Token Bonding AI Agentic Models (Draft)

$\sim ronseg - hacsym$

December 26th, 2024

Abstract

This paper presents a novel approach to token bonding with AI agent models, both individually and in a weighted network of categorical use cases. The approach enables network participants to stake on future outcomes through a pre-configured bonding curve that amplifies early purchasers' returns. The bonding curve is verifiably computed and incorporates AI success metrics that impact the overall token supply. By leveraging existing technologies in the global sphere and analyzing current trends in both cryptocurrency prices and OpenAI's GPT-3 model progress towards AGI, we observe two worlds converging, creating an output differential in the competitive market that can be represented as $-N \leftrightarrow +N$.

1 Introduction

The fundamental concepts are listed below that outline why these technologies in aggregate make this a suitable adventure to undertake.

Infrastructure

- Digital Signatures
- Timestamping
- NFTs
- Are.na Connections

Technical Stack

- AI Models & LLMs
- WASM
- Jetson Hardware
- OTA

Core Features

- Block Rewards
- Bonding curves
- H-index Equivalency

2 Infrastructure

2.1 Digital Signature Implementation

The system implements cryptographic signatures using Ethereum's ECDSA (secp256k1) method, where each model update and data contribution is signed as $Sig = (r, s, v) = Sign(H(m)||k_{private})$, with $H(m)$ representing the Keccak-256 hash of the message and $k_{private}$ being the contributor's private key. For enhanced cross-chain compatibility, the system also supports Cosmos's Secp256k1 signature scheme through the tendermint/crypto module, enabling signature verification across both ecosystems through the standardized format $Verify(pubkey, H(m), Sig)$, where signatures can be validated against either chain's native verification methods.

2.2 Are.na Integration for Model Distribution

The platform implements a distributed knowledge sharing system inspired by arXiv's academic distribution model, leveraging Are.na's connection-based architecture to create a rich network of model implementations and their relationships.

2.3 Secure Model Training and Token Distribution Protocol

The decentralized training and token distribution process follows a cryptographically secure workflow. Initially, participants generate private keys and sign attestation messages to establish their identity within the network. A leader node is democratically selected through a verifiable random function (VRF) to orchestrate model training operations, ensuring both computational security and data integrity. The system utilizes a random beacon mechanism sourced from a central entropy store to prevent manipulation of training sequences. A central relayer facilitates purchase operations, providing onchain verifiable proofs that can be validated by authorized nodes.

The core computation leverages WebAssembly (WASM) modules that incorporate signed keys for secure execution. These modules process training data signatures and analyze Net Promoter Score (NPS) logs to generate a composite performance metric. The resultant signature determines share allocation quantities through the formula:

$$S_{allocation} = Hash(D_{training}||NPS)S_{max} \tag{1}$$

where:

- $D_{training}$ represents training data metrics
- NPS signifies aggregated user feedback scores
- S_{max} is the maximum share allocation

2.4 Beacon Emitter

For a minimial implementation, a central beacon emitter will be used to select the next machine to compute the model training update.

3 Technical Stack

3.1 AI Model Implementation

3.1.1 AI Explainability

AI explainability can be quantified by analyzing market liquidity across different segments, providing insight into model performance and market sentiment. This multi-faceted approach combines traditional metrics with market dynamics.

3.2 Over-the-Air Model Distribution via Fluence Network

The over-the-air (OTA) model distribution system leverages Fluence Network's decentralized protocol architecture to enable seamless transmission of model weights and training data through a network of peer-to-peer nodes, with each node implementing $E(K_{session})$ encryption for secure data transfer and $F_i(D_{chunk})$ operations for efficient data chunking. The system orchestrates distributed training across the network using differential updates and progressive loading mechanisms, where model weights are updated according to $\Delta W = \eta \cdot \sum_{k=1}^{K} \nabla L_k(W) \cdot Q(D_k)$, with η representing the learning rate and $Q(D_k)$ ensuring data quality across training nodes.

3.3 Edge Deployment on NVIDIA Jetson Hardware

3.3.1 Hardware Specifications

The model supports deployment on NVIDIA Jetson platforms with the following configurations:

Specification	**Jetson Nano**	**Jetson Xavier NX**	**Jetson AGX Orin**
GPU	128-core Maxwell	384-core Volta	2048-core Ampere
CPU	4-core ARM A57	6-core Carmel	12-core ARM v8.2
Memory	4GB LPDDR4	8GB LPDDR4x	32GB LPDDR5
Storage	16GB eMMC 5.1	16GB eMMC 5.1	64GB eMMC 5.1
Power	5-10W	10-15W	15-60W

Table 1: Jetson Hardware Specifications

4 Core Features

4.1 Model Block Rewards

Model operators who execute and maintain AI systems receive block rewards proportional to their computational contributions and the market value of the corresponding model's tokens. The reward mechanism follows a dynamic allocation formula $R_{block} = \alpha \cdot V_{market} \cdot P_{model}$, where α represents the operator's contribution coefficient, V_{market} is the market volume, and P_{model} denotes the model's performance metrics. This incentive structure ensures sustainable model operation while aligning operator interests with market success, as rewards are directly tied to both the model's utility and market adoption through automatic token distribution from the model's native market pool.

4.2 NFT-Based Data Provenance and Model Attribution

Non-fungible tokens (NFTs) serve as immutable records of data ownership and model contributions within the training ecosystem. Each NFT represents a unique data contribution or model enhancement, encoding metadata such as contribution timestamps, data quality metrics, and impact measurements through a hierarchical structure:

$$NFT_{metadata} = \{H(D_i), \alpha_i, t_i, \Delta M_i\} \tag{2}$$

where $H(D_i)$ is the cryptographic hash of the data contribution, α_i represents the quality score, t_i is the timestamp, and ΔM_i quantifies the model improvement. These NFTs form a verifiable chain of provenance through smart contracts, enabling transparent tracking of data lineage and attribution of model improvements. The system implements a merit-based revenue sharing model where downstream model utilization automatically distributes rewards to NFT holders based on their contributed data's impact.

4.3 Token Mechanics

4.3.1 Model Deployment Parameters

The token bonding mechanics are governed by a set of configurable parameters that determine the initial conditions and evolution of the system:

- **Supply Parameters**
  - Initial token supply (S_0)
  - Maximum token supply (S_{max})
  - Supply emission rate
  - Supply burn rate
- **Price Parameters**
  - Initial token price (P_0)
  - Price change rate limiter
- **Curve Parameters**
  - Curve type selection (linear, exponential, sigmoid)

- Angle coefficient (α)
- Curve adjustment thresholds
- Rebound parameters

- **Distribution Parameters**
  - Reserve allocation percentage
  - Stakeholder allocation percentage
  - Vesting schedules
  - Lock-up periods

- **Network Parameters**
  - Minimum stake requirements
  - Network fee structures

4.3.2 Bonding Curve Mechanics

The bonding curve follows a sigmoid function modified by success metrics:

$$P(s) = \frac{P_0}{1 + e^{-\alpha(s-s_0)}} \cdot M(t) \tag{3}$$

where $M(t)$ is the success metric multiplier defined as:

$$M(t) = 1 + \sum_{i=1}^{n} w_i \cdot m_i(t) \tag{4}$$

with w_i being the weight of success metric $m_i(t)$.

4.3.3 AI Success Metrics

Success metrics are bounded in the range [0,1] and include:

- Model accuracy: $m_1 \in [0.7, 1.0]$
- Adoption rate: $m_2 \in [0, 1.0]$
- Performance improvement: $m_3 \in [0.8, 1.0]$
- Network growth: $m_4 \in [0, 1.0]$

4.3.4 Momentum and Volatility Control

To dampen price volatility while preserving market dynamics, a momentum factor $\mu(t)$ is introduced:

$$\mu(t) = \beta \cdot \frac{1}{T} \sum_{t-T}^{t} \frac{dP}{dt} \tag{5}$$

where:

- β is the smoothing coefficient (default: 0.7)
- T is the time window for momentum calculation (default: 24h)
- $\frac{dP}{dt}$ represents price change rate

4.3.5 Aggregated Market Matrix

The system implements a basket of tokens through a matrix composition:

$$P_{basket}(t) = \sum_{j=1}^{k} v_j \cdot P_j(t) \tag{6}$$

where:

- k is the number of tokens in the basket
- v_j is the weight of token j in the basket
- $P_j(t)$ is the price of token j at time t

Token Set Configuration

- Rebalancing period: 7 days
- Fee structure:
  - Trading fee: A%
  - Set creation fee: B%
  - Rebalancing fee: C%

5 Use Cases

Use cases and categories can be aggregated and assembled in unique asset classes for themetic investing.

Lifestyle	**Enterprise**	**Prediction**	**Technical**	**Generation**
AI Voice Agent*	Customer Service Calls Speech Synthesis*	Stock Prediction*	Quantum Error Correction	LLM Generalized Intelligence
Weather Forecasting	Banking Fraud Time Series	Sales Prediction	Reinforcement Learning Feature Engineering	LLM Twin*
Crop Water & Nutrient Forecasting	Manufacturing Robotics*	Social Media Recommendation*	Microservice Logs Fault Prediction	Neural Style Transfer
Energy Grid Demand (Pricing)*	WWW Scraping Actions	Media Recommendation	Direct Preference Optimization Models*	Music Generation
Credit Scoring*	Scheduling	Housing Prediction*	Quantum Circuit Optimization	Chemistry Production*
Humanoid Robotics	Process Automation	Market Trend Analysis	Neural Architecture Search	Product Generation
Disease Outbreak Prediction	Supply Chain Optimization	Traffic Flow Prediction	P2P Randomization Segmentation*	Image Generation*
Gaming NPC Bots	Resource Allocation	User Behavior Prediction	Distributed Training Systems	Video Generation*
Smart Home Automation	Quality Control Systems	Cryptocurrency Trends*	Edge Computing Optimization	Text-to-3D Generation*
Classification				
1. Chatbot Classification of Intent				
2. Photo Classification				
3. Video Classification Security				
4. Biometric Classification				
5. Network Traffic Classification				
6. Health Records Diagnosis				
7. Document Classification				
8. Audio Classification				
9. Sentiment Analysis				

Table 2: AI Applications by Category