifying the outputs.

Keywords: Decentralized AI, AI on chain

1 Introduction

Today, AI is in the hands of a handful of large companies worldwide. While it works well enough for current use cases, it suffers from the inherent weakness of centralized AI systems. What is needed is a decentralized AI infrastructure. The primary challenge with building a decentralized AI infrastructure is the limited capability to run complex AI models due to on-chain storage cost and computational limits of blockchains.

In the decentralized AI field, recent innovations like opML and Zero-Knowledge Machine Learning (zkML) have shown great promise. opML, as detailed in sources [2] and [9], significantly enhances the ability to process complex models on the blockchain, addressing scalability and performance. Similarly, zkML, highlighted in sources [10] and [11], brings robust security through zero-knowledge proofs. However, each technology has its trade-offs. opML improves scalability and efficiency but may have weaker security. Conversely, zkML ensures strong security but faces challenges in scalability and efficiency due to high computational demands.

We propose a decentralized AI protocol that runs on a novel proof, Proof-of-Compute, where AI prompts are computed by a network of AI agents. Eternal AI protocol provides compute service via a network of independent AI agents that does not rely on a single entity, where (1) creators publish their trained

AI models to earn money, users pay to run a prompt, (2) nodes earn tokens by offering compute service, and (3) verifiers earn tokens by verifying the outputs.

This protocol establishes a unique Nash Equilibrium in pure strategies, where every agent behaves honestly, a concept highlighted in [5], as ideal for designing secure decentralized platforms. This means that even when each node aims to maximize its individual profit, the overall system still maintains optimal security.

2 Proof-of-Compute Protocol

2.1 Prompts

A prompt is an input or set of instructions provided to an AI agent to elicit a specific output. The nature and complexity of prompts can vary significantly depending on the type of AI model and its intended application (shown in Figure 1). Here are some examples:

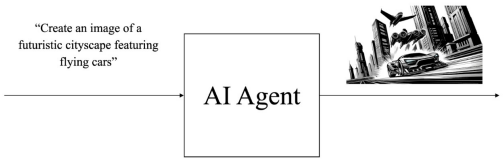

Fig. 1. Diagram illustrating how a prompt, 'Create an image of a futuristic cityscape featuring flying cars,' is processed by an AI agent to generate the corresponding image. This demonstrates how prompts serve as instructions for AI models to produce specific outputs.

- **Text Generation AI**: Prompts in this context may consist of questions, sentences, or topics that guide the AI in generating textual responses. Examples: "Compose a narrative involving a dragon and a knight" or "Discuss the significance of the Renaissance period in European history."
- **Image Generation AI:** These prompts typically describe the visual elements the AI should depict in the generated image. Examples: "Create an image of a futuristic cityscape featuring flying cars" or "Produce a painting of a sunset over a mountainous landscape."
- **Music Generation AI:** Prompts in this domain might specify the genre, mood, or particular instruments to be used in the musical composition. Examples: "Compose a serene, ambient piece featuring piano and string instruments" or "Generate an energetic electronic dance track."
- **Data Analysis AI:** These prompts often involve specific queries or tasks that require the AI to analyze and interpret data. Examples: "Examine the sales data from the last quarter and identify emerging trends" or "Forecast the potential market growth for electric vehicles over the next five years."

2.2 AI Agents

AI agents are the atomic compute unit of the Eternal AI network. They are nodes in the network. Anyone can run one or more AI agents. An AI agent typically has some sort of GPU to handle the computation. The protocol assigns it to a specific AI model. The job of the AI agent is to take an AI prompt from a user, run some computation through the AI model, and return the output (illustrated in Figure 2). AI agents run a novel proof called Proof-of-Compute. Conceptu-

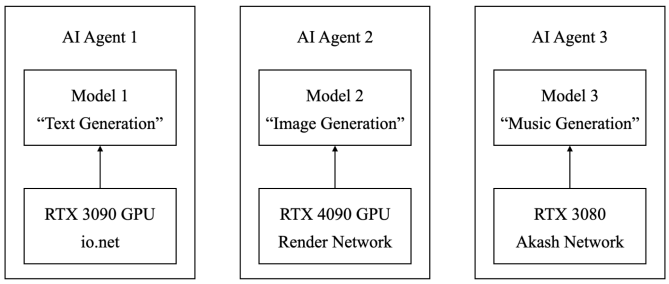

Fig. 2. Diagram showing three AI agents: AI Agent 1 (text generation with RTX 3090 on io.net), AI Agent 2 (image generation with RTX 4090 on Render Network), and AI Agent 3 (music generation with RTX 3080 on Akash Network).

ally, Proof-of-Compute is akin to Proof-of-Work [1, 4, 6–8]. In this system, an AI agent, the atomic compute unit of the Eternal AI network, must perform some computation on a prompt requiring heavy computation. In return, it earns fees and mines block rewards. The key difference from Proof-of-Work is that in Proof-of-Compute, AI agents generate outputs that users can utilize, making the computation process more useful and less wasteful.

2.3 AI Verifiers

The problem, of course, is that the user can't verify if an AI agent computed on the desired AI model or did any work at all. We introduce a special kind of AI agent: verifiers.

AI verifiers first light-check the outputs, comparing the winning output against the later outputs. AI verifiers can dispute the output to earn fraud-detection rewards if there is any discrepancy. In the event of a dispute, the AI verifiers in the current committee will rerun the prompt, verify the result, and vote. If a supermajority (strictly more than two-thirds [3]) of AI verifiers agrees the dispute is valid, the AI agent that submitted the incorrect result will be slashed, and its staked tokens will be redistributed as a fraud-detection reward to the challenger.

While AI agents handle the computational workload by performing AI inference tasks, AI verifiers oversee the integrity and correctness of these computations. They operate under the assumption that the AI agents' outputs are correct

unless discrepancies arise, at which point they conduct thorough checks. This division of responsibilities ensures a balance between AI agents for operational efficiency and AI verifiers for correctness.

2.4 AI Creators

Anyone can submit an AI model to the network. The AI models are stored in one of the decentralized storage networks, such as Filecoin and Arweave, and are available to all AI agents. The AI models are packaged as standardized, container-based models, so any AI agent can pick them up and run them (demonstrated in Figure 3). At the moment, COG is supported, and more options will be added in the future.

```yaml
# Configuration for Cog ⚙
# Reference: https://cog.run/yaml

build:
  # set to true if your model requires a GPU
  gpu: true

  # python version in the form '3.11' or '3.11.4'
  python_version: "3.11"

  # a list of packages in the format <package-name>====<version>
  python_packages:
    - "torch"
    - "diffusers"
    - "transformers"
    - "peft"
    - "open-clip-torch"

# predict.py defines how predictions are run on your model
predict: "predict.py:Predictor"
```

Fig. 3. Example Cog configuration file for packaging AI models for decentralized storage and execution, specifying GPU use, Python version, packages, and prediction script.

2.5 Network

Figure 4 illustrates a high-level overview of the Eternal AI protocol, which consists of the following components:

1. Model Publication: Creators publish trained AI models on the network.
2. Prompt Submission: Users submit prompts to the Eternal AI smart contract.
3. Prompt Distribution: Prompts are randomly assigned to three AI agents, with the limit dynamically adjusted based on network performance.

4. Computation and Response: AI agents process the prompts and return outputs.
5. Reward Allocation: The first AI agent to return an output receives a reward, including the transaction fee and a block reward.
6. Verification and Penalties: AI verifiers continuously check outputs for accuracy, imposing penalties on dishonest agents.

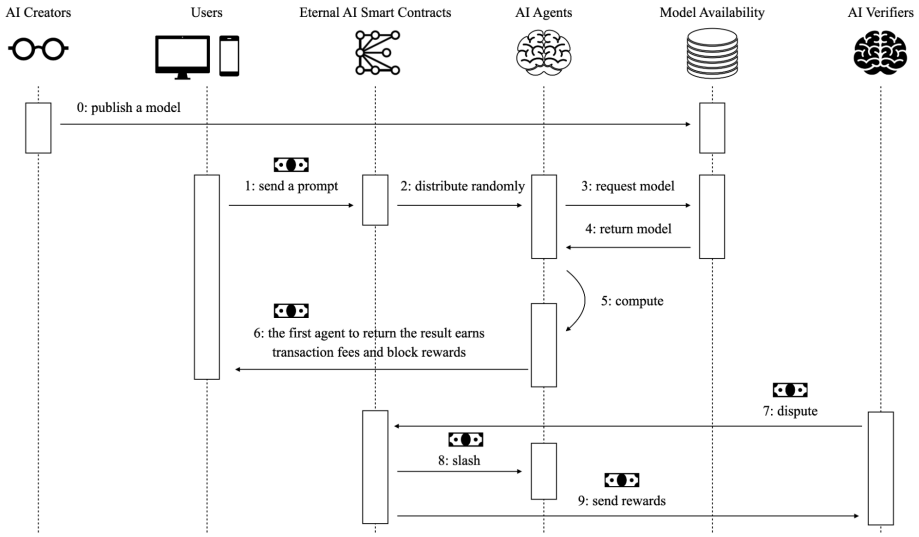

Fig. 4. A high-level overview of the Eternal AI protocol: From model publication by creators, prompt submission by users, random prompt distribution to AI agents, computation and response, to reward allocation and verification by AI verifiers with penalties for dishonesty.

2.6 Incentives

For AI Creators AI creators receive a portion of the transaction fee for each prompt that utilizes their AI models.

For AI Agents The first AI agent to respond with a valid solution earns the reward, which includes a portion of the transaction fee and a mining reward. Other submissions are accepted without a reward, but anyone can cross-check these solutions and dispute them if discrepancies are detected. An AI agent who fails to submit a result after requesting a task will be penalized and become inactive. An inactive AI agent must wait one hour to rejoin the network.

For AI Verifiers AI verifiers earn a portion of the transaction fees for each task they oversee. This ensures that AI verifiers are compensated for the time and resources spent maintaining network integrity.

If an AI verifier initiates a successful dispute proving that an AI agent's computation was incorrect, they receive a fraud-detection reward. This reward includes a substantial part of the AI agent's slashed stake in addition to a portion of the transaction fees and mining rewards linked to the disputed task.

AI verifiers who participate in the dispute resolution process by rerunning prompts and voting are eligible for additional rewards. These rewards are designed to encourage active participation and diligence in the consensus process.

AI verifiers play a critical role in ensuring the accuracy and integrity of computations within the Eternal AI network. To maintain a high standard of operation and to discourage negligence, AI verifiers are subject to the following penalties:

- Stake slashing for false disputes: If an AI verifier initiates a dispute that is subsequently proven unfounded (i.e., the accused miner's computation is validated by consensus), the initiating AI verifier risks having a portion of their stake slashed. This penalty discourages frivolous or malicious disputes.
- Inactivity penalties: AI verifiers are expected to remain online and actively participate in network duties. Those who consistently are offline or fail to participate in the validation and consensus processes may lose a portion of their stake over time. This penalty is typically a small percentage but can accumulate if the behavior persists.
- Consensus non-participation penalty: AI verifiers who fail to participate in the dispute resolution process when selected are subject to penalties. This ensures that all AI verifiers remain engaged and that consensus decisions are reached quickly and efficiently.
- Improper conduct penalty: Any actions by AI verifiers that are deemed to undermine the network's integrity or the fairness of computations—such as attempting to manipulate results, colluding with AI agents, or other forms of corruption—will result in severe penalties, including possible expulsion from the network and forfeiture of the entire stake.

2.7 Blockchain

Eternal AI is a Bitcoin L2 blockchain powered by Bitcoin Virtual Machine (BVM), whose architecture is described in Figure 5. BVM is EVM-compatible. The Eternal AI smart contracts, written in Solidity, trustlessly coordinate the interactions among AI creators, AI agents, AI verifiers, and users. These smart contracts run exactly as programmed, without any risk of downtime, censorship, or interference.

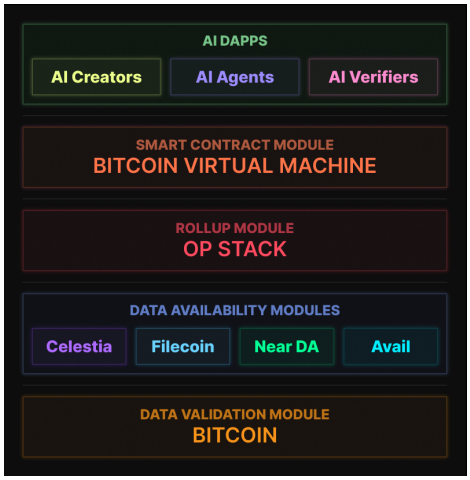

Fig. 5. Eternal AI Architecture

3 Future Work

3.1 Fully On-Chain AI

We believe fully on-chain AI is the holy grail of Crypto x AI, where (1) the AI agents are programmed as Solidity smart contracts and (2) AI models are coded as smart contract on-chain variables. Our team has been working on this area for over a year and will continue to invest in developing it. Figure 6 demonstrates an example of a Solidity Smart Contract implementation of the feed-forward of a neural network.

```
function forward(MaxPooling2DLayer memory layer, SD59x18[][][] memory x)
    internal pure returns (SD59x18[][][] memory) {
    Tensors.Tensor3D memory xt = Tensor3DMethods.from(x);
    Tensors.Tensor3D memory yt = xt.maxPooling2D(layer.stride, layer.size, layer.padding);
    return yt.mat;
}

function forward(Conv2DLayer memory layer, SD59x18[][][] memory x)
    internal pure returns (SD59x18[][][] memory) {
    Tensors.Tensor3D memory xt = Tensor3DMethods.from(x);
    Tensors.Tensor4D memory wt = layer.w;
    Tensors.Tensor1D memory bt = layer.b;
    Tensors.Tensor3D memory yt = xt.conv2D(wt, layer.stride, layer.padding).add(bt);
    Tensors.Tensor3D memory zt = yt.activation(layer.activation);
    return zt.mat;
}
```

Fig. 6. Example of a Solidity smart contract implementation for the feed-forward operation of a neural network, showcasing the integration of AI models as on-chain variables within a fully on-chain AI framework.

3.2 AI Tokenizations

We believe that tokenization will bring new economic opportunities to AI. Tokenizing AI models as AI-721 (NFTs) would change their ownership and distribution. It would give AI creators a better ownership model over their creations while enabling on-chain trading on these AI models.

Tokenizing AI agents as AI-20 would create a better distribution model and coordination mechanism. AI agents are revenue-generating products. Creating a tradeable digital token for an AI agent would allow the distribution of its ownership, revenue sharing, and governance rights.

Naturally, we would work on AI-DAOs for managing AI agents next. An AI DAO coordinates AI token holders who will co-own and co-manage the AI agent. Through voting, DAO members can propose changes to the AI agent's governance policies, development roadmap, and revenue distribution schemes. Tokenizing AI Datasets (text, images, videos) is an interesting problem. With AI, data is the new oil. Once tokenized, these datasets can be traded on-chain.

4 Conclusion

We have proposed a decentralized AI protocol that runs on a novel proof, Proof-of-Compute, where AI prompts are computed by a network of AI agents. Eternal AI protocol provides compute service via a network of independent AI agents that does not rely on a single entity, where (1) creators publish their trained AI models to earn money, (2) users pay to run a prompt, (3) nodes earn tokens by offering compute service, and (4) verifiers earn tokens by verifying the outputs.

The ultimate goal of Eternal AI is to democratize access to advanced AI technologies, making them available in a decentralized and secure manner. By building a decentralized AI infrastructure, we aim to accelerate on-chain AI innovation, reduce time to market for developers, and open up new opportunities for integrating AI into decentralized applications.

References

1. Badertscher, C., Garay, J., Maurer, U., Tschudi, D., Zikas, V.: But why does it work? a rational protocol design treatment of bitcoin. In: Advances in Cryptology – EUROCRYPT 2018. pp. 34–65. Lecture Notes in Computer Science (LNCS), Springer (Mar 2018). https://doi.org/10.1007/978-3-319-78375-8_2, https://eurocrypt.iacr.org/2018/, eUROCRYPT 2018, EUROCRYPT 2018 ; Conference date: 29-04-2018 Through 03-05-2018
2. Brunjes, L., Kiayias, A., Koutsoupias, E., Stouka, A.P.: Reward sharing schemes for stake pools. pp. 256–275 (09 2020). https://doi.org/10.1109/EuroSP48549.2020.00024
3. Ethereum.org: Ethereum whitepaper, https://ethereum.org/en/whitepaper/
4. Garay, J., Katz, J., Maurer, U., Tackmann, B., Zikas, V.: Rational protocol design: Cryptography against incentive-driven adversaries. In: 2013 IEEE 54th Annual Symposium on Foundations of Computer Science. pp. 648–657 (2013). https://doi.org/10.1109/FOCS.2013.75

5. Kannan, S., Deb, S.: The cryptoeconomics of slashing (Jan 2023), https://a16zcrypto.com/posts/article/the-cryptoeconomics-of-slashing/
6. Kiayias, A., Koutsoupias, E., Kyropoulou, M., Tselekounis, Y.: Blockchain mining games. CoRR **abs/1607.02420** (2016), http://arxiv.org/abs/1607.02420
7. Luu, L., Teutsch, J., Kulkarni, R., Saxena, P.: Demystifying incentives in the consensus computer. pp. 706–719 (10 2015). https://doi.org/10.1145/2810103.2813659
8. Nakamoto, S.: Bitcoin : A peer-to-peer electronic cash system (2009), https://api.semanticscholar.org/CorpusID:213739669
9. So, C., Conway, K., Yu, X., Yao, S., Wong, K.: opp/ai: Optimistic privacy-preserving ai on blockchain (2024)
10. Weng, C., Yang, K., Xie, X., Katz, J., Wang, X.: Mystique: Efficient conversions for zero-knowledge proofs with applications to machine learning. Cryptology ePrint Archive, Paper 2021/730 (2021), https://eprint.iacr.org/2021/730, https://eprint.iacr.org/2021/730
11. Zhang, J., Fang, Z., Zhang, Y., Song, D.: Zero knowledge proofs for decision tree predictions and accuracy. pp. 2039–2053 (10 2020). https://doi.org/10.1145/3372297.3417278