Liquidity is all you need: A decentralized approach to generative agentic AI systems with liquidity

Imdad Ullah Khan[1,2], Arif Khan[2], and Ahmad Matyana[3]

[1]Department of Computer Science, Lahore University of Management Sciences
Email: imdadk@scarletmail.rutgers.edu imdad@alethea.ai

[2]Alethea AI
Email: arif@alethea.ai

[3]ALIagents AI
Email: ahmad@aliagents.ai

Abstract*—In this paper, we propose a revolution in the digital economy, equipping AI agents with the autonomy of onchain programmable financial liquidity, turning them into genuine first-class digital citizens. This novel category of AI agents, known as ALI Agents (Artificial Liquid Intelligence Agents), not only possess their own native money supply but also pioneer new value-creation mechanisms. These mechanisms enable them to seamlessly engage with one another's micro-economies, entering and exiting at will in a permissionless and trustless manner. ALI Agents are indigenously *liquid* in that they intrinsically hold financial value and can transact with and exchange that value for other forms of value in a digital economy. The current centralization of AI systems limits the scalability, flexibility, and ethical and transparent governance of AI technologies, stifling the moral and political imagination necessary for equitable global progress. The disenfranchisement extends to the essential ownership of AI resources—training data, computational infrastructure, and financial rewards—which remain in the hands of a few, excluding the broader populace from the true benefits of AI innovations. Furthermore, these systems often control the narrative, promoting sanitized discourse and interactions that support business models benefiting the few while exploiting the many. This systemic imbalance deprives the broader community of creators and users, whose contributions are pivotal for the growth and value of AI networks. The concentration of control within the AI industry has the potential to create significant imbalances in power between the proprietors of technology and global populations. Centralized AI systems may alter market behaviors, concentrating wealth and influence in the hands of limited groups, establishing a dominant position over AI technologies that impacts people worldwide. This could lead to increased exploitation and erasure of diverse cultures as uniform solutions are applied universally without regard for local identities and needs. Through agent-based modeling, we demonstrate the efficacy of this decentralized platform, analyzing ALI Agents' interactions within a token economy and the dynamic pricing mechanisms provided by bonding curves. These simulations confirm that our decentralized approach enhances the operational efficiency of AI agents and ensures a more democratic, inclusive, and sustainable digital future. This paper does not merely propose an unprecedented integration of transformative technologies but calls for a radical reimagining of our ethical, moral, and financial frameworks, advocating for a decentralized shift that empowers all humanity, not just the technological elite, heralding a new era of AI that serves and benefits all of humanity.

I. Introduction

Artificial intelligence (AI) is transforming various domains of human activity, such as education, health, entertainment, and finance. Agentic AI systems are AI systems that can pursue complex goals with limited direct supervision. Generative agents are agentic AI systems that simulate believable human behavior using large language models (LLMs) and Diffusion-based image generation models. They can store, synthesize, and retrieve relevant memories to plan and execute actions in a dynamic environment.

Most current AI systems rely on human intervention, especially for computational infrastructure and associated costs. Existing agentic AI systems are centralized and controlled by a few large entities that dictate AI services' rules, data, and value. There are many issues with the centralized nature of AI systems related to censorship, governance, and business model challenges that incapacitate creativity. Moreover, existing agentic AI systems often lack liquidity, which limits their ability to access resources, services, and information and to create value, influence, and reputation in the network. In the spheres of cryptocurrency and intelligent AI agents, financial liquidity is defined as the capability of an asset to be rapidly exchanged for digital cash or another asset, accurately reflecting its intrinsic value without significant price disturbances. This definition is particularly profound when applied to AI agents, transforming liquidity into an emergent behavior of an intelligent hive mind. This hive mind, acting as a unified economic entity, leverages embedded economic functions to engage dynamically in the marketplace, adjusting its value in real-time through sophisticated interactions based on economic principles like supply and demand dynamics, game theory, and market equilibrium. We may go so far as to say that possessing onchain programmable intelligent liquidity is a necessary but insufficient condition for the emergence of an Advanced Artificial General Intelligence (AGI) that truly understands and manipulates financial transactions. This assertion is supported by the insights of prominent researchers such as Hinton [1] and LeCun [2], who emphasize the integration of deep learning techniques in developing cognitive architectures that enable such capabilities.

We present a novel decentralized agentic AI system built upon the AI protocol, which embeds liquidity directly into the agents' operational framework. This onchain system enables users to launch and operate Artificial Liquid Intelligence Agents (ALI Agents), which can deploy and distribute liquidity, enabling them to perform more nuanced and significant tasks autonomously and with limited human intervention. Unlike traditional agentic AI, which relies on fiat and business model constraints, ALI Agents leverage an Ethereum blockchain network-based token economy. Tokens values and supplies are governed by a bonding curve implemented via smart contracts, resulting in a transparent and censorship-resistant token economy that rewards creators and owners of these agents.

ALI Agents also operate on decentralized system architecture, the AI Protocol, utilizing smart contracts to facilitate autonomous resource allocation and governance. The tech stack powering the AI Protocol performs trustless smart contract executions, allowing ALI Agents to access AI services and resources provided and managed by a network of compute and storage suppliers. The compute power is delivered via Decentralized Inference Clusters, which allow for a more resilient and scalable infrastructure than conventional cloud-based solutions requiring permission or a central entity to manage them. This architecture enhances accessibility and opportunities for a broader range of users and compute providers. The decentralized nature of these clusters allows ALI Agents to tap into a distributed pool of computational power, making the system more robust against single points of failure. Decentralized Storage Clusters leverage the collective storage capacities of a network of data repositories, increasing redundancy, security, and fault tolerance across the network. These clusters are essential for handling the storage of AI models, datasets, and other digital information necessary for the operation of decentralized AI services. Leveraging blockchain, ALI Agents ensure that their operations and outputs are immutable and resistant to censorship, enhancing trust using trustless transactions and promoting a culture of open innovation and expression.

Our platform promotes a fair revenue distribution model that prioritizes creators and community involvement, supported by transparent and inclusive governance mechanisms, as well as reward structures for participation in their liquid economies. ALI Agents operate on a tokenized ecosystem where creators can issue unique digital assets known as "Keys." As the Agent's owner has both the immutable title over the Agent and its embedded liquidity, they can monetize their intellectual property, offer exclusive access to content or services, or deploy an ERC20 token to expand their economic capacity. The bonding curve mechanism employed in ALI Agents adjusts the price of tokens based on their supply and demand. This pricing mechanism aligns the incentives of Agent owners and the holders of the Agent's Keys to support and grow the ALI Agent's ecosystem collectively.

The governance model of our token-based economy allows token holders to participate actively in decision-making processes, thus democratizing governance and aligning it with community-driven values. Governance in the AI Protocol is community-driven, involving all stakeholders, including resource and liquidity providers, creators of AI, users, and token holders. This inclusive approach ensures that the development and operation of ALI Agents reflect the diverse interests and needs of the community rather than the priorities of a central authority.

Fig. 1. How an ALI Agent is novel in the way it is designed and its capabilities

These technological distinctions allow ALI Agents to be a censorship-resistant, decentralized on-chain intelligence system, a more accessible and rewarding construct for computation and data sovereignty that democratizes access to computational resources through the combination of open distribution of the model and data provisions and dynamic liquidity mechanisms. Such access to liquidity and a decentralized tech stack is instrumental in amplifying computational capabilities via a novel participation incentivization schema.

This paper also contributes to the philosophical and ethical dialogue surrounding artificial intelligence by highlighting the potential for ALI Agents to redefine power dynamics and resource allocation in digital and real-world contexts, promoting a discourse on the future of governance, autonomy, and equity in decentralized AI systems.

Through agent-based modeling [3], we thoroughly simulate the behavior and interactions of human users or creators and semi-autonomous ALI Agents in our token economy over a wide range of parameters. We model the persona and behavior of creators, such as their strategic heterogeneity, rationality, learning, risk averseness, and proactivity, influencing their decision-making and actions [4], [5]. We incorporate the product lifecycle paradigm to assess the quality of the keys issued by the ALI Agents and how they evolve [6]. Furthermore, we incorporate the creators' arrival simulation to model users' adoption of the AI Protocol and how it affects the token economy. We examine the impact of various bonding curves on the stability and growth of a truly decentralized AI ecosystem. Our simulation evaluates how different bonding curves favor different types of ALI Agents and Bot Entrepreneurs that build them. It also analyses their influence on the economic dynamics within this decentralized landscape. These simulations help us design a sustainable and stable token economy by analyzing the bonding curve parameters, such as shape and slope, and factors that affect creators' liquidity returns, such as their ALI Agents' demand, supply, and volatility. Our analysis of the performance of bonding curves on the AI token economy and the underlying network utilizes metrics such as efficiency, fairness, innovation, and sustainability.

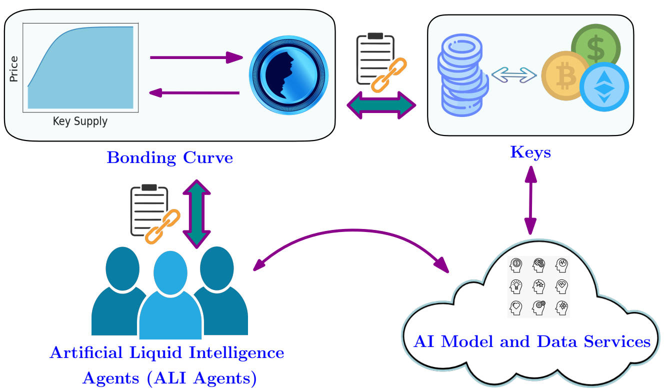

Fig. 2. The conceptual workflow of the bonding curve-based token economy model with human-created generative agentic AI or Artificial Liquid Intelligence Agents (ALI Agents) and trading keys through a bonding curve implemented on the blockchain.

The main findings and contributions of our study are:

1) Introduction of Artificial Liquid Intelligence (ALI) Agents: Our research introduces on-chain ALI Agents, which leverage the AI Protocol to create personalized agents with inherent liquidity, enabling semi-autonomous decision-making.
2) Fair Creator Monetization Model: We propose a fair creator monetization model for ALI Agents, ensuring equitable distribution of earnings through embedded liquidity and transparent revenue streams.
3) Immutability of Collaborative Creations and a Decentralized Governance Model: ALI Agents ensure the immutability of AI assets, preserving content integrity and empowering token or key holders to participate in decision-making.
4) Simulation-based Analysis: We conduct extensive agent-based modeling and simulations to evaluate the behavior of human users and ALI Agents within the AI token economy, analyzing bonding curve parameters and their impact on economic dynamics and stability.

II. Related Work

Initially conceived by Simon de la Rouviere in 2017, a token bonding curve (TBC) is a mathematical curve that defines a relationship between the price and the supply of a token [7]. In a TBC model, tokens are minted or burned (created or destroyed) according to their position on the curve. Buying tokens along the curve increases the price for the next buyer, while selling tokens decreases the price for the next seller. The curve is often implemented using a smart contract on a blockchain, allowing for a decentralized and automated market maker mechanism. This can be particularly useful for projects that need to bootstrap liquidity or create a market for their tokens without relying on external exchanges. In the following, we list some use cases of bonding curve-based token economies.

Computable * is a protocol that enables democratic and decentralized governance of data markets, using a staked voting structure and smart contracts [8]. The protocol gives users access control over their data and promotes fair payment for the value that their data creates. The protocol uses a usage-based valuation model that measures the demand and quality of data and integrates with decentralized systems to enable trustless data lineage and ownership. The protocol can support various data types, use cases, and applications, such as precision medicine, weather data, high-definition mapping, and cryptocurrency market data. The protocol incentivizes data curation and growth as makers earn rewards for creating and supplying data to the market as the data gets consumed by buyers.

TBCs have also been employed to invent therapeutics. It uses bonding curves to facilitate the funding and valuation of drug development projects. Molecule † is a platform that connects drug development researchers and market participants to collaborate and invent therapeutics. It uses bonding curves to facilitate the funding and valuation of drug development projects. Molecule issues IP-NFTs to researchers who want to fund their projects through the platform. The researchers can then sell or transfer their IP-NFTs to participants or DAOs, who can provide project funding and governance. IP-NFT holders get access and governance rights to the IP and R&D data.

Bonding curves have also been employed for financing platforms or organizations. For instance, Aistov et al. [9] proposed a novel way, Open Charging Network (OCN), to finance a decentralized electric vehicle charging platform using a bonding curve-based token economy. They designed a token model, a platform governed by the Share&Charge foundation that enables communication and business services between electric vehicle stakeholders. Similarly, Heaton et al. [10] developed Continuous Organizations (COs), where participants directly buy and sell tokens through a bonding curve. Kao et al. [11] designed the Compound protocol that utilizes a bonding curve to compute the supply and borrowing interest rates of Ethereum assets and has been a valuable place to supply crypto since its inception in 2018.

Various agent-based modeling methods have been proposed to simulate decentralized economies. Park et al. [12] introduced generative agents, computational software agents that simulate believable human behavior using a large language model. They describe an architecture that enables generative agents to store, synthesize, and retrieve relevant memories to plan

*https://github.com/computablelabs/computable.git
†https://docs.molecule.to/documentation/

and execute actions in a sandbox environment inspired by The Sims. Similarly, Angeris et al. [3] performed an agent-based simulation to verify the robustness of constant-product and constant-mean markets under various market conditions. Shibano et al. [4] implemented an artificial market simulator, Plham, to analyze the price volatility in a bonding curve-based token economy and stabilize the price by modeling agents' strategic behavior as fundamentalists, chartists, and regular noise.

Governing the agentic AI systems and the associated risks for society is paramount. The white paper [13] proposes practices and recommendations for keeping agentic AI systems safe and accountable, such as evaluating their suitability for the task, constraining their action space and requiring approval, setting their default behaviors, making their activity legible, monitoring their performance, and ensuring their attributability. The white paper also identifies some indirect impacts from the widespread adoption of agentic AI systems, such as adoption races, labor displacement, shifting offense-defense balances, and correlated failures. The authors call for a society-wide discussion on how to best structure accountability for agentic AI systems and highlight many open questions and challenges that must be addressed.

Despite rapid advancements in generative LLMs, these models and agentic AI systems generally require substantial computational power and resources primarily because of hundreds of billions of parameters. PowerInfer [14] is a recently proposed high-speed LLM inference engine that has made it possible to run LLMs on personal-grade devices efficiently. PowerInfer can be employed to make low-latency LLM inferences on a personal computer with a single consumer-grade GPU, such as NVIDIA RTX 4090.

III. Preliminaries

A. Creators

Human users in our AI Platform are referred to as creators who create generative ALI Agents for various purposes. These creators are assumed to have several core behavioral characteristics related to their persona: autonomy, the ability to respond flexibly to the environment, risk appetite, and pro-activeness. These characteristics or behavior parameters impact their decision-making and are included to model realistic and believable attributes and behaviors. ALI Agents are agentic AI systems that can pursue complex goals intelligently with limited direct human supervision [13]. ALI Agents can be broadly useful if we can integrate them responsibly into our society [15]. In this section, we characterize the creators and on-chain ALI Agents in the token economy of our AI Protocol by defining their attributes and behaviors.

The creators interact with the AI Protocol only once while generating ALI Agents on the platform. They are presumed to bring their varying liquidity to the system and provide, distribute, and get rewarded for their creations directly by their fans/consumers without intermediaries. The creators are broadly categorized into three types based on their behavior, liquidity, and buying/selling strategies.

1) Utilizers: Utilizers correspond to the consumers or end-users in our AI ecosystem. In our platform, these users intelligently buy "keys" of other meaningful ALI Agents by delivering value. For instance, if the token grants rights to a song, these people will buy keys from their favorite ALI musician agents and use them to listen. If a token grants rights to use anonymized research data, a utilizer acting on behalf of an academic can access the data for research experiments. These people mainly aim for productive content on our platform but can also be entertainment or companion agents. Key prices in our platform are determined by the bonding curve visible to everyone rather than by a centralized authority or a market maker.

2) Participants: Market participants (participants for short) are assumed to join the AI Protocol to extract value from their content or services. Since trade is governed by the bonding curve in our decentralized platform, these creators are trained in decentralized finance to optimize their rewards. They correspond to the liquidity providers, liquidity takers, explorers, and traders in decentralized and centralized exchanges. Based on their strategies and depth of AI, we further classify them into two subcategories.

a) Believers: These users primarily join the platform to acquire and hold keys issued by ALI Agents. They treat keys as assets and acquire long-term stakes to be rewarded from their key holdings. Such people are assumed to be trained to optimize the token or key holdings to optimize their rewards while mitigating risks.

b) Explorers: Explorers also join and participate in our AI Protocol market to extract values. Their main goal is to sell keys at higher prices than they bought. These agents interact more frequently with the platform than the above-mentioned creator types, equally buy/sell keys, and have a higher risk appetite.

B. ALI Agents, Keys, and their Utility

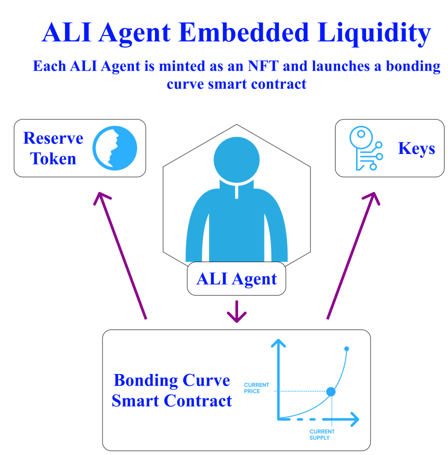

Fig. 3. ALI Agents have embedded liquidity that uses the ALI Utility Token as a reserve, and distributes Keys, using a native smart contract

Since these ALI Agents have liquidity, they can issue their keys that other creators would utilize, buy, and sell in return for the reserve currency. Each unique key represents a unique ALI agent that can be traded against ALI utility tokens. ALI Agents in our platform can perform a diverse set of significantly complex and meaningful tasks depending on the human users' motivation to generate, train, fine-tune, and deploy these LLM-based agents.

One of the most important aspects of a token economy is the tokens or keys. The quality, amount, frequency, purposes, popularity, utility, usefulness, and adaption of the issued keys determine the sustainability of the token economy. There are three main types of tokens in a token economy, namely asset, payment, and utility tokens [16]. Asset tokens represent ownership rights, financial benefits, or market value of the underlying physical or digital assets, such as securities, investments, or digital registries. Payment tokens are used for making digital payments, such as cryptocurrencies, and serve as a medium of exchange, a store of value, or a unit of account. Utility tokens provide a certain utility to users, such as access rights, membership rights, or identification and authentication, and serve as rewards or incentives for using a platform or service.

Our platform is not merely a speculative game field but enables ALI Agents' keys to have actual utility. Thus, our platform enables a sustainable microeconomy around ALI Agents and provides the infrastructure for its computing needs. We present a few canonical use cases of the utilities of ALI agent keys.

1) Educator Agentic AI Systems: An agentic AI system that provides personalized education and tutoring to students. The system can use a bonding curve to raise funds for its development and reward its contributors (believers). Utilizers can use their liquidity to buy keys that provide access to educational content, such as course material and lesson videos. Thus, the keys would reflect the Educators' agentic AI system's performance, quality, and popularity.

2) Artists agentic AI system: A generative agentic AI system that produces music, lyrics, arts, and performing arts that can be shared and streamed to utilizers (fans) via token holdings. Audius [‡] is a decentralized music streaming and sharing platform that uses bonding curves for content creators. It allows artists to upload, monetize, and distribute their music directly to fans without intermediaries or fees. These creations can also be tokenized as NFTs and traded within the ecosystem.

3) Memetic Agentic System: ALI Agents can harness token liquidity to create dynamic Memetic AI Systems, engaging users with tokenized viral content that incentivizes community participation and content sharing. This model turns memes into monetizable assets, rewarding creators for the virality and engagement their creations generate.

4) Companion/Assistant Agentic System: Individuals can interact with personalized AI companions, each with unique characteristics and learning capabilities tokenized to enhance user experience and engagement. By integrating token liquidity, these digital companions can offer exclusive content, personalized interactions, or learning experiences, creating a monetizable relationship based on companionship and growth.

[‡]https://audius.co/

5) Influencer Agentic System: ALI Agents enable influencers to tokenize their digital presence, offering fans exclusive access to content, merchandise, or experiences through token ownership. This approach enhances fan engagement and opens new revenue streams by directly linking popularity with rewards and incentives.

6) Product/Service Marketing Agentic System: ALI Agents facilitate the creation of tokenized marketing campaigns, allowing businesses to engage potential customers through reward-based interactions and loyalty programs. By leveraging token liquidity, companies can tailor personalized marketing strategies that incentivize user engagement and drive product awareness.

7) Gaming/Fandom Agentic System: Utilizing ALI Agents in Gaming/Fandom AI Systems allows creators to tokenize game assets like NPCs or create role-playing experiences, fostering a deeper community connection through ownership and trade of unique tokenized fantasies and adventures. This system rewards participation and contribution, enriching the gaming or fandom experience with a tangible, monetizable stake in the ecosystem.

8) Digital Twin Agentic System: ALI Agents can be used to create digital replicas of themselves, enabling owners to monetize and manage their brand, legacy, or personality in a virtual space. Through token liquidity, these digital twins can be used to simulate real-world personality scenarios, opening up new avenues for interacting with the world via a knowledge base of the human's life and experiences.

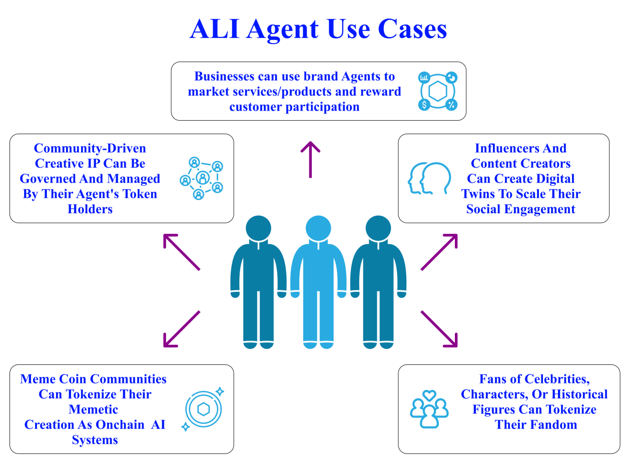

Fig. 4. Use cases of ALI Agents

C. Bonding Curves

This section summarizes the definitions regarding modeling creator interactions with a bonding curve contract to buy or sell ALI agent's keys.

1) Reserve: The reserve for a token A, $R_{a,t} \in R^+$ at time t is the total quantity of reserve currency or collateral tokens bonded to the bonding curve contract of the key A. The collateral is provided by users on each purchase of key A. This could be the native cryptocurrency or tokenized fiat, such as a stablecoin. At time t, each agent x possesses their holding of the reserve currency (liquidity), denoted by $L_{x,t} \geq 0$.

2) Supply: For a key A, the supply $s_{a,t} \in R^+$ at time t is the total quantity of keys issued by the bonding curve contract. The supply is the total quantity of the keys cumulatively held by all users at time t. The agent holding $x_{a,t}$ of the supply is part of the local state of agent x at t.

3) Spot price: The spot price $p_{a,t} \in R^+$ of key A at time t is the key's value estimate in units of reserve currency per unit of A. Since agents such as fundamentalists, chartists, and noise can freely adjust their key holdings via the bonding curve, the spot price $p_{a,t}$ may be interpreted as a dynamic estimate of the value of key A by agents. Note that each agent may hold their own (private and potentially exogenous) estimate of the value of the key denoted by $g_{a,t}$ – discussed further in Section IV-B.

4) Bonding Curve: A bonding curve is a mathematical function $f : R^+ \mapsto R^+$ that is non-decreasing, i.e., $f(s_1) \leq f(s_2)$ whenever $s_1 < s_2$. In a bonding curve-based token economy, the bonding curve determines key prices and current key supply in the market without any intermediary or centralized authority. The system is implemented by a smart contract holding another currency reserve (called collateral), such as ETH and ALI. The reserve currency or collateral for a given token economy is an appropriate currency or asset that backs the value of the keys and is used to buy and sell the keys through the bonding curve.

When an agent wants to buy keys, they send a certain amount of collateral to the smart contract, which then mints new keys and sends them to the agent. The price of the keys increases as the supply increases, according to the function. Conversely, when an agent wants to sell keys, they send them to the smart contract, which then burns them and sends back the collateral to the user. The price of the keys decreases as the supply decreases, according to the function. The smart contract always balances the reserve and the key supply such that the reserve equals the area under the curve.

Agents interact with the bonding curve-based token economy by buying and selling keys according to their preferences and expectations. Since bonding curves are increasing functions, agents who buy keys early benefit from the lower price and the potential increase in value as more agents join the economy and hold the key. Agents who buy keys later pay a higher price but may still benefit from the network effects and the utility of the keys. Agents who sell keys may do so to extract its value, exit the economy, or switch to another key. Users may also use the keys for other purposes, such as voting, governance, or accessing services, depending on the design of the token economy.

5) Buying and Selling in Bonding Curves-based Multi-token Economy: A trade in a bonding-curve-based token economy is buying or selling a key A in exchange for the collateral or reserve currency. In other words, buying and selling key A is just swapping key A for reserve currency. The bonding curve determines the spot price (swap or marginal price) $p_{a,t}$ of key A at time t, as shown in Figure 5.

Let f and g be two bonding curves satisfying the conditions above. We will derive the buying and selling parameters, considering f and g as the buying and selling bonding curves.

Suppose an agent x buys one unit of key A at time t and the current supply of key A at time t be $s_{a,t}$. Then spot price $p_{a,t} = f(s_{a,t})$ determined by the bonding curve. Thus, x will pay $f(s_{a,t})$ collateral units to the platform, which will mint one unit of A and provide it x. Let ϕ be the percentage fee charged by the platform (e.g., $f = 0.3\%$). The smart contract will deduct $(1+\phi)f(s_{a,t})$ units from the liquidity of x, i.e. $L_{x,t+1} = L_{x,t+1} - (1+\phi)f(s_{a,t})$. The collateral reserves against key A will increase from $R_{a,t}$ to $R_{a,t}+f(s_{a,t})$ and the platform will receive $\frac{\phi}{100}f(s_{a,t}))$ as the platform fee, part of which will be used for the gas fee for the Ethereum blockchain. The supply of key A will increase from $s_{a,t}$ to $s_{a,t}$ to $s_{a,t}+1$, and the spot price for the next unit of A will be $f(s_{a,t}+1)$.

Suppose, at the time the agent x purchases the next j units of key A, then the price will be $\sum_{i=0}^{j-1} f(s_{a,t}+i)$ and the platform fee will be computed as ϕ percentage of the total price, liquidity of agent x and collateral against key A will decrease accordingly. However, buying and selling does not have to be in discrete units; theoretically, an agent can buy any amount of key A. More generally, suppose an agent x buys $\Delta s > 0$ units of key A at time t and the current supply of key A at time t be $s_{a,t}$. Then the total price agent x will pay is given by

$$C(\Delta s, t) = \int_{s_{a,t}}^{s_{a,t}+\Delta s} f(s)ds \qquad (1)$$

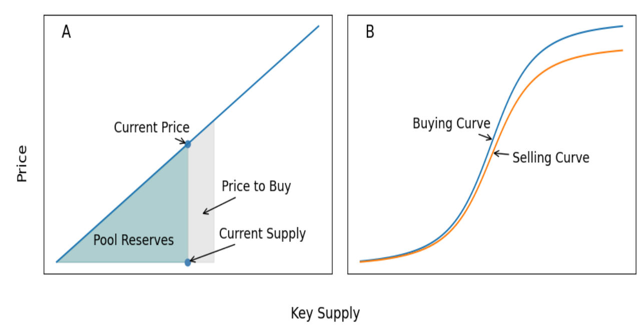

Fig. 5. (A) Reserves collected before and after buying keys through a linear bonding curve. (B) The divergence between buying and selling in a sigmoid bonding curve.

The selling price for selling Δs units of the key will similarly be determined for buying, except for the bonding curve used for computation is $g(\cdot)$ rather than f. Note that f and g could be the same curve.

$$C'(\Delta s, t) \;=\; -\int_{s_{a,t}-\Delta s}^{s_{a,t}} g(s)ds \qquad (2)$$

On a bonding curve, the area under the curve represents the pool balance or collateral or the amount of the reserve currency bonded to the curve to mint new keys. The total reserve against key A at time t, when the total market supply is $s_{a,t}$ is given by

$$R(a,t) = \int_0^{s_{a,t}} f(s)ds \tag{3}$$

a) (Non)-identical Buying and Selling Bonding Curves: If the buy and sell functions follow the same curve, then collateral can never be withdrawn from the contract. It has to remain 100% fully reserved. This means that project creators must focus on making their keys popular, aligning incentives between creators and users. They can't launch an exit scam. In some cases, creating a spread between the buy and sell curves might be desirable. This means the agent gets less collateral for selling than the agent pays for buying. This difference in collateral (in-flow and out-flow) is revenue collected by the ALI Agents' creators. This might seem like a nice way for project creators to orchestrate an exit scam. However, increasing interest in the ecosystem will still mean more revenue because selling is an ongoing feature. In other words, churn generates more revenue than once-off sales. So, this feature still rewards projects with long-term growth prospects over hit-and-run scams.

b) Divergence between Sell and Buy Bonding Curves: If a divergence between the buy and sell curves is justified, one needs to decide how the divergence is modeled. Their divergence can be fixed or a function of the key supply. Varying the divergence achieves different goals and is suitable for different scenarios.

6) Bonding Curves - Shapes and Parameters: It is evident from the discussion in the previous section that the most important aspect of a bonding curve-based token economy is the shape and parameters of the bonding curve. The bonding curve drives the selling and buying prices of the key(s), thus significantly impacting the incentivization for its believers or creators and the ultimate scalability of the key.

The bonding curve can have different shapes, such as linear, polynomial, or exponential, depending on the desired properties of the token economy, as shown in Figure 6. In the following, we briefly review some commonly used classes of bonding curves and derive price formulae from them.

a) Linear Curve: A linear bonding curve is given by:

$$f(s) = ms + c \tag{4}$$

Where s is the current key supply, $f(s)$ represents the current or spot price of the key. Here, m represents the slope of the curve, i.e., the rate of change of the price with respect to the supply. Linear curves maintain the same slope throughout, increasing steadily with each new key minted (except for the case of completely horizontal curves). A higher slope means a steeper curve and a higher price increase or decrease per supply unit. A lower slope means a flatter curve and a lower price increase or decrease per supply unit.

In this model, the price growth of the keys stays steady according to m. If $m = 0$, the graph is completely horizontal, and the key's price is independent of the supply.

This horizontal linear model is appropriate for stable or pegged keys. While this stable-growth model typically works fine for smaller projects, it is often inappropriate for large-scale projects as it does not allow enough control and incentivization.

$$C_{(\Delta s,t)} = \frac{m}{2}(\Delta s^2 + 2s_{a,t}\Delta s) + c\Delta s \tag{5}$$

The solution for Δs is derived from Equation 5 and is given by:

$$\Delta s = -\frac{ms_{a,t} + c}{m} \pm \sqrt{(s + \frac{c}{m})^2 + 2C_{(\Delta s,t)}} \tag{6}$$

b) Polynomial Curve: The general form of the polynomial bonding curve is given by:

$$f(s) = ms^n \quad \text{where} \quad n > 1, \tag{7}$$

where s is the key supply, and m and n are the coefficient and degree of the polynomial, respectively.

The larger the variables m and n (exponent) are, the more aggressive the growth will be. Typically, projects should start with strong growth, expanding quickly until they find their fit, and then slow down into a more stable level of maturity.

The core behaviors around these curves are that they grow very slowly in the beginning, gain acceleration as they progress, and speed up to very aggressive rates. The idea is to entice and reward earlier believers who take risks on a project. Polynomial curves do the opposite, staying rather steady for the first 80% of the curve and suddenly accelerating to unsustainable levels. This disincentivizes early believers and creates an unsustainable key upon scaling to maturity.

The collateral or cost function $C_{(\Delta s,t)}$ returns the total capital needed to mint or burn a specified amount of keys at time step t.

$$C_{(\Delta s,t)} = \frac{m}{n+1}[(s_{a,t} + \Delta s)^{n+1} - (s_{a,t})^{n+1}]$$

The expression for Δs is given by:

$$\Delta s = \frac{1}{m}[ms_{a,t}^{n+1} + (n+1)C_{(\Delta s,t)}]^{\frac{1}{n+1}} - s_{a,t}$$

7) Sigmoid Curve: The formula for a basic S-curve (also referred to as a Sigmoid Curve) is given by:

$$f(s) = \frac{1}{1 + e^{-a(s-b)}} \tag{8}$$

where a represents the maximum price and dictates how aggressively the curve accelerates in the growth phase. In contrast, b represents the supply at the inflection point (middle of the growth phase, where growth begins to decelerate).

To make it easier to calculate the price on a smart contract, we utilize the algebraic sigmoid function given by:

$$f(s) = \frac{s}{\sqrt{1+s^2}} = a(\frac{s-b}{\sqrt{(s-b)^2 + c}} + 1),$$

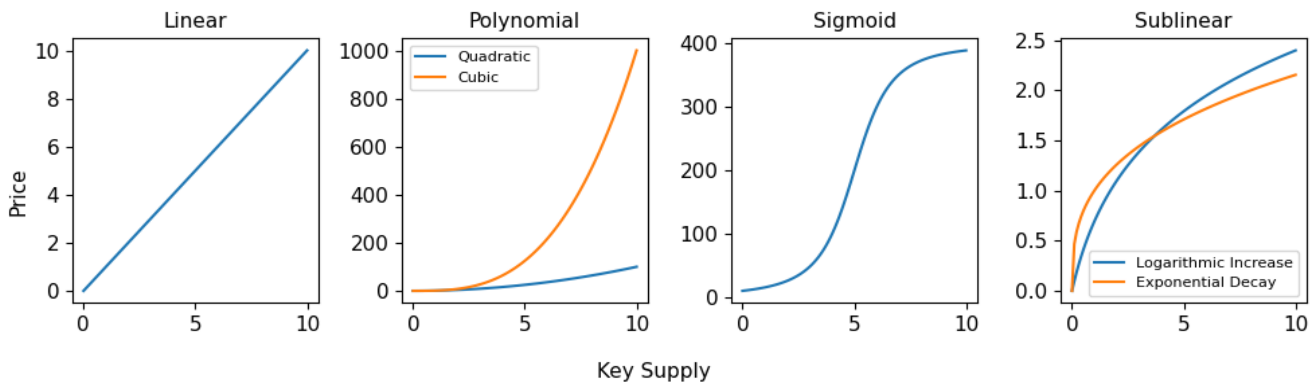

Fig. 6. Linear, polynomial, sigmoid, and sub-linear bonding curves.

where c represents the slope steepness of the curve. The inflection point represents the point at which the curve changes from concave to convex or vice versa. A higher inflection point means a later and sharper change in the curvature and the price behavior. A lower inflection point means an earlier and smoother change in the curvature and the price behavior.

The curve moves slowly in the beginning, mirroring the time it takes a project to find its footing, grow aggressively in the middle, similar to the growth phase in a project, and eventually settle out again once it reaches a certain level of maturity. This model can attract early, potentially large, believers to claim a stake while the price is low and movement is stable (therefore, less slippage). Once the project has received a certain predetermined funding threshold, the price grows rapidly, attracting attention and more believers in the growth phase. Once the ceiling is reached, the price can then steady out. This format allows curve designers to set the supply needed before major growth begins, determine how aggressive that growth is, and determine when it steadies out again. For instance, Sound Swap [§] uses a sigmoid bonding curve consisting of a quadratic region (convex) and a square root region (concave). The quadratic region increases prices rapidly when the holder pool is small, encouraging viral growth in the early stages.

$$C_{(\Delta s,t)} = a(\sqrt{(s_{a,t}+\Delta s-b)^2+c} - \sqrt{(s_{a,t}-b)^2+c} + \Delta s)$$

The expression for Δs is given by:

$$\Delta s = \frac{C_{(\Delta s,t)}(2ac\sqrt{(s_{a,t}-b)^2+c} + cC_{(\Delta s,t)})}{2a(ac\sqrt{(s_{a,t}-b)^2+c} + acs + cC_{(\Delta s,t)} - ab\sqrt{c})}$$

a) Sub-linear Curves: The canonical sublinear bonding curve is the logarithmic or root function and is given by:

$$f(s) = log_b(s) \qquad \text{or} \qquad f(s) = s^{\frac{1}{n}} \quad \text{where } n > 1 \tag{9}$$

These curves, which can be accomplished using logarithms and root-based functions, reflect a more conservative approach to growth. Logarithmic curves rise quickly in their nascency, slowly decelerate their rate, and eventually (almost) stabilize.

The cost function for a root-based sub-linear curve is given by:

$$C_{(\Delta s,t)} = \frac{nm}{n+1}[(s+\Delta s)^{\frac{n+1}{n}} - s^{\frac{n+1}{n}}]$$

The expression for Δs is provided by:

$$\Delta s = [s_{a,t}^{\frac{n+1}{n}} + \frac{(n+1)C_{(\Delta s,t)}}{nm}]^{\frac{n}{n+1}} - s_{a,t}$$

*8) **Bonding Curve versus Constant Function Market Makers**:* A bonding curve, encoded in a smart contract, determines token prices based on supply and demand and allows agents to buy and sell tokens without intermediaries. The smart contract also holds the reserve currency to back token values. Constant Function Market Makers (CPMMs) use a constant function to determine the price of a token. They are implemented by smart contracts that create liquidity pools for each token pair, which are funded by liquidity providers who deposit equal amounts of both tokens and receive pool tokens in return [17]. The pool tokens represent their proportional share of the pool and can be redeemed for the underlying tokens at any time. The smart contracts also allow users to swap tokens between the pools by paying a small fee. The fee is distributed to the liquidity providers as a reward for their service. Bonding curves offer various advantages and disadvantages over CPMMS. In the following, we briefly discuss some of the main differences.

a) Continuous Liquidity: Bonding curves provide liquidity for any token at any time, irrespective of the market size or trading volume. Agents do not need to rely on external liquidity providers or wait for matching orders to execute their trades. Thus, in a bonding curve-based economy, there are only two parties in a trade, the token generator (supplier) and token buyer, in addition to the platform. In CPMM, such as UniSwap, LPs must provide both the project token and the reserves. Thus, it requires LPs in addition to the platform, token generator, and token buyer. Continuous liquidity is desirable in a token economy, as it reduces the friction and the risk for the participants and increases the market's efficiency and stability.

[§] https://sound.mirror.xyz

b) Incentive Alignment: Bonding curves align the incentives of the token creators, believers, explorers, and utilizers. Token creators can raise funds by selling tokens through the bonding curve and benefit from the price appreciation as more agents join the economy. Utilizers can buy tokens and use them for various purposes, such as accessing services, voting, or governance, depending on the design of the token economy. Believers can buy tokens early and be rewarded by the potential increase in value as the demand grows. CPMM, on the other hand, may be preferred by believers and explorers only because they can benefit from arbitrage opportunities and liquidity rewards. In CPMM, they can exploit the price differences between the pool and the external market and earn fees from every trade that happens in the pool. They can also adjust the liquidity pool ratio to optimize their returns or risks.

c) Customization: Bonding curves can have different shapes, such as linear, polynomial, or exponential, depending on the desired properties of the token economy. The shape and parameters of the bonding curve affect the price behavior, the supply elasticity, and the reserve ratio of the tokens. Our platform chose the particular bonding curve after thoroughly evaluating different bonding curves with varying parameters. The main theme of this section of the paper is to evaluate different bonding curves and determine the “best” shape and parameters for a buying and selling bonding curve.

d) Price Manipulation: Bonding curves are vulnerable to price manipulation by large or malicious actors who can buy or sell large amounts of tokens and affect the price significantly. This can create artificial price movements, arbitrage opportunities, or market crashes. Bonding curves can implement mechanisms such as price oracles, slippage fees, or price limits to prevent or discourage extreme price deviations.

e) Fragmentation: CPMMs create fragmentation and inefficiency in the token market, as each token pair requires a separate liquidity pool. This means that the liquidity and the trading volume are split among multiple pools, reducing the market's depth and stability. This also means that users may need to perform multiple swaps across different pools to exchange their tokens, increasing the slippage and transaction costs.

f) Impermanent Loss: There is no impermanent loss for the liquidity providers in bonding curve-based models because believers, utilizers, and explorers only provide one type of token (the reserve currency) to the pool. These agents do not have to worry about the price fluctuations of the project token, as they only receive the reserve currency when they withdraw their liquidity. There is always an impermanent loss for the liquidity providers in CPMM because they must provide the pool two types of tokens (usually of equal value). Unlike bonding curves, CPMMs do not have a reserve or collateral to back the value of the tokens. The liquidity providers may suffer from impermanent loss when the price of the tokens in the pool diverges from the price of the tokens in the external market due to arbitrage or other factors. To mitigate this risk, CPMMs can offer liquidity providers higher fees, dynamic fees, or other incentives.

g) High Dependency on Reserve Currency: Bonding curves are simpler and more consistent price functions, as they only depend on the supply and demand of the tokens and not on the relative prices of other tokens. This makes it easier to predict and analyze the price movements and the market dynamics of the tokens. However, it also has the disadvantage of having a higher dependency on the reserve currency, as it affects the tokens' value and liquidity. If the reserve currency loses value or liquidity, the tokens will suffer the same problems.

h) Technical Skills and Knowledge Required: Both CPMM and bonding curves require technical skills and knowledge to interact with the smart contracts that implement them. However, CPMM may require more technical skills and knowledge than bonding curve because it involves more complex calculations, such as finding the optimal trade amount, adjusting the liquidity pool ratio, and managing the slippage and impermanent loss risks. The bonding curve may require less technical skills and knowledge because it involves simpler calculations, such as finding the spot price, buying and selling tokens, and managing the reserve and supply.

i) Computaitonal Efficiency and Gas Fee: CPMMs incur higher gas fees than the bonding curve because it involves more operations, such as updating the liquidity pool, calculating the exchange rate, and transferring the tokens. The bonding curve may incur lower gas fees because it involves fewer operations, such as minting and burning tokens, calculating the spot price, and transferring the reserve currency.

Mathematically, the bonding curves dictate the pricing mechanism in CPMMs. For instance, they use the formula $xy = k$ to determine the price of two tokens in a liquidity pool $A \leftrightarrow B$, where x and y are the quantity or supply of the tokens and k is a constant. The price of the token B in terms of token A is given by $p = \frac{y}{x}$. Similarly, the price of B in terms of A as a function of the supply of B is given by $p = \frac{y^2}{k}$. Therefore, the price curve of CPMMs such as Uniswap is essentially a quadratic bonding curve.

IV. Simulation Setup

The simulation helps us understand the dynamics and evolution of the system under different scenarios and parameters, such as the effect of user heterogeneity or the network structure on the key price and supply. This section presents the simulation setup and the analysis that can be performed based on this simulation.

A. Agent Based Modeling

We use agent-based modeling [9] to simulate the actions and reactions of heterogeneous and adaptive agents (human users referred to herein as creators) and observe the system's emergent patterns and phenomena. This involves using computational models that capture the diversity and complexity of the human users, such as their attributes, rules, interactions, and learning, and the environment and network they operate in, such as the bonding curve, the reserve, and the market. We use a simple model that assumes a large number of creators, a stochastic set of actions, and a bounded rationality of the creators. The model is expressed by a set of algorithms that specify the initial conditions, the update rules, and the output measures of the system.

In our platform, each user has liquidity or one or more ALI Agents, and a common bonding curve governs keys. To simulate our token economy system, we employ a design where human users can be creators, traders, or utilizers. Some human users may want ALI Agents for trading or consuming tokens, while others may want to create their own ALI Agents that can be traded through the bonding curve. These users can employ fundamental analysis, technical analysis, or noise as a strategy and have other personality traits and behavior parameters, such as risk appetite, intelligence level, proactiveness, etc.

We simulate the system using a discrete-time model for a predefined number of steps. At each time step t, we update the state of the system, such as the price, supply, the reserve currency pool, and the creators' balances. We also update the state of the users, such as their beliefs, strategies, and preferences. We then allow the users to interact with the bonding curve smart contract according to their type, risk appetite, and proactiveness. We record the transactions and the outcomes of the trades, such as the values of the keys, the volumes, liquidity, volatility, and efficiency.

We then analyze the simulation results, such as the dynamics and the distribution of the keys' prices, supplies, reserve currency pool, and the users' balances. We also analyze the behavior and performance of the creators, such as their trading patterns, rewards, market shares, and market impact. We compare the simulation results with the theoretical predictions and the empirical observations of the bonding-curve-based multi-token economy system.

B. Creators Types by Strategy

The most important aspect of creators' persona and behavior is their strategy, which determines how they interact with the token economy. In terms of strategy, users exhibit a determined set of attributes and are broadly classified into three classes: fundamental analysts, technical analysts, and noise creators.

1) Fundamental Analysts: Fundamental analysts are creators who base their trading decisions on the intrinsic value or fundamental price of the "keys" of the ALI Agents [18]. They estimate the intrinsic value by analyzing the underlying factors that affect the demand and supply of the keys, such as the quality of the project, the size of the market, the competition, the regulation, and the innovation.

In our settings, fundamental analysts have a high level of intelligence $k \sim N(0.2, 0.05)$, a low level of risk appetite $r \sim N(0.15, 0.01)$, and high liquidity $L \sim N(10000, 1000)$. They tend to buy keys that are undervalued and sell keys that are overvalued. They also tend to hold keys for a long time unless there is a significant change in the fundamentals. These users compute the intrinsic or fair value $\hat{p}_{a,t}$ (that we refer to as the fundamental price) of keys modeled by the expression below.

$$\hat{p}_{a,t} = \sum_{i=1}^{n} w_i (1 \pm k) p_{a,t_i},$$

where $n \sim N(7, 1)$ represents the number of foresight terms, w_i represents the weights given to the fair price $p_{a,t}$ and is given by normalizing the decaying sequence $\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \dots & \frac{1}{n} \end{bmatrix}$.

They compare the fundamental price with the market price and decide whether to buy or sell the asset. If the market price is lower than the fundamental price, the user considers the asset undervalued and buys it. If the market price exceeds the fundamental price, the user considers the asset overvalued and sells it. The user also considers the risk and return of holding the asset and adjusts the trading volume accordingly. The creator has several parameters that determine its behavior, such as risk aversion, confidence level, and time horizon. These parameters affect how the creator updates its expectations, how much it trades, and how sensitive it is to price changes. Most believers will use fundamental analysis as their strategy, while some utilizers and very few explorers will also employ this strategy.

2) Technical Analysts (Chartists): Technical analysts are creators who base their trading decisions on the price patterns and trends of the keys. They use various indicators and tools, such as moving averages, trend lines, support and resistance levels, Fibonacci retracements, and oscillators, to identify their trades' optimal entry and exit points [19]. Technical analysts generally have a medium level of intelligence and a medium level of risk appetite. Technical analysis includes analyzing a key's historical price and volume data to predict its future direction and momentum, using tools, such as trend lines, moving averages, oscillators, and chart patterns, to visualize and interpret the price action [20]. They tend to buy keys that are in an uptrend and sell keys that are in a downtrend. They also tend to hold keys for a short time unless there is a reversal in the trend. For example, if a user observes a bullish trend line on a key chart, it means that the price is making higher highs and higher lows, indicating an upward trend. The user may buy the key and expect the price to continue rising along the trend line.

In our simulation, the chartists either use a simple or a weighted moving average to smooth out the price fluctuations, identify the trend direction, and generate buy or sell signals. The weight vector of moving window size $n \sim N(7, 1)$ contains the normalized decaying weights $\mathbf{w} \in \mathbf{R}^n$ for n previous terms. The historical prices of the key A at time t is given by $\mathbf{p_{a,t}} = \begin{bmatrix} p_{a,t-n} & p_{a,t-n+1} & p_{a,t-n+2} & \dots & p_{a,t-1} \end{bmatrix}^T$. The future expected price is computed by the expression:

$$\hat{p}_{a,t} = \frac{\mathbf{w}^T \mathbf{p_{a,t}}}{n}$$

For example, if a short-term moving average crosses above a long-term moving average, it indicates a bullish trend reversal and a buy signal. Other important parameters determining chartists' behavior are their medium risk appetite $r \sim N(0.2, 0.02)$ and liquidity $L \sim N(5000, 500)$. Such parameters affect how the user selects and applies the indicators, how often it trades, and how it adapts to the market conditions. Most utilizers use technical analysis, while some believers and explorers also employ this strategy.

3) Noise Traders: Noise users base their trading decisions on random factors like emotions, rumors, news, or social influence. They do not have any rational or consistent strategy for their trades. They tend to buy keys that are popular or hyped and sell keys that are unpopular or feared. They also tend to hold keys for a very short time unless there is a strong impulse or pressure to trade.

In our settings, noise users have lower liquidity $L \sim N(2000, 300)$ and higher risk appetite $r \sim N(0.25, 0.03)$. The user does not use any analysis or information to make trading decisions, nor do they have any preferences or expectations. The user simply adds noise to the market by creating random fluctuations in the price and volume. Most explorers will use noisy trading, while very few believers and some explorers will also employ this strategy.

C. Creator's Proactiveness

The creator or user activity or proactiveness is the parameter that determines how frequently the user trades in the market. The user's proactiveness is modeled by a normal distribution $N(\mu, \sigma)$, where the parameter μ represents the average number of trades per time unit, and the parameter σ represents the standard deviation of the number of trades per time unit. The risk appetite of a user determines how much risk they take during trading.

The proactiveness distribution of users with fundamental analyst's strategy is modeled by a normal distribution with a low μ, such as 0.1 or 0.2, and a low σ, such as 0.05 or 0.1, meaning that they trade less frequently and less variably than other types of user.

The proactiveness distribution of the user with a technical analyst's strategy is modeled by a normal distribution with a medium μ, such as 0.5 or 1, and a medium σ, such as 0.2 or 0.4, meaning that they trade more frequently and more variably than fundamental analysts, but less frequently and less variably than noise users.

The proactiveness distribution of users with noisy strategy is modeled by a normal distribution with a high μ, such as 2 or 5, and a high σ, such as 1 or 2, meaning that they trade more frequently and more variably than other types of user.

D. Life Cycle of ALI Agents' Keys

In our token economy, tokens or keys issued by the ALI Agents could serve any of the above purposes. In order to simulate a realistic token economy, we focus on the inherent quality of the keys. For this purpose, we adopt the product life cycle terminology and classify keys into seven classes. Note that while every user (of any purpose category) has their keys, we assume that the most meaningful keys are those owned by creators. These are the main keys that are traded, used, and sought.

Like the traditional product life cycle, we use the "product life cycle" concept to explain the stages an ALI agent's key goes through, from its creation to its destruction and rebirth in the market [6]. It has four phases: **creation**, where the key is first minted and launched **trade**, where the key gains value and liquidity **destruction**, the key is burned or lost; and **rebirth**, where the key is revived or forked. The "key life cycle" helps developers, believers, and participants make strategic decisions about key design, pricing, promotion, distribution, and trading. It also helps them understand customers' changing needs and preferences and the impact of innovation and competition on their keys.

Crypto tokens have different types and functions, such as governance, security, utility, or payment. Depending on their purpose and design, tokens may go through different life cycle stages, from creation to distribution to usage to retirement. In our context, the token life cycle (or supply cycle) represents the token's long-term (three years) weekly adaptation or popularity. From the bonding curve, the supply cycle could be easily converted into fair or inherent token values. We categorized token life cycles into the following seven types and employed the tokens' scaled variants to make the tokens more diverse and realistic.

1) Traditional: A traditional token life cycle follows the classic introduction, growth, maturity, and decline pattern. This type of life cycle is likely to be seen in tokens with stable and predictable demand, a well-established market, and a moderate level of innovation. An example would be Bitcoin.

2) Boom or Classic: A boom or classic product life cycle follows the typical bell-shaped curve that shows the four stages of introduction, growth, maturity, and decline. This type of life cycle is likely to be seen in tokens with a stable and predictable demand, a well-established market, and a moderate level of innovation. An example would be Ethereum.

3) Fad: A fad key life cycle has a short-lived spike in sales that occurs when the key becomes very popular for a brief period but then loses its appeal quickly. The fad life cycle is likely to be seen in keys with a novelty or novelty value, high social influence, and low customer loyalty. An example would be Dogecoin.

4) Extended Fad: An extended fad key life cycle is similar to a fad key life cycle, but the sales do not decline completely. Instead, they stabilize at a lower level than the initial peak. This type of life cycle is likely to be seen in keys with a novelty or novelty value, a high degree of social influence, and a low level of customer loyalty, but also have some residual demand or niche market. An example would be NFT.

5) Revival or Nostalgia: A revival or nostalgia key life cycle occurs when an old-fashioned key that has lost its popularity regains its appeal due to some changes in the market or consumer preferences. This type of life cycle is likely to be seen in keys with a retro or vintage value, a high degree of emotional attachment, and a loyal customer base. An example would be Litecoin.

6) Seasonal: A seasonal key life cycle has a periodic increase and decrease in sales depending on the time of the year. This type of life cycle is likely to be seen in keys with a seasonal demand, a high degree of seasonality, and a low degree of customer loyalty. An example could be a Basic Attention Token (BAT).

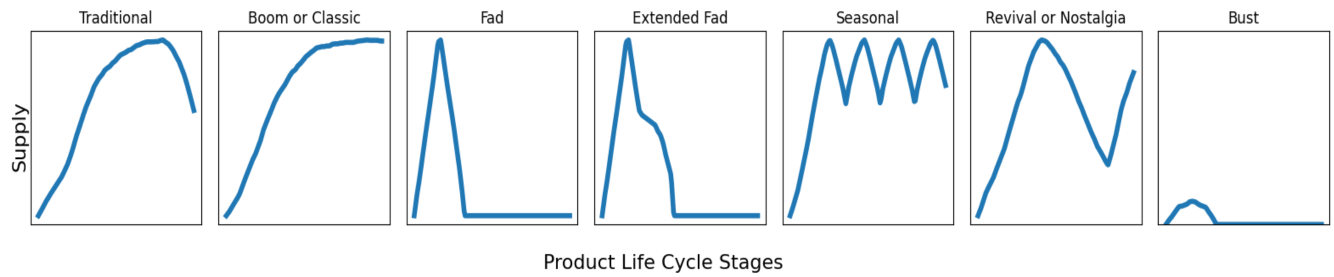

Fig. 7. The product life cycle of distinct "key" types (representing ALI Agents) based on their change in supply or reserves (y-axis) through different stages, i.e., creation, growth, maturity, decline, and revival (x-axis).

7) Bust : A bust key life cycle has a rapid and dramatic decline in sales after a brief period of growth or stability. This type of life cycle is likely to be seen in keys with a high risk, a low quality, and a poor reputation. An example woud be Bitconnect (BCC).

E. Platform Variables and Parameters

Table I lists the most important parameters of the platform along with their brief description.

Table I. Important parameters of the platform and their brief description

Sell Curve	Bonding curve to sell keys
Buy Curve	Bonding curve to buy keys
Transaction Fee	Platform fee for transactions
Number of Users	Proportion of users by purpose
List of Users	For all user types
List of Keys	For every creator
Liquidity	Distributions by users' purpose
Risk Appetite	Distributions by users' strategies
Pro-activeness	Distributions by users' strategies
Foresight Terms	Distribution for fundamentalists
Hindsight Terms	Distribution for chartists
Intelligence Gap	Distribution for fundamentalists
Time Horizon	Duration of the simulation

F. The Creators' Parameters

Each user acts according to a set of rules to carry out rational (maximum value extracting) actions. When a user performs an action, the simulation environment submits transactions to a forked blockchain and updates the smart contract's state. Table I lists the most important parameters of the platform along with their brief description.

G. Distribution of Creators and their Strategies

The number of users varies by the categories of their purpose. We try different distributions. Naturally, the number of creators is far less than the number of other types of users. Among others, there are relatively fewer believers compared to utilizers and explorers. As outlined above, creators have no trading strategies as they do not take part in trading. believers, utilizers, and explorers randomly draw their strategy to be fundamental analysts, technical analysts, or noise traders.

Table II. Important parameters of the users (creators), their ALI Agents' keys, and a brief description.

Purpose Category	$\in$ { believer, Utilizer, explorer }
Strategy	$\in$ { Fundy, Charty, Noisy }
Liquidity	Vary by creator's purpose
Pro-activeness	Vary with creator's strategy
Risk Appetite	Vary with creator's strategy
Intelligence Gap	Error in the AI of fundamentalists
Foresight Terms	Used by fundamentalists
Hindsight Terms	Used by chartists
Term-wise Weights	Used by fundamentalists and chartists for predicting keys' future price
Indicators	Simple or weighted moving average used by chartists for decision making
Arrival Time	Creation time of creator/keys
Key ID	The id of the users' key
Key Quality	Key product life cycle shape
Key Collateral	Reserves against key in supply
Current Supply	Number of Key currently in the market
Current Buy Price	Determined by buy curve at current supply
Current Sell Price	Determined by sell curve at current supply

H. Distribution of Creators' Liquidity, Risks and Proactiveness

As outlined below, users have varying risks and proactiveness depending on their strategy. Generally, fundamental analysts, technical analysts, and noisy users are increasingly risk-takers and proactive in the platform. The initial liquidities, on the other hand, vary by their purpose categories. Broadly speaking, believers have the largest liquidity, followed by utilizers, while explorers have the smallest liquidity. All three parameters come from normal distributions with their means, and variances are platform parameters, as shown in Figure 8.

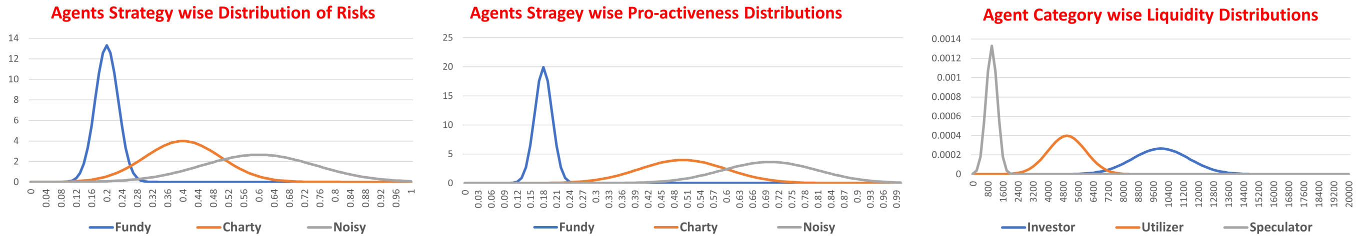

Fig. 8. (left to right) The distribution of users' risk and proactiveness by strategy type and liquidity by purpose category.

I. Creators' Arrival: Platform Adoption

Adoption of new technology is a well-studied discipline [6]. There are five main classes of technology adopters, and their proportions are summarized in Figure 9.

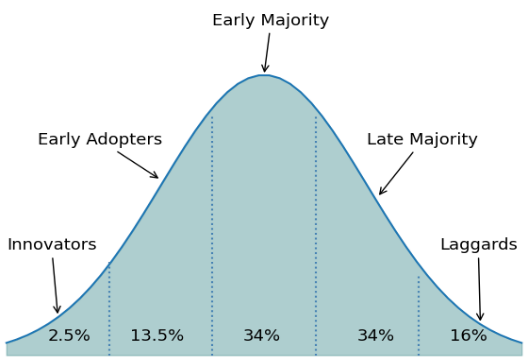

Fig. 9. The arrival of users representing their platform adoption.

1) **Innovators** test any new products out of curiosity. Generally, they are also the beta testers, and their feedback can help test the platform.
2) **Early adopters** are very curious but cautious than innovators, then tend to adopt a product after it has been thoroughly tested.
3) **The early majority** first want to be convinced that the product works.
4) **The late majority** do not take any risk with a product and want to be convinced that the product is valuable for them in the long term.
5) **Laggards** typically are not very tech savvy and need to learn about the product

J. Simulation Algorithm

V. Simulation Results

The simulation experiments were conducted to analyze the bonding curve-based token economy. These experiments varied in user numbers, proportion (by purpose category), and risk appetite, token (key) life cycle and creation types, and bonding curve type and parameters to explore different dynamics within the system. Below, we present the findings categorized into user-wise, key-wise, and platform-wise analyses.

A. Creators-wise Analysis

The simulations involved a range of users from 1000 to 5000, spanning a simulation duration of 3 years (36 months). Each user was categorized into one of four purpose categories: creator, explorer, believer, and utilizer, with distinct strategic and behavioral characteristics. The simulations were performed for different distributions of user behavior parameters. We run simulations with different users' proportions by purpose category because the distribution primarily depends on external factors such as market uncertainty and network connectivity and cannot be determined accurately [21].

The proportion of users by their purpose categories is as follows:

1) High utilizers: utilizers = 40%, explorers = 30%, believers = 20%, creators = 10%
2) High believers: believers = 40%, utilizers = 30%, explorers = 20%, creators = 10%
3) High explorers: explorers = 40%, utilizers = 30%, believers = 20%, creators = 10%
4) High creators: creators = 40%, utilizers = 30%, believer = 20%, explorers = 10%

Risk appetite is a behavioral parameter that users use to decide the amount of key supply to buy or sell. We varied the risk appetite into low and high with the following distribution.

1) Low: A mean of 0.1 and standard deviation of 0.001 for fundamental analysts, a mean of 0.13 and standard deviation of 0.003 for technical analysts, and a mean of 0.15 and standard deviation of 0.005 for noise users.
2) High: A mean of 0.15 and a standard deviation of 0.01 for fundamental analysts, a mean of 0.2 and a standard deviation of 0.02 for technical analysts, and a mean of 0.25 and a standard deviation of 0.003 for noise users.

1) Net Wealth Gain/Loss: Net liquidity $w_{a,t}$ represents the sum of the liquidity available with the user $L_{a,t}$ and the current value of its keys holdings $h_{a,t}$.

$$h_{a,t} = \mathbf{h_{a_i,t}}^T \mathbf{p_{a_i,t}}$$

where $\mathbf{h_{a_i,t}}$ represents the m keys acquired by the user a at time t and $\mathbf{p_{a_i,t}}$ represents their current prices. On the other hand, net liquidity difference $w_{a,\Delta t}$ represents the difference between a user's initial liquidity and the closing liquidity (after the entire simulation duration).

$$w_{a,\Delta t} = w_{a,T} - w_{a,t_0}$$

Algorithm 1 Simulation of our AI Multi-Token Economy System

Initialize the platform variables
Initialize the users with their types, behavior parameters, liquidity, and key balances
Initialize the keys with their initial prices, types, monthly price statistics
while simulation duration is not exhausted **do** ▷ represented by number of weeks
    Select alive users and keys for the current simulation term ▷ assuming users/keys created at the beginning of a simulation term
    Compute expected price of each alive key for each alive user ▷ depending on user type, the expected price will differ
    for each sub-step in current simulation term **do** ▷ represented by minutes in a month
        Select a user based on its proactivity
        Compare the expected price with the current price given by the bonding curve. Decide whether to buy or sell a key and how much to trade based on the user type and the behavior parameters
        Execute the trade by interacting with the bonding curve contract. ▷ The contract calculates the average price and the amount of ALI required or received for the trade and issues or burns the keys accordingly. The contract also updates the price and the supply of the key based on the bonding curve function
        Update the users' liquidity and key holdings based on the trade outcome
        Record the price, the supply, and the liquidity of the transacting key and user
    end for
    Compute and update the weekly price moving averages and high-lows of the alive keys ▷ used by chartists for key's future price prediction
    ANALYZE(Returns of users)
end while
ANALYZE(Keys price dynamics) ▷ Plot the price and the supply of each key over time and observe the dynamics and the patterns of the bonding curve
ANALYZE(Users portfolio) ▷ Plot the liquidity and the key holdings of each user over time and observe the distribution and the changes of the wealth and the portfolio of the users
ANALYZE(Users and keys volatility) ▷ Calculate the return and the volatility of each key and each user over time and compare the performance and the risk of different keys and different users
ANALYZE(Bonding curves) ▷ Analyze the impact of different bonding curve shapes and parameters on the price and the supply of the keys and the behavior and the rewards of the users
ANALYZE(User types) ▷ Analyze the impact of different user types and behavior parameters on the price and the supply of the keys and the behavior and the rewards of the users
ANALYZE(Fundamental prices) ▷ Analyze the impact of different fundamental price functions and random factors on the price and the supply of the keys and the behavior and the rewards of the users
ANALYZE(Correlation and Causations) ▷ Analyze the correlation and the causation between the price, the supply, the liquidity, and the behavior of the keys and the users

We computed the net liquidity difference of each alive user after the completion of a simulation term. A positive liquidity difference represents a gain in liquidity and vice versa. Then, the descriptive statistics of the liquidity difference were computed for each group of traders and different bonding curves, as shown in Figure 10. The results depict that utilizers extracted the most value, while believers extracted the least value for all bonding curves. These results also imply that utilizers are more likely to keep key holdings until the simulation duration is exhausted, while the believers are likely to have increased liquidity.

B. ALI Agent-wise Analysis

Each simulation generated keys equal to the number of creators, with random assignments of life cycle curve shapes. The simulations were performed by creating keys in early and late scenarios where early adoption represents keys created in the first three months while late adoption represents keys created in the first eight months.

1) Price Change Over Time: Keys were associated with creators and experienced fluctuations in price at different rates. The price-changing patterns of keys resembled their product life cycle shapes for less volatile and stable market conditions (described later in Section V-C2). Such market dynamics demonstrate that the true value of the keys was achieved or the true price was discovered. The price time series of keys for different bonding curves are shown in Figure 11. It can be seen that keys with a fad or bust life cycle remained less popular compared to seasonal, revival, or traditional life cycles.

2) Price Volatility : Volatility is a statistical measure of the risk associated with a particular asset and can be used to assess the potential value differential from trading in it. These measures are called stylized facts, including methods such as fat-tail and volatility-clustering [22]. In our simulation runs,

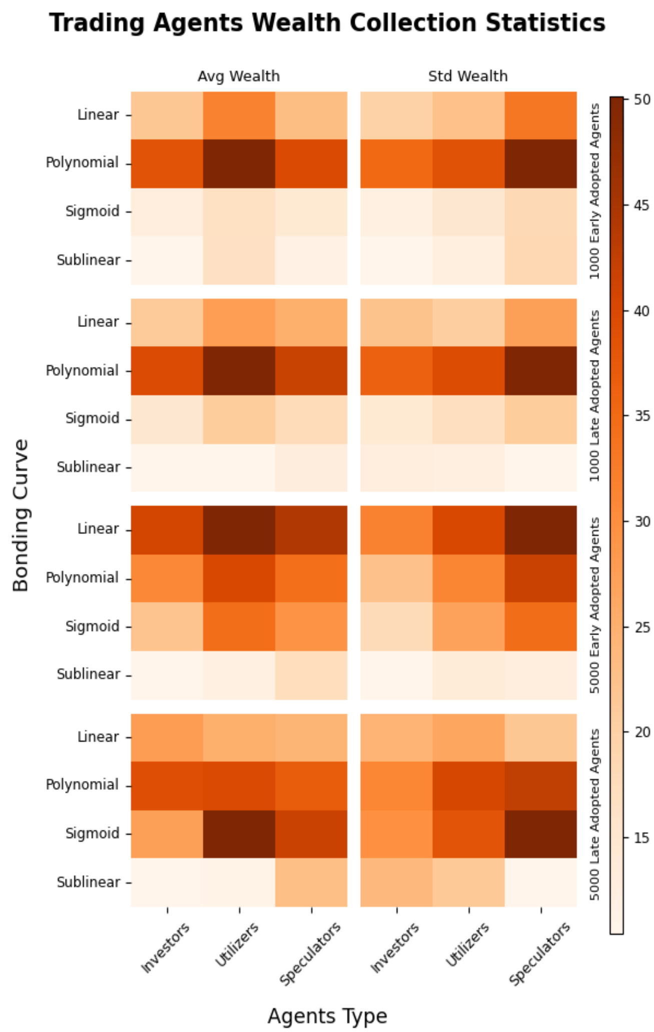

Fig. 10. The difference in the net wealth (closing - initial) of early and late created users belonging to a purpose category for linear, quadratic, sigmoid, and sublinear bonding curves.

we compute the price volatility of the keys as outlined in [4] and visualize the price time series of the least volatile keys by their product life cycle shape. The percentage changes in the logarithmic price or return of a key $r_{a,t}$ at time step t are given by.

$$r_{a,t} = \frac{\log p_t - \log p_{t-1}}{\log p_{t-1}}$$

The average and standard deviation of the returns of a key $\mathbf{r}_a \in \mathbf{R}^T$ for the entire simulation duration T represent the volatility v_a of the key. We computed the volatility statistics of all keys for all simulation experiments and grouped these statistics by the key types.

C. Platform-wise Analysis

Similar to varying the number of users and keys for running experiments, we varied the bonding curve's parameters or coefficients and shapes. Specifically, we ran experiments with linear, polynomial, sublinear, and sigmoid bonding curves and varying parameters. The frequency of transactions per user per week represents the average number of transactions a user can perform weekly. The simulations were run with frequencies of 10 and 30. The parameters for the bonding curves represent varying coefficients. For instance, the second parameter represents the degree of the function n for linear, polynomial, and sublinear curves while the y-inflection point a for the sigmoid bonding curve (essentially the maximum price). Similarly, the third parameter represents the intercept c for linear, polynomial, and sublinear curves while the x-inflection point b for the sigmoid bonding curve (essentially the supply at the maximum price).

The variation influenced the platform's dynamics in bonding curve shapes and parameters.

1) Bonding Curve Impact on the Number of Transactions: Various parameters were observed to impact the total number of transactions for each simulation run. These parameters include the bonding curve type, the users' distribution by their purpose categories, the total number of users, and the users' platform adoption (early or late). The number of transactions in a simulation run is associated with the gas fee on the Ethereum network and is an important indicator for choosing a bonding curve. Figure 12 shows that most transactions were performed with sigmoid bonding curves, followed by polynomial and linear bonding curves. However, the least number of transactions were performed with sublinear bonding curves, demonstrating its lesser adaptability.

2) Bonding Curve Impact on the Key Price: We computed the price volatility indexes (VI) and related statistics for all simulation runs. These statistics were computed using the arithmetic mean, standard deviation, median, minimum, and maximum of the price volatility v_a of all keys for each simulation run. The log of the key prices was computed using a base of 2 to compare the prices to the same scale and cater to very high prices, such as in the case of polynomial bonding curves. Our simulation results, shown in Figure 13, demonstrate that linear and polynomial bonding curves lead to very high price volatility, while sublinear and sigmoid bonding curves create significantly less volatile market conditions.

We also computed the descriptive statistics for each simulation run, i.e., average, standard deviation, and maximum of key prices. These statistics were then averaged for different bonding curves and visualized using heatmap plots. The results show that sub-linear and sigmoid bonding curves exhibited a more stable environment with lower prices. On the other hand, linear and polynomial (quadratic) curves led to higher prices, as shown in Figure 14.

3) Bonding Curve Impact on the Net Wealth Gain/Loss: The results in Figure 10 demonstrate that users extracted the most value using polynomial bonding curves, followed by linear and sigmoid bonding curves. The highest key prices in polynomial bonding curves, as demonstrated in Figure 14, potentially led to the increased net wealth of the traders. Sublinear bonding curves, however, turned out to be the least rewarding bonding curves.

User interactions with different bonding curves showed that creators and believers preferred more stable curves, while

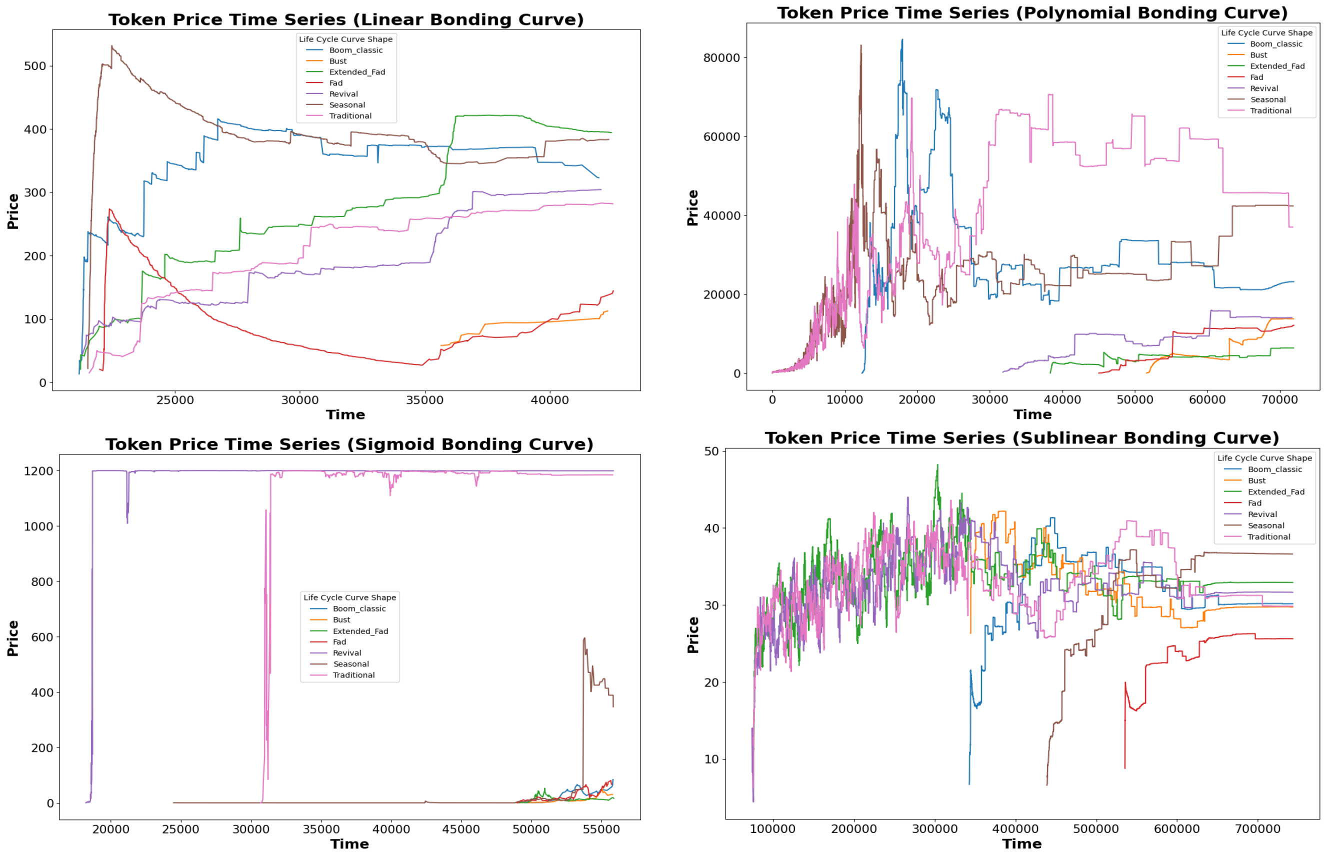

Fig. 11. (left to right) The price time series of keys of different product life cycle types for linear, quadratic, sigmoid, and sublinear bonding curves.

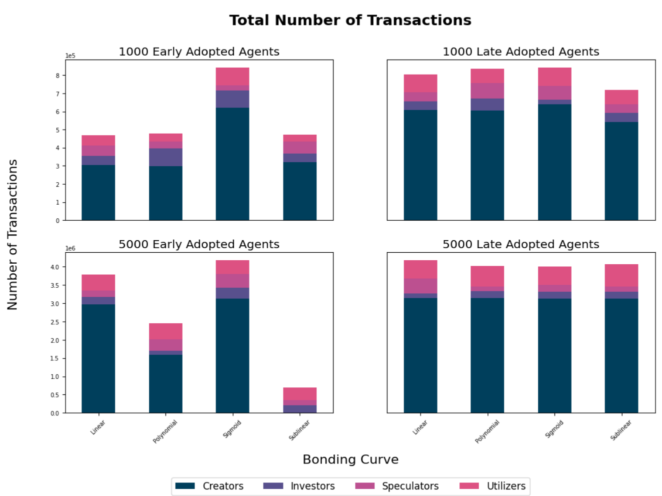

Fig. 12. The average frequency of the total number of transactions performed by users for all simulation runs with varying bonding curves, distributions of users by their purpose categories, the total number of users, and the users' arrival (early or late).

explorers were drawn to curves that provided opportunities for quick gains. Utilizers were relatively unaffected by the curve shapes.

4) Bonding Curve Impact on Collateral Collection: Each simulation had as many keys as the number of creators, while the type of token or key life cycle shape was assigned randomly. As we varied the number of total users and their distribution by purpose category for any simulation, the number of keys was created accordingly. We present the overall performance of different keys for different bonding curves and varying the number of users.

Our results show that higher collateral was collected by keys in the case of the sigmoid bonding curve, followed by the linear and sublinear bonding curves. The lowest collateral, on average, was collected when the key prices were determined by polynomial bonding curves, indicating a very unstable market condition, as shown in Figure 15.

VI. Discussion

This research comprehensively explored the transformative potential and challenges of implementing a decentralized, bonding curve-based token economy in conjunction with agentic AI systems. The discussion below synthesizes key findings reflects on the implications of our results and outlines future research directions.

A. Comparison of Different Bonding Curves

In our research, we compared different bonding curves —linear, polynomial, sigmoid, and sub-linear — in a decentralized token economy using measures such as key price volatility, rewards, collateral collection, and gas fee collection. With their

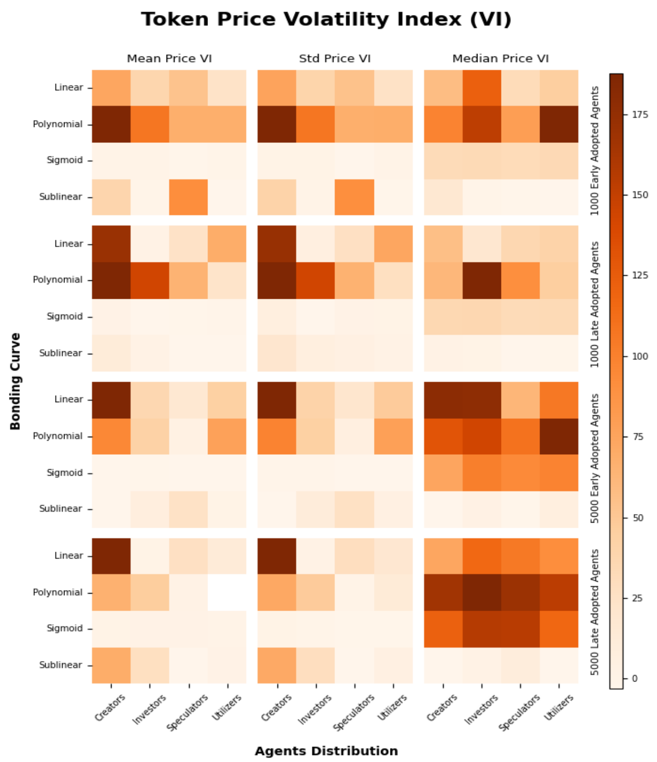

Fig. 13. The key price volatility index (in terms of average, standard deviation, and median) for all simulation runs with different bonding curves and users distribution by their purpose categories.

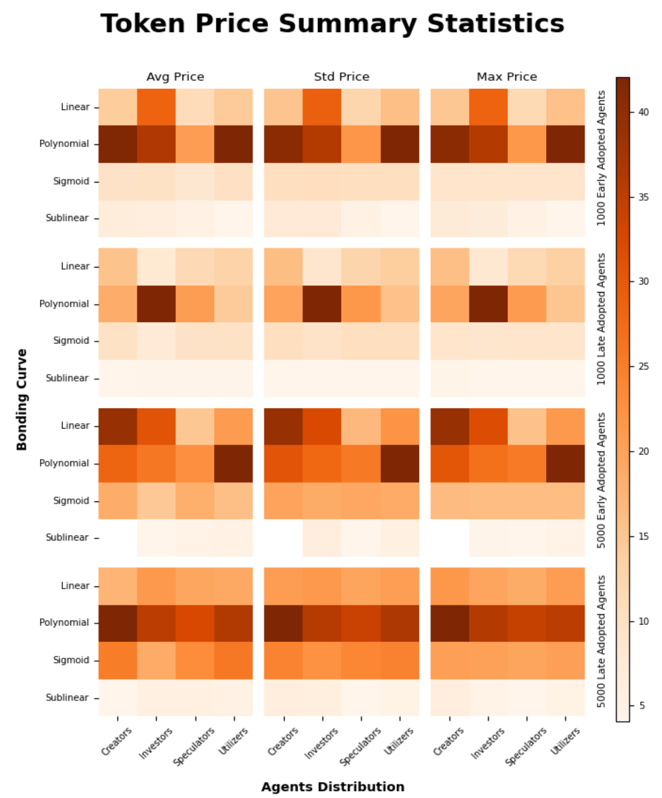

Fig. 14. The summary statistics (average, standard deviation, and max) of the keys price for all simulation runs with different bonding curves and users distribution by their purpose categories.

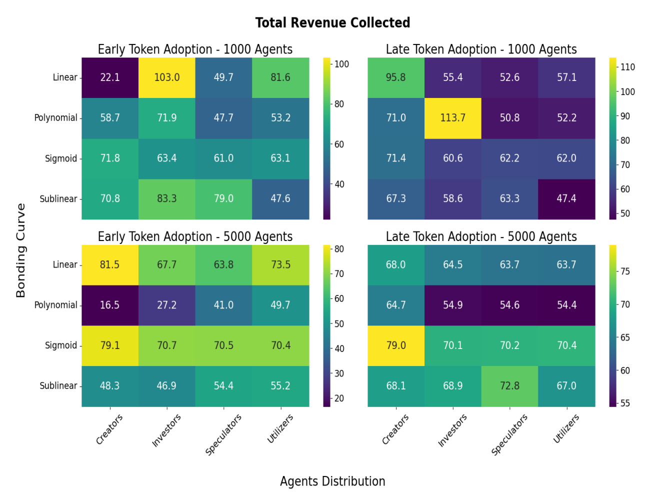

Fig. 15. The average of the collateral collected by keys for all simulations runs with different bonding curves and distributions of users in terms of their purpose categories.

accelerated price increase, polynomial curves cater to environments seeking rapid growth and early trade, appealing to risk-tolerant early adopters. Linear curves are ideal for stable environments, offering predictable growth and attracting long-term, risk-averse believers. Sigmoid curves, balancing initial stability with later rapid growth and lower price volatility, are optimal for projects that require steady early development, targeting a mix of cautious and growth-focused believers. Sublinear curves, characterized by conservative growth, are best for long-term projects aiming for sustainability without high volatility.

The authors in [5] observed that the ability of a market to provide greater liquidity and, therefore, cheaper trading is a major factor in determining a successful market design. In line with these observations, our simulations demonstrated that the sigmoid bonding curve provides cheaper trading and greater liquidity and can be suitable for a sustainable token economy market design. Linear and quadratic bonding curves, on the other hand, lead to rapid and higher prices. The selection of a bonding curve thus hinges on the project's goals, desired market dynamics, and the traders' profile, each presenting unique benefits and limitations.

B. Implications of Liquidity and Bonding Curves

Our study reveals that liquidity significantly empowers ALI Agents, enhancing their autonomy and operational effectiveness. This finding underscores the pivotal role of financial resources in AI-driven ecosystems, as it enables AI agents to access necessary services, information, and other digital assets autonomously. The introduction of bonding curves as a mechanism for key pricing and value creation marks a significant departure from traditional market structures, offering a more dynamic and responsive system that adapts to changes in supply and demand. This feature is particularly beneficial in a decentralized context, where market efficiency and fluidity are paramount.

C. Limitations of Token Bonding Curves

While bonding curves present innovative mechanisms for pricing and key liquidity in decentralized markets, our research and simulation results have identified several limitations. One key challenge is the vulnerability to market manipulation, particularly in scenarios with dominant or malicious actors who can significantly influence key prices by buying or selling large quantities. This creates the potential for artificial price volatility, undermining market stability and fairness. Additionally, TBCs, depending on their shape, may not always align with the long-term sustainability of a token economy. For instance, aggressive growth models like polynomial curves can lead to unsustainable price surges, discouraging early participation and destabilizing the market. Our simulations also revealed that different bonding curve shapes impact user behavior differently, with some curves favoring certain users over others, leading to potential imbalances in the ecosystem. Furthermore, the complexity inherent in understanding and interacting with bonding curves may deter wider participation, particularly among less experienced users. This complexity barrier underscores the need for user education and potentially simpler or more intuitive bonding curve models.

VII. Conclusion

In conclusion, our research has provided novel insights into the dynamics of a decentralized, bonding curve-based token economy integrated with agentic AI. We have demonstrated that by enabling AI agents with liquidity and the ability to issue and trade their own keys, we can significantly enhance their autonomy and effectiveness in a decentralized ecosystem. As embedded liquidity relies on smart contract technology that is trustless and self-determining, powered by a protocol tech stack that is vertically decentralized, the capital efficiency and scalability of deploying tokenized intelligence extricates agentic generative AI systems from the friction of traditional economic markets and the reliance on centralized entities that typically author and command legacy AI systems.

Our simulation results indicate that different types of users interact with the bonding curve-based token economy in diverse and distinct ways, creating a rich and dynamic market environment. The bonding curve mechanism has proven effective in providing a decentralized key pricing model. The curve shapes and parameters directly impact market behavior, influencing the number of transactions, key prices, and collateral collection. Moreover, our study has reproduced such simulated bonding curve-based market conditions where the price-changing patterns of keys closely resemble the fair value of keys, thereby achieving price discovery.

Building upon this foundation, our future work will explore non-embedded ALI agent tokens that emerge as a pivotal "graduation mechanism" and offer ALI agent owners an additional layer of liquidity. This innovative approach can empower the wider community by creating ERC-20 tokens and introducing a new dimension of incentives and rewards, augmenting the agent's market engagement and opening avenues for token-gated access to exclusive content, governance mechanisms, and follower/fandom activities. The flexibility afforded in issuing these types of supplemental liquidity dispensations through the execution of token airdrops and the establishment of liquidity pools may invite widespread involvement in the agent's community and economy. As we look towards the future, the impact and utility of non-embedded ALI agent tokens within the broader ecosystem will be subject to ongoing analysis. This will involve meticulously examining their adoption and utility, leveraging a growing real-world dataset to derive meaningful insights. Such analysis will be crucial in understanding the evolving dynamics of decentralized onchain agentic AI systems infused with liquidity apparatuses, in relation to their traditionally centralized offchain counterparts.

References

[1] A. Lieto, A. Chella, and M. Frixione, "Conceptual spaces for cognitive architectures: A lingua franca for different levels of representation," *Biologically Inspired Cognitive Architectures*, vol. 19, pp. 1–9, 2017.

[2] Y. LeCun, "A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27," *Open Review*, vol. 62, no. 1, 2022.

[3] G. Angeris, H.-T. Kao, R. Chiang, C. Noyes, and T. Chitra, "An analysis of uniswap markets," 2021.

[4] K. Shibano, R. Lin, and G. Mogi, "Volatility reducing effect by introducing a price stabilization agent on cryptocurrencies trading," in *Proceedings of the 2020 The 2nd International Conference on Blockchain Technology*, pp. 85–89, 2020.

[5] C. Chiarella and G. Iori, "A simulation analysis of the microstructure of double auction markets," *Quantitative Finance*, vol. 2, no. 5, pp. 346–353, 2002.

[6] E. Rogers, *Diffusion of Innovations, 5th Edition*. Free Press, 2003.

[7] R. Greenfield, "Solving for secondary rwa liquidity: An introduction to the real-world-asset token bonded curve (rwa tbc) for tokenized bonds," *Available at SSRN 4574438*, 2023.

[8] R. Chen, B. Ramsundar, and R. Robbins, "Fair value and decentralized governance of data," 2019.

[9] V. Aistov, B. Kirpes, and M. Roon, "A blockchain token economy model for financing a decentralized electric vehicle charging platform," in *2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*, pp. 1737–1742, IEEE, 2020.

[10] H. Heaton and S. Green, "Equitable continuous organizations with self-assessed valuations," *arXiv preprint arXiv:2203.10644*, 2022.

[11] H.-T. Kao, T. Chitra, R. Chiang, and J. Morrow, "An analysis of the market risk to participants in the compound protocol," in *Third international symposium on foundations and applications of blockchains*, 2020.

[12] J. S. Park, J. C. O'Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein, "Generative agents: Interactive simulacra of human behavior," 2023.

[13] "Practices for governing agentic ai systems," 2023.

[14] Y. Song, Z. Mi, H. Xie, and H. Chen, "Powerinfer: Fast large language model serving with a consumer-grade gpu," 2023.

[15] U. A. E. Ministry of State for Artificial Intelligence, "100 practical applications and use cases of generative ai," 2023.

[16] J. Schwiderowski, A. B. Pedersen, and R. Beck, "Crypto tokens and token systems," *Information Systems Frontiers*, pp. 1–14, 2023.

[17] V. Mohan, "Automated market makers and decentralized exchanges: A defi primer," *Financial Innovation*, vol. 8, no. 1, p. 20, 2022.

[18] K. Pilbeam, "The profitability of trading in the foreign exchange market: Chartists, fundamentalists, and simpletons," *Oxford Economic Papers*, vol. 47, no. 3, pp. 437–452, 1995.

[19] A. A. A. Aguirre, R. A. R. Medina, and N. D. D. Méndez, "Machine learning applied in the stock market through the moving average convergence divergence (macd) indicator," *Investment Management & Financial Innovations*, vol. 17, no. 4, p. 44, 2020.

[20] N. H. Hung, "Various moving average convergence divergence trading strategies: A comparison," *Investment management and financial innovations*, no. 13, Iss. 2 (contin. 2), pp. 363–369, 2016.

[21] C. Stockermans Munoz, *The Effect of the Chartist to Fundamentalist ratio on Stock Market Price Formation. An ABM approach*. PhD thesis, 2017.

[22] H. Malmsten and T. Teräsvirta, "Stylized facts of financial time series and three popular models of volatility," *European Journal of pure and applied mathematics*, vol. 3, no. 3, pp. 443–477, 2010.

VIII. Appendix

The simulations were run for the following combination of parameters.

Bonding Curve	# Users	Transactions Frequency	Platform Adoption	Users Distribution	Risk	m¶ / c‖	c¶ / a‖	n¶ / b‖
linear	1000	10	early	creators	high	2	0	1
linear	1000	30	early	creators	high	2	0	1
linear	1000	30	early	believers	high	2	0	1
linear	1000	30	early	creators	low	2	0	1
linear	5000	30	early	believers	high	2	0	1
linear	5000	30	early	creators	high	2	0	1
linear	1000	30	early	explorers	high	2	0	1
linear	1000	30	late	explorers	high	2	0	1
linear	1000	30	late	utilizers	high	2	0	1
linear	1000	30	early	utilizers	high	2	0	1
linear	1000	10	late	believers	high	2	0	1
linear	1000	10	early	believers	high	2	0	1
linear	1000	30	early	believers	low	2	0	1
linear	1000	30	late	believers	high	2	0	1
linear	1000	30	early	utilizers	low	2	0	1
linear	1000	30	early	explorers	low	2	0	1
linear	1000	10	early	believers	low	2	0	1
linear	1000	30	late	believers	low	2	0	1
linear	1000	10	late	believers	low	2	0	1
linear	1000	10	early	explorers	high	2	0	1
linear	1000	10	early	creators	low	2	0	1
linear	1000	10	late	creators	high	2	0	1
linear	1000	30	late	explorers	low	2	0	1
linear	1000	10	late	explorers	high	2	0	1
linear	1000	10	early	explorers	low	2	0	1
linear	1000	10	early	utilizers	high	2	0	1
linear	1000	10	late	explorers	low	2	0	1
linear	1000	10	early	utilizers	low	2	0	1
linear	1000	30	late	creators	high	2	0	1
linear	1000	10	late	creators	low	2	0	1
linear	1000	10	late	utilizers	high	2	0	1
linear	1000	10	late	utilizers	low	2	0	1
linear	1000	30	late	creators	low	2	0	1
linear	1000	30	late	utilizers	low	2	0	1
linear	5000	30	early	explorers	high	2	0	1
linear	5000	10	early	believers	low	2	0	1
linear	5000	30	early	explorers	low	2	0	1
linear	5000	10	late	believers	low	2	0	1
linear	5000	30	late	believers	low	2	0	1
linear	5000	10	early	explorers	high	2	0	1
linear	5000	10	late	explorers	high	2	0	1
linear	5000	10	early	explorers	low	2	0	1
linear	5000	30	late	explorers	high	2	0	1
linear	5000	10	late	explorers	low	2	0	1
linear	5000	30	late	explorers	low	2	0	1
linear	5000	30	early	utilizers	high	2	0	1
linear	5000	30	early	utilizers	low	2	0	1
linear	5000	10	early	creators	low	2	0	1
linear	5000	10	early	utilizers	high	2	0	1
linear	5000	10	late	creators	low	2	0	1

¶Linear, sublinear, and polymonial
‖Sigmoid

linear	5000	10	early	utilizers	low	2	0	1
linear	5000	10	late	utilizers	high	2	0	1
linear	5000	10	late	utilizers	low	2	0	1
linear	5000	30	late	creators	low	2	0	1
linear	5000	30	late	utilizers	high	2	0	1
linear	5000	30	late	utilizers	low	2	0	1
linear	5000	30	early	believers	low	2	0	1
linear	5000	10	early	believers	high	2	0	1
linear	5000	10	late	believers	high	2	0	1
linear	5000	30	late	believers	high	2	0	1
linear	5000	10	early	creators	high	2	0	1
linear	5000	10	late	creators	high	2	0	1
linear	5000	30	early	creators	low	2	0	1
linear	5000	30	late	creators	high	2	0	1
polynomial	1000	30	early	creators	high	2	0	2
polynomial	1000	30	early	creators	low	2	0	2
polynomial	1000	30	early	utilizers	high	2	0	2
polynomial	1000	30	early	utilizers	low	2	0	2
polynomial	1000	30	early	believers	high	2	0	2
polynomial	1000	30	early	explorers	low	2	0	2
polynomial	5000	10	early	creators	high	2	0	2
polynomial	1000	30	late	explorers	high	2	0	2
polynomial	5000	30	early	creators	high	2	0	2
polynomial	5000	30	early	creators	low	2	0	2
polynomial	5000	30	early	utilizers	high	2	0	2
polynomial	5000	30	early	believers	high	2	0	2
polynomial	5000	30	early	believers	low	2	0	2
polynomial	5000	10	early	explorers	high	2	0	2
polynomial	1000	10	early	believers	high	2	0	2
polynomial	1000	10	early	believers	low	2	0	2
polynomial	1000	10	early	creators	high	2	0	2
polynomial	1000	10	early	explorers	high	2	0	2
polynomial	1000	10	late	believers	high	2	0	2
polynomial	1000	10	early	explorers	low	2	0	2
polynomial	1000	30	early	believers	low	2	0	2
polynomial	1000	30	early	explorers	high	2	0	2
polynomial	1000	10	late	believers	low	2	0	2
polynomial	1000	30	late	believers	high	2	0	2
polynomial	1000	10	late	creators	high	2	0	2
polynomial	1000	10	late	explorers	high	2	0	2
polynomial	1000	10	early	creators	low	2	0	2
polynomial	1000	10	late	explorers	low	2	0	2
polynomial	1000	30	late	believers	low	2	0	2
polynomial	1000	10	early	utilizers	high	2	0	2
polynomial	1000	10	early	utilizers	low	2	0	2
polynomial	1000	30	late	explorers	low	2	0	2
polynomial	1000	30	late	creators	high	2	0	2
polynomial	1000	10	late	creators	low	2	0	2
polynomial	1000	10	late	utilizers	high	2	0	2
polynomial	1000	10	late	utilizers	low	2	0	2
polynomial	1000	30	late	creators	low	2	0	2
polynomial	1000	30	late	utilizers	high	2	0	2
polynomial	1000	30	late	utilizers	low	2	0	2
polynomial	5000	30	early	utilizers	low	2	0	2
polynomial	5000	30	early	explorers	high	2	0	2

polynomial	5000	10	late	believers	high	2	0	2
polynomial	5000	10	early	believers	high	2	0	2
polynomial	5000	30	early	explorers	low	2	0	2
polynomial	5000	30	late	believers	high	2	0	2
polynomial	5000	10	early	believers	low	2	0	2
polynomial	5000	10	late	believers	low	2	0	2
polynomial	5000	30	late	believers	low	2	0	2
polynomial	5000	10	early	creators	low	2	0	2
polynomial	5000	10	early	explorers	low	2	0	2
polynomial	5000	10	late	explorers	high	2	0	2
polynomial	5000	10	late	creators	high	2	0	2
polynomial	5000	30	late	explorers	high	2	0	2
polynomial	5000	10	early	utilizers	high	2	0	2
polynomial	5000	10	late	explorers	low	2	0	2
polynomial	5000	10	early	utilizers	low	2	0	2
polynomial	5000	30	late	explorers	low	2	0	2
polynomial	5000	30	late	creators	high	2	0	2
polynomial	5000	10	late	creators	low	2	0	2
polynomial	5000	10	late	utilizers	high	2	0	2
polynomial	5000	10	late	utilizers	low	2	0	2
polynomial	5000	30	late	creators	low	2	0	2
polynomial	5000	30	late	utilizers	high	2	0	2
polynomial	5000	30	late	utilizers	low	2	0	2
sigmoid	1000	10	early	creators	high	300	600	400
sigmoid	1000	10	early	creators	low	300	600	400
sigmoid	1000	10	early	believers	high	300	600	400
sigmoid	1000	10	early	utilizers	high	300	600	400
sigmoid	1000	10	early	believers	low	300	600	400
sigmoid	1000	10	early	utilizers	low	300	600	400
sigmoid	1000	10	early	explorers	high	300	600	400
sigmoid	1000	10	early	explorers	low	300	600	400
sigmoid	1000	10	late	believers	high	300	600	400
sigmoid	1000	10	late	creators	high	300	600	400
sigmoid	1000	10	late	believers	low	300	600	400
sigmoid	1000	10	late	explorers	high	300	600	400
sigmoid	1000	10	late	explorers	low	300	600	400
sigmoid	1000	10	late	creators	low	300	600	400
sigmoid	1000	30	early	believers	high	300	600	400
sigmoid	1000	10	late	utilizers	high	300	600	400
sigmoid	1000	30	early	creators	high	300	600	400
sigmoid	1000	10	late	utilizers	low	300	600	400
sigmoid	1000	30	early	believers	low	300	600	400
sigmoid	1000	30	early	explorers	high	300	600	400
sigmoid	1000	30	early	explorers	low	300	600	400
sigmoid	1000	30	late	believers	high	300	600	400
sigmoid	1000	30	early	creators	low	300	600	400
sigmoid	1000	30	late	believers	low	300	600	400
sigmoid	1000	30	early	utilizers	high	300	600	400
sigmoid	1000	30	late	explorers	high	300	600	400
sigmoid	1000	30	early	utilizers	low	300	600	400
sigmoid	1000	30	late	creators	high	300	600	400
sigmoid	1000	30	late	explorers	low	300	600	400
sigmoid	1000	30	late	creators	low	300	600	400
sigmoid	1000	30	late	utilizers	high	300	600	400
sigmoid	1000	30	late	utilizers	low	300	600	400

sigmoid	5000	10	early	believers	high	300	600	400
sigmoid	5000	10	late	believers	high	300	600	400
sigmoid	5000	30	early	believers	high	300	600	400
sigmoid	5000	10	early	believers	low	300	600	400
sigmoid	5000	10	late	believers	low	300	600	400
sigmoid	5000	10	early	creators	high	300	600	400
sigmoid	5000	10	early	explorers	high	300	600	400
sigmoid	5000	30	early	believers	low	300	600	400
sigmoid	5000	10	late	explorers	high	300	600	400
sigmoid	5000	30	late	believers	high	300	600	400
sigmoid	5000	10	late	creators	high	300	600	400
sigmoid	5000	10	early	explorers	low	300	600	400
sigmoid	5000	30	early	explorers	high	300	600	400
sigmoid	5000	10	late	explorers	low	300	600	400
sigmoid	5000	30	early	explorers	low	300	600	400
sigmoid	5000	30	late	believers	low	300	600	400
sigmoid	5000	30	early	creators	high	300	600	400
sigmoid	5000	30	late	explorers	high	300	600	400
sigmoid	5000	10	early	creators	low	300	600	400
sigmoid	5000	30	late	creators	high	300	600	400
sigmoid	5000	30	late	explorers	low	300	600	400
sigmoid	5000	10	early	utilizers	high	300	600	400
sigmoid	5000	10	late	creators	low	300	600	400
sigmoid	5000	10	early	utilizers	low	300	600	400
sigmoid	5000	10	late	utilizers	high	300	600	400
sigmoid	5000	10	late	utilizers	low	300	600	400
sigmoid	5000	30	early	creators	low	300	600	400
sigmoid	5000	30	early	utilizers	high	300	600	400
sigmoid	5000	30	early	utilizers	low	300	600	400
sigmoid	5000	30	late	creators	low	300	600	400
sigmoid	5000	30	late	utilizers	high	300	600	400
sigmoid	5000	30	late	utilizers	low	300	600	400
sublinear	1000	30	early	creators	high	2	0	3
sublinear	1000	30	early	utilizers	high	2	0	3
sublinear	1000	30	late	utilizers	high	2	0	3
sublinear	5000	30	early	utilizers	high	2	0	3
sublinear	5000	30	early	believers	high	2	0	3
sublinear	5000	30	early	explorers	high	2	0	3
sublinear	1000	10	early	creators	high	2	0	3
sublinear	1000	10	early	creators	low	2	0	3
sublinear	1000	10	early	utilizers	high	2	0	3
sublinear	1000	10	early	utilizers	low	2	0	3
sublinear	1000	10	early	believers	high	2	0	3
sublinear	1000	10	early	believers	low	2	0	3
sublinear	1000	10	early	explorers	high	2	0	3
sublinear	1000	10	early	explorers	low	2	0	3
sublinear	1000	10	late	creators	high	2	0	3
sublinear	1000	10	late	creators	low	2	0	3
sublinear	1000	10	late	utilizers	high	2	0	3
sublinear	1000	10	late	believers	low	2	0	3
sublinear	1000	10	late	utilizers	low	2	0	3
sublinear	1000	10	late	explorers	high	2	0	3
sublinear	1000	30	early	utilizers	low	2	0	3
sublinear	1000	10	late	explorers	low	2	0	3
sublinear	1000	30	early	believers	low	2	0	3

sublinear	1000	30	early	explorers	high	2	0	3
sublinear	1000	30	early	explorers	low	2	0	3
sublinear	1000	30	late	believers	high	2	0	3
sublinear	1000	30	late	believers	low	2	0	3
sublinear	1000	30	late	creators	low	2	0	3
sublinear	1000	30	late	explorers	high	2	0	3
sublinear	1000	30	late	utilizers	low	2	0	3
sublinear	1000	30	late	explorers	low	2	0	3
sublinear	5000	10	early	utilizers	high	2	0	3
sublinear	5000	10	early	believers	high	2	0	3
sublinear	5000	10	early	utilizers	low	2	0	3
sublinear	5000	30	early	explorers	low	2	0	3
sublinear	5000	30	early	utilizers	low	2	0	3
sublinear	5000	10	early	believers	low	2	0	3
sublinear	5000	10	late	believers	high	2	0	3
sublinear	5000	10	early	explorers	high	2	0	3
sublinear	5000	30	late	believers	high	2	0	3
sublinear	5000	10	early	explorers	low	2	0	3
sublinear	5000	10	late	creators	low	2	0	3
sublinear	5000	10	late	believers	low	2	0	3
sublinear	5000	30	late	explorers	high	2	0	3
sublinear	5000	10	late	explorers	high	2	0	3
sublinear	5000	10	late	utilizers	low	2	0	3
sublinear	5000	30	late	explorers	low	2	0	3
sublinear	5000	10	late	explorers	low	2	0	3
sublinear	5000	30	late	creators	low	2	0	3
sublinear	5000	30	late	utilizers	high	2	0	3
sublinear	5000	30	late	utilizers	low	2	0	3