One Small Beautiful BSDE to Solve Cryptocurrency Lucas Tree Model

January 22, 2026

Abstract

Solving continuous-time macroeconomic finance models is very challenging. Especially when modeling the dynamics of multiple assets, state variables can easily lead to the curse of dimensionality once their number increases slightly. On the other hand, without appropriate boundary conditions, traditional methods for solving such models face significant challenges in convergence (for example, using the iterative method provided by Brunnermeier and Sannikov (2016)). This paper proposes a two Lucas trees model, consisting of a crypto tree. A key feature of this crypto tree is its dividends are not exogenous but tied to the value of the crypto tree itself. It's an abstraction of Bitcoin in reality: Bitcoin's total supply is tightly capped, with new coins periodically mined. The value of newly mined coins is ultimately determined by the market value of Bitcoin. Another tree produces dividends with follows OU process As a probabilistic method for solving PDEs, Deep BSDE has two key advantages. First, it bypasses the complexity of finite difference methods. Second, it is not highly sensitive to boundary conditions, allowing one to solve the PDE within a region of the state space that has a high probability(mass), which is helpful to solve this crypto tree model.

Keywords: Deep Learning, Macro-Finance, BSDE, Bitcoin

1 Introduction

Dynamic programming forms a foundational pillar of contemporary macro and financial economics. The actions of households, firms, and policymakers (like central bank or government&social planner) are commonly modeled as outcomes of optimizing their corresponding discounted objective functions. Nevertheless, this methodology suffers from the notorious "curse of dimensionality" (Bellman, 1957), meaning that computational demands for both processing time and storage scale exponentially with each additional state variable, and solving a dynamic optimization problem with just four or five state variables can take an extremely long time.

To circumvent the curse of dimensionality, this paper employs deep neural networks to parameterize both the value function and its associated volatility within a continuous-time stochastic framework. Within a continuous-time framework, this paper devises a Deep-BSDE method to evaluate the drift and volatility terms for any specified functional form. At present, this approach demonstrates superior computational efficiency compared to automatic differentiation when simultaneously solving for the volatility terms of each state variable in the presence of multiple states and Brownian motions according to Huang, Ji (2023a) and Huang, Ji (2023b). Also this paper is encouraged by honorable, esteemed

absolutely irreproachable Huang, Ji (2023a), Huang, Ji (2023b) and Geng (2025).

Using the BSDE approach makes the problem even cleaner. The neural network directly outputs the value functions of both households and experts together with their volatilities (exposure to the Brownian shock), the asset price q and its volatility, and the risk-free rate r. The experts' Euler Equation then directly delivers the drift of q, thereby completely eliminating the need for finite-difference approximations of PDE derivatives. Although households may not always hold positive capital and therefore may lack an Euler Equation(equaility) for the capital, their stochastic discount factor can still serve as a powerful consistency check: the risk-free rate implied by the household SDF is supposed to match the risk-free rate directly output by the network.

In the realm of asset pricing, we examine the second example which is the Lucas orchard economy introduced by Martin (2013), which generalizes the canonical single-tree exchange economy of Lucas (1978) into a multi-tree framework, like Cochrane et al. (2008). Follow the framework of Sauzet (2021), We solve a model with three investors and three productive trees, in which each investor can allocate wealth to any of the three trees. A bond market exists in the economy, allowing investors to borrow and lend at the equilibrium interest rate, with the market-clearing condition that the aggregate net bond supply is zero.Similarly, the network directly outputs the value functions of the three investors along with their respective volatilities (market price of risk exposures), the dividend-price ratios F of the three trees together with their volatilities, and the equilibrium risk-free rate. Given the risk-free rate, the drift of each F can be directly recovered from the corresponding Euler equation of the investor who holds a positive amount of that tree. Finally, we verify the consistency condition that the risk-free rate implied by each investor's stochastic discount factor exactly coincides with the risk-free rate directly produced by the network.

2 Lucas Trees Model

The standard one-tree specification, which results in a static price-dividend ratio and i.i.d. returns, fails to capture the rich asset-price dynamics that emerge when the model is expanded to include multiple trees (Cochrane et al., 2008). A fundamental characteristic of this class of models is that asset prices are not exogenous assumptions but are determined endogenously within the system. Consequently, an asset's valuation ratio is primarily a function of its contribution to aggregate consumption. A minor asset may exhibit pronounced positive co-movement in reaction to favorable dividend signals from a dominant asset, thereby acquiring a positive market beta—despite its cash flows remaining fundamentally unrelated to the broader market. This phenomenon reveals a deeper structural interdependence: risk premia emerge not merely from direct cash-flow correlations, but from the market's implicit hierarchical signaling network, where large assets serve as informational anchors that propagate expectations across otherwise disconnected segments of the economy.

The principal value of studying multiple trees lies in exposing asset pricing mechanisms that are completely absent in a single-tree world. A classical case is the pricing of a small asset with independent dividends: the standard intuition—that it should earn no risk premium and be priced by the Gordon growth formula—proves incorrect. This revelation confirms that multi-tree models are not merely an extension but a necessity for uncovering the genuine logic of asset prices (Martin, 2013; Cochrane et al., 2008).

Our Lucas-tree economy features mean reversion in the dividend of the normal tree, following Santos and Veronesi (2006) but I make just a little difference to make sure the state variable is stable to solve, I will explain in next section which is very important.

2.1 Cryptocurrency Lucas Tree

Our economy is built on a key foundation of two trees, one normal tree and another Cryptocurrency Tree I called it Bitcoin tree because it looks like Bitcoins. On the supply side, the two trees continuously produce random consumption goods as their dividends or fruits.

$Y_{i,t}$ is the dividend produced by tree-i at time t which could be immediately consumed with no costs, while $q_{i,t}$ is the equilibrium price of tree-i. As for the first Bitcoin Tree, we carefully model its dividend process as:

$$Y_{1,t} = Bq_{1,t} - Cq_{1,t}^2 \tag{1}$$

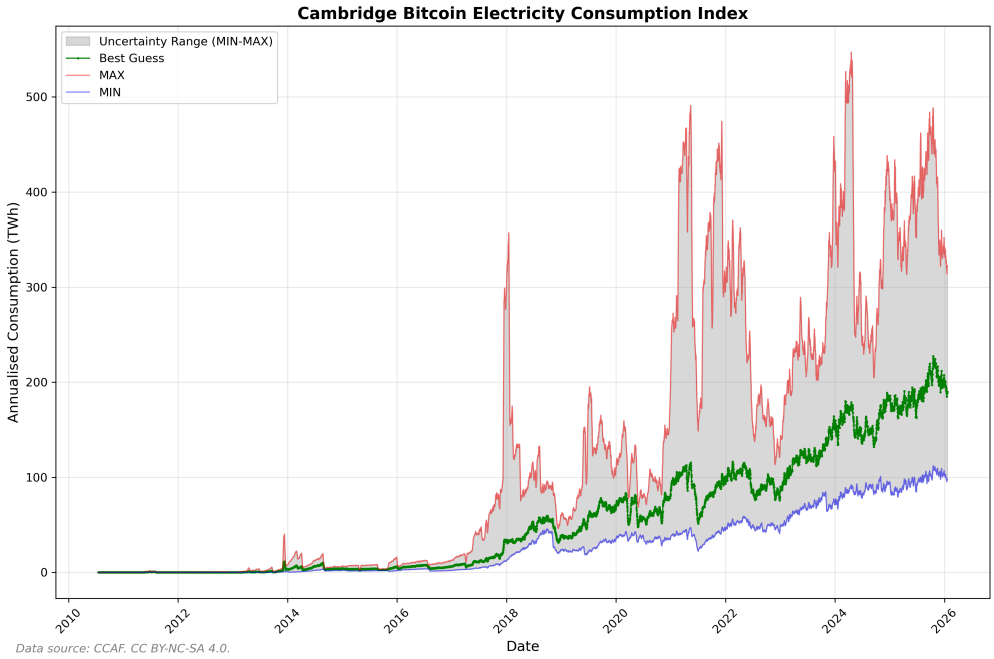
Bitcoin's supply is strictly capped at 21 million. New bitcoins are "mined" at a fixed, predetermined rate—regardless of the number of miners, the number of new coins generated per unit of time is constant. Although the reward for mining new bitcoin undergoes a periodic "halving" approximately every four years or every 210,000 blocks—a mechanism designed to control inflation and preserve Bitcoin's scarcity and value. There have been four halving events to date (in 2012, 2016, 2020, and 2024), with the next expected around March 30, 2028. The underlying process of miners receiving new bitcoin as a reward for adding a block remains a defining feature of the system. In our model, this paper abstracts away from the halving schedule, as it does not fundamentally alter the core economic mechanism we are analyzing.

In the model, the newly mined bitcoins are treated as the "dividends" or "fruits", while the existing fixed supply of Bitcoin represents the "tree" itself. Critically, the "fruit" (new bitcoin) and the "tree" (existing bitcoin) are identical assets. Thus, when the price of Bitcoin (the tree) is high, the value of the newly mined coins (the fruit) is also high. This creates a two-way feedback loop, in which the value of the tree influences the value of the fruits, finally resulting in a highly endogenous system that is very challenging to solve both analytically and numerically.

Under this richly detailed real-world background, parameter B represents the reward rate of Bitcoin which is the number of newly minted Bitcoins awarded for successfully mining a block. Meanwhile, pa-

rameter C captures the mining cost, reflecting Bitcoin's substantial energy expenditure during mining, an energy-intensive process that involves heavy computation to verify transactions. According to an analysis by Cambridge University cited by the BBC, by 2021 Bitcoin's annual electricity consumption already exceeded that of the whole of Argentina, see Criddle (2021). Additionally, holding Bitcoin carries the risk of password loss. Once a password is forgotten, the associated Bitcoin is virtually irrecoverable. According to Binance (2025), around 2.3 million to 3.7 million BTC are lost forever, making nearly 11% to 18% of the 21 million Bitcoin maximum supply.

Futhermore, Bitcoin is often held by transnational criminal groups for illicit activities such as money laundering and scams. It is important to note that despite its decentralized nature, Bitcoin is not immune to seizure by government authorities. A prominent example is the U.S. Department of Justice filed largest ever forfeiture action against approximately $15 Billion in Bitcoin in 2025, see the Press Releases (U.S. Department of Justice). Currently, the total electricity consumption of Bitcoin mining seems continues to increase overall through analyzing Cambridge Bitcoin Electricity Consumption Index[1], but the growth rate has slowed significantly, and the marginal increase is becoming progressively more difficult. This evidence indicates that the cost of acquiring new Bitcoin is rising alongside its soaring price.



After thoroughly analyzing the revenue and costs associated with Bitcoin mining (dividends or fruits of this tree), we require a key concept in economics, which is the Representative Household, to integrate these fragmented economic logics and arrive at the final result: the dividends of the Bitcoin tree are the mining revenue minus the costs, which is exactly the equation (1).

[1]Cambridge Centre for Alternative Finance. (n.d.) Cambridge Bitcoin Electricity Consumption Index. Available at: https://ccaf.io/cbnsi/cbeci (Accessed: 21 01 2026). Licensed under CC BY-NC-SA 4.0.

While heterogeneous agents have increasingly become a focus in economics, see Brunnermeier and Sannikov (2016); Kaplan et al. (2018), a review of the cryptocurrency literature reveals a complex landscape of diverse interest groups. Concepts such as validators who are responsible for mining and may also hold Bitcoin—coexist with retail investors and whales, who benefit from price appreciation without bearing the electricity costs of mining. There are also criminal organizations that exploit Bitcoin's decentralized nature. Modeling all these actors explicitly is practically infeasible.

The representative household framework posits a single type of household that encompasses all economic actors and interest groups—consuming, investing, holding Bitcoin. Thus, all Bitcoin holders are abstracted into one representative household, which enjoys mining revenue $Bq_{1,t}$ while also bearing the cost $Cq_{1,t}^2$. This abstraction diverges from reality, as most Bitcoin holders do not directly incur mining costs. However, for macroeconomic modeling, the representative household approach offers a tractable and empirically valid simplification without compromising the core insights.

Moreover, the dividend function in our model is specified as a quadratic function, which is a simplification that might seem arbitrary. And the function also has a maximum value at $q_{1,t} = \frac{B}{2C}$. In economic terms, this means that there exists a particular Bitcoin price at which Bitcoin tree output is highest, and the social welfare contributed by the Bitcoin tree is also maximized.

If there were a social planner who could set the price to benefit everyone, then they would push the price to that exact point $\frac{B}{2C}$ to maximize overall welfare. For whatever form I may have, I have no expectation that Bitcoin price is determined by maximizing social welfare because according to common sense Bitcoin isn't designed that way, so I just fall in love with the simplest quadratic form. By its very nature, Bitcoin is decentralized and resistant to government control. Its goal isn't to reduce poverty or spread goodwill. Different groups hold Bitcoin for different reasons for example some for speculation, and some for anonymity. So in reality, Bitcoin isn't meant for social good, and its market price naturally drifts away from that "ideal, altruistic" price. So we could accept the quadratic function which isn't perfectly realistic, while it doesn't really matter because the actual price is almost certainly far from the theoretical optimum anyway. And this model doesn't focus on whether that ideal price is "correct" or achievable. It aims at building a micro-foundation for why Bitcoin's price can be driven up by market behavior not by a planner, but by real-world incentives and speculation.

For classical papers like Cochrane et al. (2008) and Martin (2013), these papers typically model the dividend process as a geometric Brownian motion, which aligns better with infinite-horizon dividend discount models for asset pricing. However, this study focuses on medium- to long-term cryptocurrency prices rather than the very long term, because in the long run Bitcoin's supply will inevitably diminish and eventually be fixed at 21 million coins, so at that point the Lucas tree model would lose relevance for Bitcoin. Furthermore, this paper adopts a highly abstract approach aimed at capturing the most essential characteristics of assets. The core feature of the Bitcoin tree is that its dividends are unrelated to economic fundamentals,

which reflects the pricing behavior of crypto assets, see Griffin and Shams (2020). Thus, for a comparative benchmark, we only need a second asset—such as a stock-like asset—whose dividends are linked to economic fundamentals. Accordingly, the dividend process of the second tree in this paper follows an Ornstein–Uhlenbeck (OU) process, which effectively captures business cycle fluctuations and simultaneously helps close the general equilibrium.

Finally this paper would like to develop an extended model that incorporates a greater variety of assets, along with their associated shocks. This would provide a more accurate depiction of the real world, as risks in reality often transmit through different channels and affect various assets differently. Relying on a single asset may therefore be insufficient for a comprehensive analysis. At this point, it is reasonable to use numerical solutions obtained from neural networks to explore the asset price dynamics in a multi-asset, multi-investor setting (Sauzet, 2021; Duarte et al., 2024).

2.2 Basic Setting

There are two trees and one kind of representative households (investors) and one good produced by both trees. The output of first Bitcoin tree is (1). For the second tree, its dividend process is

$$dY_{2,t} = \kappa(\bar{Y} - Y_{2,t})dt + \sigma dZ_t, j = 1, 2 \tag{2}$$

This is a simplest Ornstein–Uhlenbeck (OU) process, and dividends are not restricted to be positive so they can sometimes be negative. However, $\bar{Y}$ must be positive, reflecting a long-run average level around which the economy fluctuates and to which it reverts. The term dZ_t is a standard Brownian motion which captures economic shocks. In reality, jump shocks are also important, see CNN Business. President Donald Trump's threat to impose an additional 100% tariff on imports from China sparked a massive cryptocurrency sell-off within Oct 12, 2025. Digital currencies bitcoin, ether and solana were among the most affected cryptocurrencies, bringing total liquidations to $18.28 billion. But this paper temporarily ignores that issue because I was hurt badly by BSDE with jump terms.

Since there are only 1 good to consume, the price of good can be set as 1. Then we go to the dynamic price process of two trees. In our economy, there exists an investor who discounts future utility, which implies a stochastic discount factor that can discount the future dividends of each tree to the present. In this context, although the dividend process of the two trees are completely different, it is relatively safe to model the price dynamics using geometric Brownian motion. Denote price of risky asset (trees) as $q_{j,t}$ for tree-j, they follow:

$$\frac{dq_{j,t}}{q_{j,t}} = \mu_t^{q,j} dt + \sigma_t^{q,j} dZ_t, j = 1, 2 \tag{3}$$

In equilibrium, all kinds of investors are aggregated into a representative investor. While these individuals may lend to and borrow from one another, the net supply of risk-free bonds is zero at the macro level. The representative investor fully owns the Bitcoin tree and the Normal tree, collecting all their dividends and fruits. As a result, we could directly derive the return on holding each tree. The return consists of two parts: the first is the dividend received during the holding period, and the second is the capital gain from the appreciation of the asset's value. For Bitcoin tree:

$$\begin{aligned} dR_{1,t} &= \underbrace{\frac{Bq_{1,t} - Cq_{1,t}^2}{q_{1,t}}dt}_{\text{dividend yield}} + \underbrace{\frac{dq_{1,t}}{q_{1,t}}}_{\text{capital gains rate}} \\ &= (B - Cq_{1,t} + \mu_t^{q,1})dt + \sigma_t^{q,1}dZ_t \end{aligned} \tag{4}$$

For the second tree:

$$\begin{aligned} dR_{2,t} &= \underbrace{\frac{Y_{2,t}}{q_{2,t}}dt}_{\text{dividend yield}} + \underbrace{\frac{dq_{2,t}}{q_{2,t}}}_{\text{capital gains rate}} \\ &= (\frac{Y_{2,t}}{q_{2,t}} + \mu_t^{q,2})dt + \sigma_t^{q,2}dZ_t \end{aligned} \tag{5}$$

2.3 Representative Agent

A representative agent has constant relative risk aversion (CRRA) utility from consumption C_t with parameter $\gamma > 0$, and discounts the future at the discount rate $\rho > 0$. The utility function is in CRRA form:

$$U_t = \frac{C_t^{1-\gamma}}{1-\gamma} \tag{6}$$

And suppose the agent's life-time value function is

$$V_t = \int_t^\infty e^{-\rho t}\frac{C_t^{1-\gamma}}{1-\gamma}dt \tag{7}$$

According to Brunnermeier and Sannikov (2016), with CRRA utility the agent's value function takes a power form:

$$V_t = \frac{(J_t W_t)^{1-\gamma}}{\rho(1-\gamma)} \tag{8}$$

where $W_t = q_{1,t} + q_{2,t}$ is the wealth of agent, it means the agents household all of the trees with a net bond supply of zero. In most cases, wealth tends to exhibit explosive exponential growth. However in

this paper wealth is essentially stable and has an upper bound—though I expect this to have little impact on the overall analysis. And J_t is a factor or multiplier of wealth like opportunities to use these wealth, this expression (1) shows no additional information about the agent's stochastic investment opportunities is needed. J_t is assumed to follow:

$$\frac{dJ_t}{J_t} = \mu_t^J dt + \sigma_t^J dZ_t \tag{9}$$

Given representative agents hold all the risky assets in the economy, then denote the portfolio weight of Bitcoin tree as $x_{1,t} = \frac{q_{1,t}}{q_{1,t}+q_{2,t}}$, for the second tree it's $x_{2,t} = \frac{q_{2,t}}{q_{1,t}+q_{2,t}}$.To align with the de-scaling of wealth, this paper shifts consumption C_t to the consumption rate $c_t = \frac{C_t}{W_t}$, which refers to the proportion of wealth consumed at time t. Then the wealth of agents evolves as:

$$\frac{dW_t}{W_t} = \underbrace{\{r_t + x_{1,t}(\mu_t^{q,1} - r_t) + x_{2,t}(\mu_t^{q,2} - r_t) - c_t\}}_{\mu_t^W} dt + \underbrace{(x_{1,t}\sigma_t^{q,1} + x_{2,t}\sigma_t^{q,2})}_{\sigma_t^W} dZ_t \tag{10}$$

HJB equation for representative agent:

$$\rho V_t = U_t + \frac{\partial V_t}{\partial W_t} W_t \mu_t^W + \frac{\partial V_t}{\partial J_t} J_t \mu_t^J + \frac{1}{2}\frac{\partial^2 V_t}{\partial W_t^2}(W_t\sigma_t^W)^2 + \frac{1}{2}\frac{\partial^2 V_t}{\partial J_t^2}(J_t\sigma_t^J)^2 + \frac{\partial^2 V_t}{\partial W_t \partial J_t} W_t \sigma_t^W J_t \sigma_t^J \tag{11}$$

$$\begin{aligned}\frac{(J_tW_t)^{1-\gamma}}{1-\gamma} = &\max_{c_t, x_{1,t}, x_{2,t}} \frac{(W_tc_t)^{1-\gamma}}{1-\gamma} + \frac{1}{\rho} J_t^{1-\gamma} W_t^{1-\gamma}\{r_t + x_{1,t}(\mu_t^{q,1} - r_t) + x_{2,t}(\mu_t^{q,2} - r_t) - c_t\} + \\ &\frac{1}{\rho} J_t^{-\gamma} W_t^{1-\gamma} J_t \mu_t^J - \frac{\gamma}{2\rho} J_t^{1-\gamma} W_t^{1-\gamma}(x_{1,t}\sigma_t^{q,1} + x_{2,t}\sigma_t^{q,2})^2 - \frac{\gamma}{2\rho} J_t^{-\gamma-1} W_t^{1-\gamma}(J_t\sigma_t^J)^2 + \\ &\frac{1-\gamma}{\rho} J_t^{-\gamma} W_t^{-\gamma} J_t \sigma_t^J W_t (x_{1,t}\sigma_t^{q,1} + x_{2,t}\sigma_t^{q,2})\end{aligned}$$

divide $(J_tW_t)^{1-\gamma}$ on both sides of equation and then we could get

$$\frac{1}{1-\gamma} = \max_{c_t, x_{1,t}, x_{2,t}} \frac{c_t^{1-\gamma}}{(1-\gamma)J_t^{1-\gamma}} + \frac{1}{\rho}\{r_t + x_{1,t}(\mu_t^{q,1} - r_t) + x_{2,t}(\mu_t^{q,2} - r_t) - c_t\} + \frac{1}{\rho}\mu_t^J \tag{12}$$

$$-\frac{\gamma}{2\rho}(x_{1,t}\sigma_t^{q,1} + x_{2,t}\sigma_t^{q,2})^2 - \frac{\gamma}{2\rho}(\sigma_t^J)^2 + \frac{1-\gamma}{\rho}\sigma_t^J(x_{1,t}\sigma_t^{q,1} + x_{2,t}\sigma_t^{q,2}) \tag{13}$$

The First Order Condition of consumption:

$$c_t = \rho^{\frac{1}{\gamma}} J_t^{\frac{\gamma-1}{\gamma}} \tag{14}$$

Since we don't have the variable called consumption ratio, we express c_t by consumption and wealth

$$c_t = \frac{C_t}{W_t} = \rho^{\frac{1}{\gamma}} J_t^{\frac{\gamma-1}{\gamma}} \tag{15}$$

The First Order Condition for Bitcoin tree portfolio weight $x_{1,t}$:

$$\mu_t^{q,1} - r_t = \gamma(x_{1,t}\sigma_t^{q,1} + x_{2,t}\sigma_t^{q,2})\sigma_t^{q,1} - (1-\gamma)\sigma_t^J \sigma_t^{q,1} \tag{16}$$

The First Order Condition for the normal tree portfolio weight $x_{2,t}$:

$$\mu_t^{q,2} - r_t = \gamma(x_{1,t}\sigma_t^{q,1} + x_{2,t}\sigma_t^{q,2})\sigma_t^{q,2} - (1-\gamma)\sigma_t^J \sigma_t^{q,2} \tag{17}$$

2.4 Markovian Equilibrium

The good market clearing condition:

$$C_t = Bq_{1,t} - Cq_{1,t}^2 + Y_{2,t} \tag{18}$$

Then focus on its volatility term:

$$\sigma_t^C = Bq_{1,t}\sigma_t^{q,1} - 2Cq_{1,t}^2\sigma_t^{q,1} + \sigma \tag{19}$$

The capital market market clearing condition:

$$x_{1,t} = \frac{q_{1,t}}{q_{1,t}+q_{2,t}}, x_{2,t} = \frac{q_{2,t}}{q_{1,t}+q_{2,t}} \tag{20}$$

There is one important variable we don't know which is σ_t^J, go back to:

$$\frac{C_t}{W_t} = \rho^{\frac{1}{\gamma}} J_t^{\frac{\gamma-1}{\gamma}}$$

take the derivative to get volatility term:

$$\begin{aligned} \frac{\sigma_t^C}{W_t} - \frac{C_t}{W_t^2}\sigma_t^W &= \frac{\gamma-1}{\gamma}\rho^{\frac{1}{\gamma}} J_t^{\frac{\gamma-1}{\gamma}}\sigma_t^J \\ \sigma_t^J &= \frac{\gamma}{\gamma-1}\{\frac{\sigma_t^C}{C_t} - \frac{\sigma_t^W}{W_t}\} \end{aligned} \tag{21}$$

where volatility of wealth W_t could be derived by

$$\sigma_t^W = q_{1,t}\sigma_t^{q,1} + q_{2,t}\sigma_t^{q,2} \tag{22}$$

With equation

$$\mu_t^C = Bq_{1,t}\mu_t^{q,1} - 2Cq_{1,t}^2\mu_t^{q,1} - Cq_{1,t}^2\left(\sigma_t^{q,1}\right)^2 + \mu_t^{Y,2} \tag{23}$$

we can easily get an analytic expression of risk free rate r_t from Stochastic Discount Rate:

$$m_t = e^{-\rho t}C_t^{-\gamma} \tag{24}$$

Take first-order derivative and check drift term:

$$\begin{aligned} dm_t &= -\rho e^{-\rho t}C_t^{-\gamma}dt - \gamma e^{-\rho t}C_t^{-\gamma-1}\mu_t^C dt + \frac{\gamma(\gamma+1)}{2}C_t^{-\gamma-2}(\sigma_t^C)^2 dt + \cdots \\ \frac{dm_t}{m_t} &= -\rho dt - \gamma\frac{\mu_t^C}{C_t}dt + \frac{\gamma(\gamma+1)}{2}\frac{(\sigma_t^C)^2}{C_t^2}dt + \cdots \end{aligned} \tag{25}$$

The risk-free rate is straightforward to handle for a representative agent model, as it equals the minus drift of the single agent's stochastic discount factor (SDF). However, in heterogeneous agent models, this problem becomes significantly more complex, often requiring neural networks to approximate the risk-free rate.

$$r_t = \rho + \gamma\frac{\mu_t^C}{C_t} - \frac{\gamma(\gamma+1)}{2}\frac{(\sigma_t^C)^2}{C_t^2} \tag{26}$$

solve the equations (26) and also (16)

$$\begin{aligned} r_t &= \rho + \gamma\frac{(Bq_{1,t} - 2Cq_{1,t}^2)\mu_t^{q,1} - Cq_{1,t}^2(\sigma_t^{q,1})^2 + \mu_t^{Y,2}}{C_t} - \frac{\gamma(\gamma+1)}{2}\frac{(\sigma_t^c)^2}{C_t^2} \\ \frac{Y_{1,t}}{q_{1,t}} + \mu_t^{q,1} - r_t = \text{prem}_1 \quad &\Longrightarrow \quad \mu_t^{q,1} = r_t + \text{prem}_1 - \frac{Y_{1,t}}{q_{1,t}} \\ r_t - \gamma\frac{Bq_{1,t} - 2Cq_{1,t}^2}{C_t}r_t &= \rho + \gamma\frac{(Bq_{1,t} - 2Cq_{1,t}^2)\left(\text{prem}_1 - \frac{Y_{1,t}}{q_{1,t}}\right) - Cq_{1,t}^2(\sigma_t^{q,1})^2 + \mu_t^{Y,2}}{C_t} - \frac{\gamma(\gamma+1)}{2}\frac{(\sigma_t^C)^2}{C_t^2} \end{aligned} \tag{27}$$

so if we know $prem_1$, then we could solve the risk free rate, $prem_1$ comes from the First Order Condition:

$$prem_1 = \gamma(x_{1,t}\sigma_t^{q,1} + x_{2,t}\sigma_t^{q,2})\sigma_t^{q,1} - (1-\gamma)\sigma_t^J\sigma_t^{q,1} \tag{28}$$

Then given r_t from First Order Condition of two trees we could get $\mu_t^{q,1}, \mu_t^{q,2}$

$$\mu_t^{q,1} - r_t = \gamma(x_{1,t}\sigma_t^{q,1} + x_{2,t}\sigma_t^{q,2})\sigma_t^{q,1} - (1-\gamma)\sigma_t^J\sigma_t^{q,1}$$
$$\mu_t^{q,2} - r_t = \gamma(x_{1,t}\sigma_t^{q,1} + x_{2,t}\sigma_t^{q,2})\sigma_t^{q,2} - (1-\gamma)\sigma_t^J\sigma_t^{q,2}$$

3 Different Approaches Starting from Calculating σ_t^q

3.1 Iterative Method

Although our model has only one state variable, Y_2 (for simplicity in notation, we denote the state variable Y_2 by Y in the following analysis), it is particularly difficult to solve using the shooting method, which involves converting the equations above into a system of ODEs. The main challenge lies in determining the prices of the two trees and their first derivatives, such as $q_1'(0)$ and $q_2'(0)$ or at a possible lower bound $q_1'(\underline{Y}), q_2'(\underline{Y})$.

According to Brunnermeier and Sannikov (2016), there is another method called Iterative Method which finds the equilibrium, by solving a system of partial differential equations back in time away from a terminal condition. The key is guessing a terminal condition. The Iterative Method is based on the premise that as we let $T \to +\infty$, behavior at time 0 should converge to the equilibrium of the infinite-horizon economy.

Select N points $Y_1, \cdots, Y_N$ within a reasonable range of the state variable Y. And in this framework, once the price functions $q_{1,T}(Y)$ and $q_{2,T}(Y)$are specified, their derivatives obtained via finite differences can be immediately plugged into Itô's lemma to derive $\sigma_T^{q,1}, \sigma_T^{q,2}$. For example at point i:

$$q_{1,T}(Y_i)\sigma_T^{q,1}(Y_i) = \frac{\partial q_{1,T}(Y_i)}{\partial Y_i}\sigma \tag{29}$$
$$q_{2,T}(Y_i)\sigma_T^{q,2}(Y_i) = \frac{\partial q_{2,T}(Y_i)}{\partial Y_i}\sigma \tag{30}$$

σ is exactly the volatility term of dividend process Y. Since we have only one kind of representative agent who consume all of goods in the economy, so the value function related variable J_t has been pin down by goods market clearing condition, at the same time we are more interested in price dynamics of the two trees. So we just skip J_T and go to σ_T^J, also for point i:

$$\sigma_T^J(Y_i) = \frac{\gamma}{\gamma - 1}\{\frac{\sigma_T^C(Y_i)}{C_T(Y_i)} - \frac{\sigma_T^W(Y_i)}{W_T(Y_i)}\} \tag{31}$$

where $C_T(Y_i), \sigma_T^C(Y_i), W_T(Y_i), \sigma_T^C(Y_i)$ can be expressed the variables we have calculated, including $q_{1,T}(Y_i), \sigma_T^{q,1}(Y_i), q_{2,T}(Y_i), \sigma_T^{q,2}(Y_i)$. Also we could pin down portfolio weight with capital market clearing condition, $x_{1,T} = \frac{q_{1,T}}{q_{1,T}+q_{2,T}}, x_{2,T} = \frac{q_{2,T}}{q_{1,T}+q_{2,T}}$. Then very easily the risk premium of two trees are:

$$prem_1(Y_i) = \gamma\{x_{1,T}(Y_i)\sigma_T^{q,1}(Y_i) + x_{2,T}(Y_i)\sigma_T^{q,2}(Y_i))\}\sigma_T^{q,1}(Y_i) - (1-\gamma)\sigma_T^J(Y_i)\sigma_T^{q,1}(Y_i) \tag{32}$$

$$prem_2(Y_i) = \gamma\{x_{2,T}(Y_i)\sigma_T^{q,1}(Y_i) + x_{2,T}(Y_i)\sigma_T^{q,2}(Y_i))\}\sigma_T^{q,2}(Y_i) - (1-\gamma)\sigma_T^J(Y_i)\sigma_T^{q,2}(Y_i) \tag{33}$$

We can solve the system of equations simultaneously for the risk-free rate $r_T(Y_i), \mu_T^{q,1}(Y_i), \mu_T^{q,2}(Y_i)$. Then we obtain partial differential equations for the functions $q_1(Y,t), q_2(Y,t)$:

$$\mu_t^{q,1} q_{1,t} = \frac{\partial q_{1,t}}{\partial Y}\mu_t^Y + \frac{1}{2}\frac{\partial^2 q_{1,t}}{\partial Y^2}\sigma^2 + \frac{\partial q_{1,t}}{\partial t} \tag{34}$$

$$\mu_t^{q,2} q_{2,t} = \frac{\partial q_{2,t}}{\partial Y}\mu_t^Y + \frac{1}{2}\frac{\partial^2 q_{2,t}}{\partial Y^2}\sigma^2 + \frac{\partial q_{2,t}}{\partial t} \tag{35}$$

Go through Time Step: at any point Y_i, calculate $\frac{\partial q_{j,t}}{\partial t}(Y_i), j = 1, 2$, backward finding:

$$q_{j,t-\Delta}(Y_i) = q_{j,t}(Y_i) - \frac{\partial q_{j,t}}{\partial t}(Y_i), j = 1, 2, i = 1, \cdots, N \tag{36}$$

Until converging. The iterative Time Step in this paper is implemented using an implicit method. As the number of state variables increases, the iteration becomes more complex and convergence becomes more difficult to achieve, especially when it comes to computing second-order partial derivatives involving multiple mixed variables.

3.2 Deep BSDE with Automatic Differentiation(AD) Support

Given the prohibitive computational cost of Iterative Method, this paper employs Han et al. (2018) and Han et al. (2024) BSDE with Deep Learning framework to solve the price dynamics, with the key insight being to treat the prices of the two trees as a system of backward stochastic differential equations.Within the framework of BSDE, variable q and its volatility term σ_t^q are typically coupled variables that must be solved for simultaneously as a set of solutions, rather than finding q alone.

Similarly, we randomly sample N random numbers within the plausible range of Y as its realizations (a uniform distribution is chosen in this paper). A computationally expensive approach is: for any given Y_i, we use a neural network to approximate price variable as

$$q_{1,t} = q^1(Y_t), q_{2,t} = q^2(Y_t) \tag{37}$$

then compute its derivative with respect to Y using the built-in automatic differentiation of the neural network. Then for any sample-i at any time t, we adopt a method that is computationally intensive but requires minimal human effort to calculate σ_q.

$$q_{1,t}(Y_{i,t})\sigma_t^{q,1}(Y_{i,t}) = \frac{dq_{1,t}(Y_{i,t})}{dY_{i,t}}, q_{2,t}(Y_{i,t})\sigma_t^{q,2}(Y_{i,t}) = \frac{dq_{2,t}(Y_{i,t})}{dY_{i,t}} \tag{38}$$

so we solve $\sigma_t^{q,1}(Y_{i,t}), \sigma_t^{q,2}(Y_{i,t})$ just by network Automatic Differentiation at each point $Y_{i,t}$.To derive a numerical algorithm to compute the global solution of equilibrium prices, follow Han et al. (2018) we We apply a temporal discretization. Given a partition of the time interval $[0, T]$, we divide it into H small intervals with length Δ. At each time point t, we simulate the corresponding Brownian motion increment $dZ_t \sim \sqrt{\Delta}N(0,1)$, where $N(0,1)$ is a standard Normal Distribution. First, we simulate one step forward (with time step Δ) for $Y_{i.t}$, where i denotes any sample (or path) and t represents any given time point. Y itself is an exogenous dividend process so its drift and volatility is very clear so we don't need do any calculation, just simulate

$$Y_{i,t+\Delta} = Y_{i,t} + \kappa(\bar{Y} - Y_{i,t})\Delta + \sigma dZ_t \tag{39}$$

Sometimes $Y_{i,t}$ could be negative but it doesn't matter, finally it will revert to $\bar{Y}$, and currently we didn't encounter catastrophic scenarios where a negative Y leads to negative aggregate social output, which in turn results in negative consumption, ultimately causing the model to produce NaN or None values.

Since network has provided the necessary values so we could directly calculate $r_t(Y_{i,t}), \mu_t^{q,1}(Y_{i,t}), \mu_t^{q,2}(Y_{i,t})$ for i-th sample path. In this framework the price of each tree follows BSDE, so we also update

$$q_{1,t+\Delta} = q_{1,t}(Y_{i,t}) + q_{1,t}\mu_t^{q,1}\Delta + q_{i,t}\sigma_t^{q,1}dZ_t \tag{40}$$

$$q_{2,t+\Delta} = q_{2,t}(Y_{i,t}) + q_{2,t}\mu_t^{q,2}\Delta + q_{2,t}\sigma_t^{q,2}dZ_t \tag{41}$$

$$\tag{42}$$

While for updated $Y_{i,t+\Delta}$, the network implies:

$$\hat{q}_{1,t+\Delta} = q^1(Y_{i,t+\Delta}), \hat{q}_{2,t+\Delta} = q^2(Y_{i,t+\Delta}) \tag{43}$$

Finally the loss of this Deep BSDE approach is

$$l = \frac{1}{NH}\{(q_{1,t+\Delta} - \hat{q}_{1,t+\Delta})^2 + (q_{2,t+\Delta} - \hat{q}_{2,t+\Delta})^2\} \tag{44}$$

and we assign a optimizer to minimize this loss. Using a very unconventional understanding, I can think of minimizing this loss as solving a second-order ordinary differential equation with respect to asset(tree) price, for example $q_{1,t}$. For any path-i with the Brownian Motion dZ_t, the expected loss.

$$E(q^1(Y_{i,t+\Delta}) - q^1(Y_{i,t}) - q^1(Y_{i,t})\mu_t^{q,1}\Delta - q^1(Y_{i,t})\sigma_t^{q,1}dZ_t)^2 \tag{45}$$

$$= E(q^1(Y_{i,t}) + \frac{dq^1}{dY_{i,t}}(\kappa(\bar{Y} - Y_{i,t})\Delta + \sigma dZ_t) + \frac{1}{2}\frac{d^2q^1}{dY_{i,t}^2}\sigma^2\Delta - q^1(Y_{i,t})(Y_{i,t}) - q^1(Y_{i,t})\mu_t^{q,1}\Delta - q^1(Y_{i,t})\sigma_t^{q,1}dZ_t)^2$$

$$= E(\{\frac{dq^1}{dY_{i,t}}\kappa(\bar{Y} - Y_{i,t}) + \frac{1}{2}\frac{d^2q^1}{dY_{i,t}^2}\sigma^2 - q_{1,t}\mu_t^{q,1}\}\Delta + \{\frac{dq^1}{dY_{i,t}}\sigma - q^1(Y_{i,t})\sigma_t^{q,1}\}dZ_t)^2 \tag{46}$$

Since we have applied Auto-Differentiation(AD)

$$\frac{dq^1}{dY_{i,t}}\sigma = q^1(Y_{i,t})\sigma_t^{q,1}$$

Then this term vanishes, loss turns to be:

$$E(\{\frac{dq^1}{dY_{i,t}}\kappa(\bar{Y} - Y_{i,t}) + \frac{1}{2}\frac{d^2q^1}{dY_{i,t}^2}\sigma^2 - q_{1,t}\mu_t^{q,1}\}\Delta)^2 \tag{47}$$

which is exactly the ODE of asset price.

3.3 Deep BSDE with σ_t^q Approximated by Neural Network

Using a Neural Network to directly approximate the volatility term or the first-order derivative of asset price q with respect to the state variable aligns more closely with the methodology in work of Han et al. (2018) and Han et al. (2024). Particularly when the number of state variables is large, on one hand, extensive automatic differentiation requires significant computational resources, and on the other hand, when the input dimension of the neural network is high (effectively fitting a high-dimensional function), numerically computing each partial derivative for each state variable may not be stable. Moreover, it may not necessarily aid in the convergence of the numerical solution.

In this paper, we designate a group of neurons specifically to approximate σ_t^q given the input Y_t. This can be implemented as an independent network for σ_t^q or as several non-independent layers.

$$\sigma_t^{q,j}(Y_{i,t}) = \sigma_q^j(Y_{i,t}), j = 1, 2 \tag{48}$$

where σ_q^j is the approximated results by some neurons which could be trained with other neurons approximating q. Then we expand the BSDE implied expected loss for any given asset price into three parts. The first part is:

$$E(\{\frac{dq^1}{dY_{i,t}}\kappa(\bar{Y} - Y_{i,t}) + \frac{1}{2}\frac{d^2q^1}{dY_{i,t}^2}\sigma^2 - q_{1,t}\mu_t^{q,1}\}\Delta)^2 \tag{49}$$

The second part is

$$E(\{\frac{dq^1}{dY_{i,t}}\kappa(\bar{Y} - Y_{i,t}) + \frac{1}{2}\frac{d^2q^1}{dY_{i,t}^2}\sigma^2 - q_{1,t}\mu_t^{q,1}\}\{\frac{dq^1}{dY_{i,t}}\sigma - q^1(Y_{i,t})\sigma_t^{q,1}\}dZ_t\Delta) = 0 \tag{50}$$

During the actual training of the neural network, given an input $Y_{i,t}$, as long as we simulate a sufficient number of realized Brownian motion increments dZ_t, according to the law of large numbers, we can also observe that the gradient of this term is always zero, meaning it does not affect the training results.

The third part is:

$$E(\{\frac{dq^1}{dY_{i,t}}\sigma - q^1(Y_{i,t})\sigma_t^{q,1}\}^2 dZ_t^2) = E(\{\frac{dq^1}{dY_{i,t}}\sigma - q^1(Y_{i,t})\sigma_t^{q,1}\}^2)\Delta \tag{51}$$

When training the neural network, we inevitably minimize this part, so the initially approximated σ_t^q will converge to the σ_t^q computed via automatic differentiation, while significant computational resources are saved.

3.4 Approximate both σ_t^q and r_t by Neural Network: Hard to Understand

This paper considers an extremely special case, where there is only one representative agent, so the risk-free rate can be derived analytically through his Stochastic Discount Factor (SDF). However, once there are multiple representative agents, the situation becomes very complex.

One strategy is to go all out: let the network approximate σ_t^q, then continue approximating the risk-free rate r_t, and finally compute μ_t^q from the Euler equation. Take the Bitcoin tree as an example, if r_t is approximated by network then

$$\mu_t^{q,1}(Y_{i,t}) = r_t(Y_{i,t}) + \gamma(x_{1,t}\sigma_t^{q,1}(Y_{i,t}) + x_{2,t}\sigma_t^{q,2}(Y_{i,t}))\sigma_t^{q,1}(Y_{i,t}) - (1-\gamma)\sigma_t^J(Y_{i,t})\sigma_t^{q,1}(Y_{i,t})$$

The painful part is that when minimizing the loss, the network has too many objectives. Assuming an extreme case where we assign an independent network to each variable that needs approximation, thereby minimizing their mutual influence—the process of minimizing the BSDE loss for the Bitcoin tree would then take the following form.

$$\begin{aligned}\min_{q^1(Y_{i,t}),r(Y_{i,t}),\sigma^{q,1}(Y_{i,t})} E[q^1(Y_{i,t+\Delta}) - q^1(Y_{i,t}) - q^1(Y_{i,t})\{r_t(Y_{i,t}) + \gamma(x_{1,t}\sigma_t^{q,1}(Y_{i,t}) + x_{2,t}\sigma_t^{q,2}(Y_{i,t}))\sigma_t^{q,1}(Y_{i,t}) \\ - (1-\gamma)\sigma_t^J(Y_{i,t})\sigma_t^{q,1}(Y_{i,t})\}\Delta - q^1(Y_{i,t})\sigma_t^{q,1}(Y_{i,t})dZ_t]^2\end{aligned}$$

When computing gradients to update the networks, the gradients for $q^1(Y_{i,t})$, $r(Y_{i,t})$, and $\sigma^{q,1}(Y_{i,t})$ get tangled up and torment each other, making convergence difficult to ensure.

Moreover, we have inadvertently introduced the higher-order term such as $(\sigma^{q,1})^2$, which makes the intrinsically nonlinear model even more nonlinear. This escalation in nonlinearity also means the optimization process now involves solving for two correlated stochastic volatility terms simultaneously, effectively turning the problem into a system of intertwined nonlinear equations. This not only complicates gradient flow by introducing additional saddle points and local minima but also amplifies the risk of instability during training, as small perturbations in one volatility term can propagate and magnify through the coupled structure. Therefore, what might appear as a mere additional term actually deepens the structural nonlinearity, turning an already challenging high-dimensional BSDE into a numerically more delicate and potentially less tractable problem.

4 Implementations

This paper employs a set of simple parameters to examine whether the four models discussed in the previous section can yield consistent results.If the results for a single output level are reasonable, this paper will extend the analysis to a model featuring a Bitcoin tree alongside multiple normal trees. This implies that the number of normal tree dividend processes will increase, thereby raising the dimensionality of the model.

Table 1: Key Model Parameters and Their Values

Parameter	**Description**	**Value**
κ	Output reverting parameter	0.20
$\bar{Y}$	Steady-state output(dividend) level	2.00
γ	Relative risk aversion coefficient	1.20
ρ	Time preference rate	0.03
σ	Technology(Fundamental) shock standard deviation	0.03
B	Bitcoin Tree dividends	1.00
C	Bitcoin Tree costs	0.10

4.1 Description of Methods

Iterative Method. Although the mean level of dividend for the normal tree is $\bar{Y} = 2$, we aim to expand the solution domain to [0, 10] to identify necessary boundary conditions at both ends. After extensive analysis and deliberation over a period of about ten days, this paper finds that at the point 0, the asset prices q_1 and q_2 likely do not possess reasonable first- or second-order boundary conditions.

Since this paper employs an implicit method to accelerate convergence during the solution process, although convergence is not always guaranteed, we have imposed two boundary conditions at Y=10 that may not be entirely rigorous from a scientific standpoint. Specifically, for the Bitcoin tree price q_1:

$$\frac{\partial q_1(Y,t)}{\partial Y}|_{Y=10} = 0 \tag{52}$$

The rationale behind this approach is that we assume the mean of Y is $\bar{Y} = 2$, and Y reverts to this mean at a rate determined by $\kappa = 0.2$. Even accounting for stochastic shocks, $Y = 10$ is already far removed from the mean of 2. Consequently, investors would reasonably conclude that Y will revert rapidly toward the mean. Consider an extreme scenario: when Y is exceedingly large, even over a very short time interval Δ, Y would decrease significantly. This implies that changes in Y would have a minimal impact on the price of Bitcoin tree q_1, as the Bitcoin market would struggle to price an asset whose underlying fundamental (Y) is moving at an extremely rapid pace.

For another normal tree price q_2, we suppose:

$$\frac{\partial^2 q_2(Y,t)}{\partial Y^2}|_{Y=10} = 0 \tag{53}$$

Denote the price-dividend ratio as $F_t = \frac{q_t}{Y_t}$. For the O-U process, when Y is extremely large and reverts to the mean with a nearly deterministic drift of $\kappa(Y - \bar{Y})$, the behavior resembles an exponential decay process. The representative agent discounts future consumption using $e^{-\rho t}$. When Y is sufficiently large, the output from the Bitcoin tree becomes negligible in comparison. So the representative agent effectively discounts only the Normal tree dividends, and the discounted value is unsurprisingly proportional to the enormous Y. It means that the price-dividend ratio F is a constant. Consequently, the second derivative of q_2 with respect to Y should approach 0 when Y is extremely large.

Deep BSDE with Automatic Differentiation(AD) Support. As a probabilistic solution method, BSDEs do not require perfect coverage of the entire state space. Focusing on the interval [0.05, 5] seems sufficient, given that Y follows an O-U process with a mean of 2. The probability of Y falling within this range is substantial. Moreover, allocating excessive samples to simulate rare events at the extremes would be inefficient and potentially misleading, especially considering that the neural network's attention capacity is inherently limited. So we draw each initial value (of sample path) Y_i uniformly from this range [0.05, 5], where i is the sample index. And then we simulate this path including $Y_i, q_{1,i}, q_{2,i}$ for 30 periods and minimize BSDE loss. The length of each period is 0.005, and the randomness comes from Brownian Motion increment within each period. The input variables for the neural network are first passed through a single hidden layer of size 32. The output from this layer then serves as a new input for 2 separate sub-networks: one outputs the Bitcoin tree price $q_{1,t}$, and the other outputs the normal tree price $q_{2,t}$. Automatic Differentiation(AD) is done by TensorFlow which needs just a few lines of codes.

Deep BSDE with σ_t^q Approximated by Neural Network. The key improvement of this method lies in using the neural network to approximate the volatility terms $\sigma^{q,1}, \sigma^{q,2}$ of the asset prices, while keeping all other settings unchanged. The input variables for the neural network are first passed through a single hidden layer of size 32. The output from this layer then serves as a new input for 4 separate sub-networks. Those output $q_{1,t}, q_{2,t}$ and also $\sigma^{q,1}, \sigma^{q,2}$.

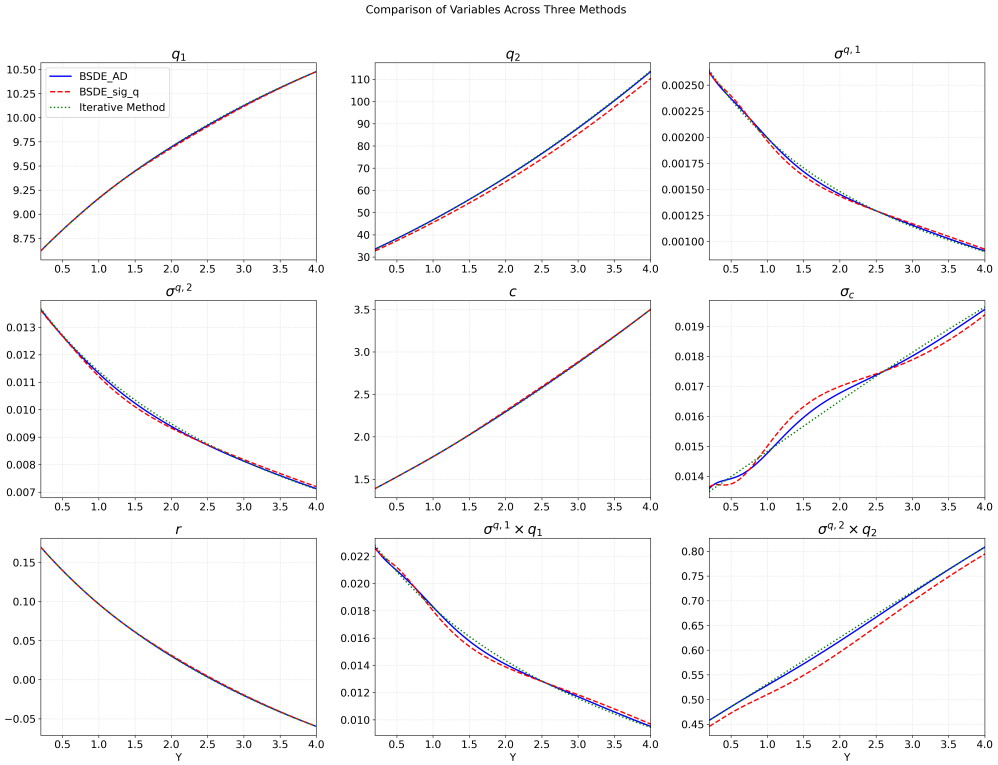
Approximate both σ_t^q and r_t by Neural Network. This algorithm is by far the most entertaining and mind-blowingly unconventional one. The input variables for the neural network are first passed through a single hidden layer of size 32. The output from this layer then serves as a new input for 5 separate sub-networks. Those output $q_{1,t}, q_{2,t}$ and also $\sigma^{q,1}, \sigma^{q,2}$ and most importantly r_t.

One major advantage of the BSDE approach is that it allows us to focus solely on the region of interest. For instance, I am primarily concerned with asset prices within the range of Y = [0.2, 4], and I have little interest in scenarios where Y is extremely large. Furthermore, the BSDE method eliminates the need to manually specify boundary conditions. This is particularly beneficial because if I were to add another tree

to the model, the task of deriving appropriate boundary conditions and handling the corresponding PDE implicitly would become prohibitively complex, if not entirely intractable.

4.2 Results

This paper compares the results obtained using three methods: Iterative Method, BSDE_AD, and BSDE_sig_q—with the state variable ranging over [0.2, 4]. The objects of study include the bitcoin tree price q_1, another normal tree price q_2, the equilibrium risk-free interest rate r, the volatility terms $\sigma^{q,1}$ and $\sigma^{q,2}$, the total consumption (output) of the economy, the volatility of consumption, and the total volatilities of the two assets, $\sigma^{q,1}q_1$ and $\sigma^{q,2}q_2$.



It can be observed that the results from the three methods are largely consistent, with the only exception being a slight deviation in the outcomes from the network approximation of σ_t^q. This deviation might be attributed to the influence of randomly generated Brownian motion increments during the simulation process.

As can be seen from the figure, even when Y is very small that indicates poor economic fundamentals and severely low output, Bitcoin tree still exhibits a price bubble. The definition of a bubble in this paper is quite straightforward: since the maximum level of consumption goods that the Bitcoin tree can provide is 5, a price significantly above 5 indicates the presence of a bubble, whereas a price significantly below 5 suggests either insufficient liquidity or underinvestment. Given that Y follows an Ornstein–Uhlenbeck process, it possesses a strong mean-reverting force when Y is very low. Consequently, investors already anticipate that the economy will soon emerge from the downturn, and the recovery is expected to be extremely rapid.

Moreover, during the upward phase of the economic cycle, the prices of both assets rise at an even faster pace than the economic recovery (dividends of normal tree will gradually increase) itself. This anticipated price appreciation motivates investors to purchase the Bitcoin tree, thereby creating a bubble.

There is also an insightful summary of this phenomenon in the literature on digital asset pricing. Griffin and Shams (2020) and Scheinkman and Xiong (2003) says financial bubbles often coincide with the belief that a rapid gain can be obtained from simply selling an asset to another speculator. Investors purchase assets not because of their belief in the underlying cash flows, but because they can sell the asset to another individual with a higher valuation. Also Liu et al. (2020) found most and perhaps all of the coins represent bubbles and fraud and the findings of these eminent scholars align with the results presented in this paper.

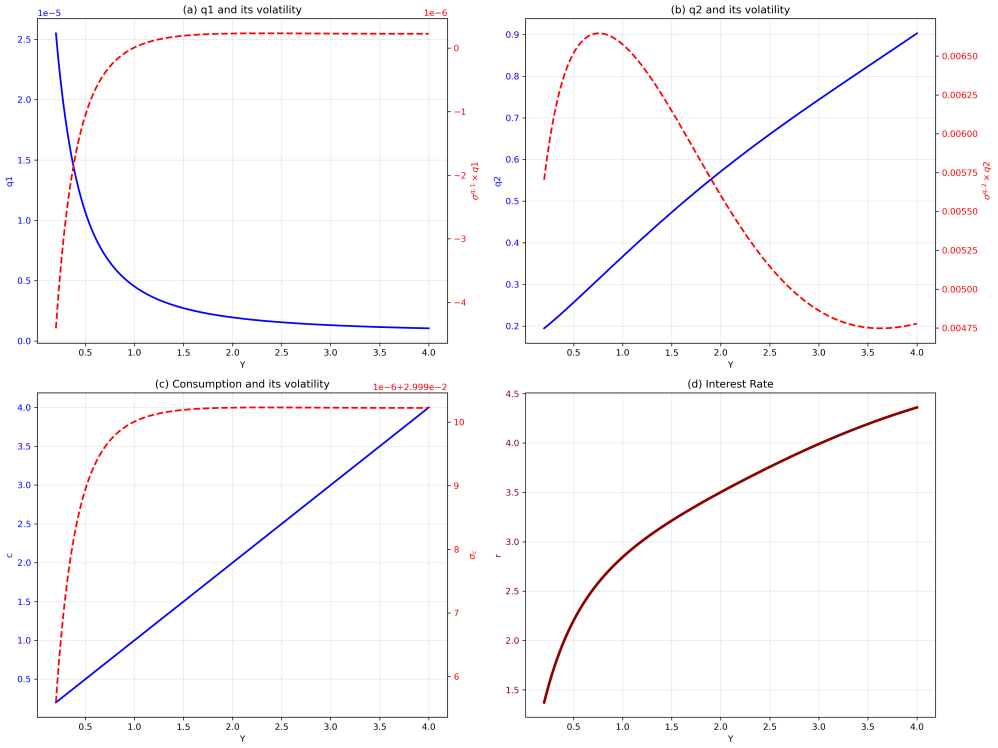
When the fundamental economic state variable Y reverts to its mean level of around 2, the fruits or dividends generated by the Bitcoin tree approaches zero, which is a situation more perilous than a mere bubble. A bubbly asset at the very least still produces dividends; the bubble arises because overzealous investors overvalue those dividend payouts. In contrast, here the dividends have vanished entirely. The growth in the asset's value is now driven solely by investment demand, with the contribution from dividends having completely vanished.

When the economic state variable Y significantly exceeds its long-term mean, it signals an overheated economy that itself exhibits bubble-like characteristics. Under these conditions, the price of the normal tree continues to rise with the economic heat. While the Bitcoin tree's price increases but shows a marked slowdown in its growth rate, coupled with dividends turning completely negative. Cochrane et al. (2008) provides an insightful explanation for this phenomenon, which is fundamentally determined in the structure of the Lucas tree model itself. Modeling assets as trees implies that their supply elasticity is zero, as trees can only produce fruit but cannot be arbitrarily split or replicated, which is more like Bitcoin who generates block rewards but has a strictly capped total supply of 21 million coins. In an economy with two trees, overheating inevitably drives up the price of the normal tree. However, investors perceive this overheating as unsustainable, especially when Y is extremely high, anticipating a swift economic cool-down and a subsequent sharp decline in asset prices. Consequently, they are compelled to realize profits from the normal tree which means they will sell normal tree. With assets exhibiting zero supply elasticity, the capital withdrawn from the normal tree is forced to flow into the Bitcoin tree, even though holding the latter not only yields no positive dividends but actually incurs negative dividends (i.e., holding costs). This forced capital inflow sustains the Bitcoin tree's price growth despite its negative fundamentals.

A central failure of this paper lies in the fact that, while it successfully replicates the phenomenon of the Bitcoin tree price decoupling from economic fundamental, which is an important finding consistent with the existing digital asset pricing literature, the volatility of Bitcoin in my model is unrealistically low. This starkly contradicts the high volatility observed in real-world Crypto currency markets. The likely culprit is the overly simplistic and limited shock structure employed here. In reality, Crypto currency market sentiment

is often driven by a rich variety of jump shocks, a complexity that our model fails to capture. I suspect that even extensive parameter tuning would be futile in addressing this core limitation, so I have opted not to pursue it further. Secondly, the paper provides relatively little insight into social welfare. The results depict an exceedingly smooth path for total social consumption, which exhibits weak correlation with asset prices and instead closely tracks economic fundamentals. While this consumption pattern is stable, it offers minimal analytical value from a welfare perspective, as it largely ignores the welfare implications of asset price dynamics.

Approximate both σ_t^q and r_t by Neural Network. The results are funny, but I will present them to provide some emotional value for the audience. Additionally, this demonstrates that substituting the risk-free rate r guessed by the neural network into the Euler equation effectively transfers the power to determine asset prices from the representative agent to the network. Typically, the network tends to depress asset prices, and once the scale of asset prices is reduced, the loss naturally decreases. For instance, in the model of this paper, the normal BSDE loss is around 10^{-8} , but after granting the network the power to determine the risk-free rate, the loss can be reduced to as low as 10^{-10}.



5 Experiment: Introducing a Second Normal Tree

This mathematical experiment aims to investigate whether the price dynamics of the Bitcoin tree change when we introduce a second normal tree that is also linked to macroeconomic fundamentals. The setup will involve solving the model with one Bitcoin tree and two distinct normal trees, both of which are correlated with the underlying economic state variables Y_2, Y_3. The objective is to observe if the interplay between three trees including one speculative tree (Bitcoin) and two fundamental-driven tree alters the equilibrium pricing behavior.

This paper has no choice but to place the trust in the solutions obtained via the BSDE method, as attempting to solve the model using the Iterative method with two state variables would be prohibitively complex. Given that the reliability of the BSDE approach has already been verified in the single-state-variable case, it is anticipated that the addition of an extra normal tree will not substantially alter the core model and also its solution.

This time the dividends of Bitcoin tree is still:

$$Y_{1,t} = Bq_{1,t} - Cq_{1,t}^2$$

As for the second and the third tree their dividends follow two O-U processes while the big difference is that we have two kinds of Brownian Motion capturing two kinds of independent fundamental shocks.

$$dY_{2,t} = Y_{2,t} + \kappa_2(\bar{Y}_2 - Y_{2,t})dt + \sigma dZ_t^1 \tag{54}$$

$$dY_{3,t} = Y_{3,t} + \kappa_3(\bar{Y}_3 - Y_{3,t})dt + \sigma dZ_t^2 \tag{55}$$

Denote price of risky asset (trees) as $q_{j,t}$ for tree-j, they follow:

$$\frac{dq_{j,t}}{q_{j,t}} = \mu_t^{q,j}dt + \sigma_t^{q,j,1}dZ_t^1 + +\sigma_t^{q,j,2}dZ_t^2, j = 1, 2, 3 \tag{56}$$

The return of holding Bitcoin tree:

$$dR_{1,t} = (B - Cq_{1,t} + \mu_t^{q,1})dt + \sigma_t^{q,1,1}dZ_t^1 + \sigma_t^{q,1,2}dZ_t^2 \tag{57}$$

For the second and the third tree:

$$dR_{j,t} = (\frac{Y_{j,t}}{q_{j,t}} + \mu_t^{q,j})dt + \sigma_t^{q,j,1}dZ_t^1 + \sigma_t^{q,j,2}dZ_t^2, j = 2, 3 \tag{58}$$

Introduce some vectorized notation methods. For example:

$$\boldsymbol{\sigma_t^{q,j}} = \begin{pmatrix} \sigma_t^{q,j,1} \\ \sigma_t^{q,j,2} \end{pmatrix}, \boldsymbol{\sigma_t^{q,i}} \cdot \boldsymbol{\sigma_t^{q,j}} = \sigma_t^{q,i,1}\sigma_t^{q,j,1} + \sigma_t^{q,i,2}\sigma_t^{q,j,2} \tag{59}$$

The investment opportunity of representative agent J_t turns to be

$$\frac{dJ_t}{J_t} = \mu_t^J dt + \sigma_t^{J,1} dZ_t^1 + \sigma_t^{J,2} dZ_t^2 \tag{60}$$

Then the evolution of wealth now is affected by the price dynamic of all three tress, its drift and volatility terms turn to be

$$\mu_t^W = r_t + x_{1,t}(\mu_t^{q,1} - r_t) + x_{2,t}(\mu_t^{q,2} - r_t) + x_{3,t}(\mu_t^{q,3} - r_t) - c_t \tag{61}$$

$$\sigma_t^{W,1} = x_{1,t}q_{1,t}\sigma_t^{q,1,1} + x_{2,t}q_{2,t}\sigma_t^{q,2,1} + x_{3,t}q_{3,t}\sigma_t^{q,3,1} \tag{62}$$

$$\sigma_t^{W,2} = x_{1,t}q_{1,t}\sigma_t^{q,1,2} + x_{2,t}q_{2,t}\sigma_t^{q,2,2} + x_{3,t}q_{3,t}\sigma_t^{q,3,2} \tag{63}$$

The new good market clearing condition:

$$C_t = Bq_{1,t} - Cq_{1,t}^2 + Y_{2,t} + Y_{3,t} \tag{64}$$

with the drift term:

$$\mu_t^C = Bq_{1,t}\mu_t^{q,1} - 2Cq_{1,t}^2\mu_t^{q,1} - Cq_{1,t}^2\{(\sigma_t^{q,1,1})^2 + (\sigma_t^{q,1,2})^2\} + \mu_t^{Y,2} + \mu_t^{Y,3} \tag{65}$$

$$= Bq_{1,t}\mu_t^{q,1} - 2Cq_{1,t}^2\mu_t^{q,1} - Cq_{1,t}^2\boldsymbol{\sigma_t^{q,1}} \cdot \boldsymbol{\sigma_t^{q,1}} + \mu_t^{Y,2} + \mu_t^{Y,3} \tag{66}$$

Its volatility terms:

$$\sigma_t^{C,1} = Bq_{1,t}\sigma_t^{q,1} - 2Cq_{1,t}^2\sigma_t^{q,1} + \sigma \tag{67}$$

$$\sigma_t^{C,2} = Bq_{1,t}\sigma_t^{q,2} - 2Cq_{1,t}^2\sigma_t^{q,2} + \sigma \tag{68}$$

Then we modify the HJB equation, the change of first term about wealth:

$$(\sigma_t^W)^2 \Rightarrow \sum_i \sum_j x_{i,t}x_{j,t}\boldsymbol{\sigma_t^{q,i}} \cdot \boldsymbol{\sigma_t^{q,j}} \tag{69}$$

For the cross term between J and W.

$$\sigma_t^J \sigma_t^W \Rightarrow x_{1,t}\boldsymbol{\sigma_t^J} \cdot \boldsymbol{\sigma_t^{q,1}} + x_{2,t}\boldsymbol{\sigma_t^J} \cdot \boldsymbol{\sigma_t^{q,2}} + x_{3,t}\boldsymbol{\sigma_t^J} \cdot \boldsymbol{\sigma_t^{q,3}} \tag{70}$$

The First Order Condition for Bitcoin tree portfolio weight $x_{1,t}$:

$$\mu_t^{q,1} - r_t = \gamma(x_{1,t}\boldsymbol{\sigma_t^{q,1}} + x_{2,t}\boldsymbol{\sigma_t^{q,2}} + x_{3,t}\boldsymbol{\sigma_t^{q,3}}) \cdot \boldsymbol{\sigma_t^{q,1}} - (1-\gamma)\boldsymbol{\sigma_t^J} \cdot \boldsymbol{\sigma_t^{q,1}} \tag{71}$$

The First Order Condition for the first normal tree portfolio weight $x_{2,t}$:

$$\mu_t^{q,2} - r_t = \gamma(x_{1,t}\boldsymbol{\sigma_t^{q,1}} + x_{2,t}\boldsymbol{\sigma_t^{q,2}} + x_{3,t}\boldsymbol{\sigma_t^{q,3}}) \cdot \boldsymbol{\sigma_t^{q,2}} - (1-\gamma)\boldsymbol{\sigma_t^J} \cdot \boldsymbol{\sigma_t^{q,2}} \tag{72}$$

The First Order Condition for the second normal tree portfolio weight $x_{2,t}$:

$$\mu_t^{q,3} - r_t = \gamma(x_{1,t}\boldsymbol{\sigma_t^{q,1}} + x_{2,t}\boldsymbol{\sigma_t^{q,2}} + x_{3,t}\boldsymbol{\sigma_t^{q,3}}) \cdot \boldsymbol{\sigma_t^{q,3}} - (1-\gamma)\boldsymbol{\sigma_t^J} \cdot \boldsymbol{\sigma_t^{q,3}} \tag{73}$$

Fortunately the First Order condition of consumption doesn't change so we can still use its to calculate $\boldsymbol{\sigma_t^J}$

$$\sigma_t^{J,1} = \frac{\gamma}{\gamma-1}\{\frac{\sigma_t^{C,1}}{C_t} - \frac{\sigma_t^{W,1}}{W_t}\}, \sigma_t^{J,2} = \frac{\gamma}{\gamma-1}\{\frac{\sigma_t^{C,2}}{C_t} - \frac{\sigma_t^{W,2}}{W_t}\} \tag{74}$$

What is most crucial and gives me the greatest confidence is that the calculation of the risk-free rate has not become overly complex.

The risk-free rate is straightforward to handle for a representative agent model, as it equals the minus drift of the single agent's stochastic discount factor (SDF). However, in heterogeneous agent models, this problem becomes significantly more complex, often requiring neural networks to approximate the risk-free rate.

$$r_t = \rho + \gamma\frac{\mu_t^C}{C_t} - \frac{\gamma(\gamma+1)}{2}\frac{(\sigma_t^{C,1})^2 + (\sigma_t^{C,2})^2}{C_t^2} \tag{75}$$

solve the equations

$$r_t = \rho + \gamma\frac{(Bq_{1,t} - 2Cq_{1,t}^2)\mu_t^{q,1} - Cq_{1,t}^2\boldsymbol{\sigma_t^{q,1}} \cdot \boldsymbol{\sigma_t^{q,1}} + \mu_t^{Y,2} + \mu_t^{Y,3}}{C_t} - \frac{\gamma(\gamma+1)}{2}\frac{\boldsymbol{\sigma_t^C} \cdot \boldsymbol{\sigma_t^C}}{C_t^2}$$

$$\frac{Y_{1,t}}{q_{1,t}} + \mu_t^{q,1} - r_t = \text{prem}_1 \qquad \Longrightarrow \qquad \mu_t^{q,1} = r_t + \text{prem}_1 - \frac{Y_{1,t}}{q_{1,t}}$$

$$r_t - \gamma \frac{Bq_{1,t} - 2Cq_{1,t}^2}{C_t} r_t = \rho + \gamma \frac{(Bq_{1,t} - 2Cq_{1,t}^2)\left(\text{prem}_1 - \frac{Y_{1,t}}{q_{1,t}}\right) - Cq_{1,t}^2 \boldsymbol{\sigma_t^{q,1}} \cdot \boldsymbol{\sigma_t^{q,1}} + \mu_t^{Y,2} + \mu_t^{Y,3}}{C_t} - \frac{\gamma(\gamma+1)}{2} \frac{\boldsymbol{\sigma_t^C} \cdot \boldsymbol{\sigma_t^C}}{C_t^2} \tag{76}$$

so if we know $prem_1$, then we could solve the risk free rate, $prem_1$ comes from the First Order Condition of Bitcoin tree portfolio weight $x_{1,t}$.

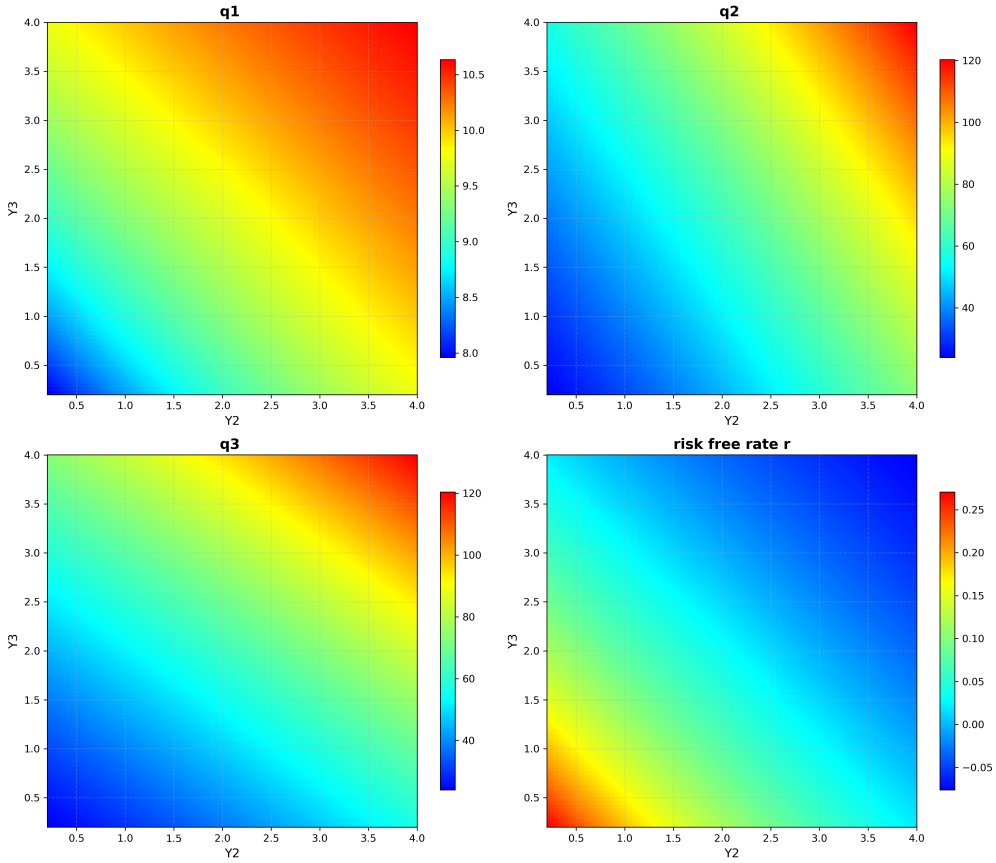
Call upon TensorFlow's Automatic Differentiation support and then we have

$$q_{j,t}\sigma_t^{q,j,1} = \frac{\partial q_{j,t}}{Y_{1,t}}\sigma$$
$$q_{j,t}\sigma_t^{q,j,2} = \frac{\partial q_{j,t}}{Y_{2,t}}\sigma$$

5.1 Results

To facilitate solving the solutions and comparison, this paper assumes the two normal trees have identical parameters, both consistent with the values of κ and $\bar{Y}$ provided in Table 1.

This way we can once again use the previous Deep BSDE methods to solve the problem and also test all three BSDE methods. The most reliable results are still solved by Method BSDE_AD



Currently it appears that the impact of adding one tree and an economic fundamental shock on assets is not particularly significant. When both fruits of the Normal tree are in a low state, the interest rate seems to have increased from the previous 15% (with one Bitcoin tree paired with one Normal tree, and the Normal tree dividend close to 0) to 25%. This suggests that investors expect the economy to recover quickly and consumption to grow rapidly, accompanied by strong borrowing demand, which has pushed interest rates to an even more incredible level. Of course, high interest rates seem to have suppressed asset prices, with Bitcoin decreasing from the previous 8.6 (with one Bitcoin tree paired with one Normal tree, and the Normal tree dividend close to 0) to 8.

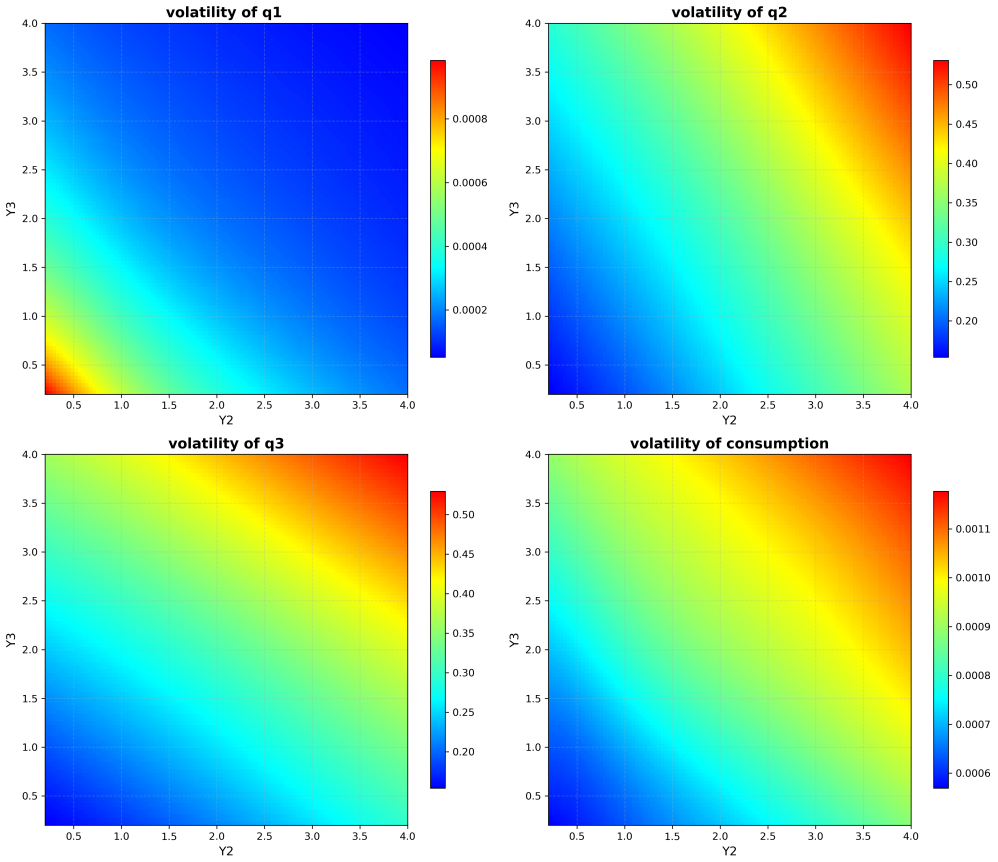
When the economy is in a prosperous phase, and both normal tree dividends are significantly above the average (e.g., both normal tree dividends are close to 4, with total dividends around 7), the price of Bitcoin remains almost the same as in a single normal tree economy where the normal tree dividend is close to 4, both hovering around 10.5. Additionally, the prices of the two normal trees are around 120, showing little change compared to the single normal tree economy. This indicates that when economic fundamentals are completely overheated (with both trees exceeding expected dividends), investors do not become overly irrational or create more aggressive bubbles.

When one normal tree has a dividend at the average of 2, and the other normal tree has a dividend close to 4, the price of Bitcoin remains around 10.3, while the price of the normal tree with a dividend of 2 is around 90, and the price of the normal tree with a dividend of 4 is about 100. As the sources of fundamental shocks increase, asset bubbles in normal trees appear to be suppressed, but the Bitcoin tree remains unaffected. This reflects that when normal tree prices deviate from the average, investors still have the need to take profits, leading to funds flowing into the Bitcoin tree.

Then observe the volatility of the three assets and consumption across different states. The volatility level of tree-j is calculated by

$$vol(q_{j,t}) = q_{j,t}^2((\sigma_t^{q,j,1})^2 + (\sigma_t^{q,j,2})^2) \tag{77}$$

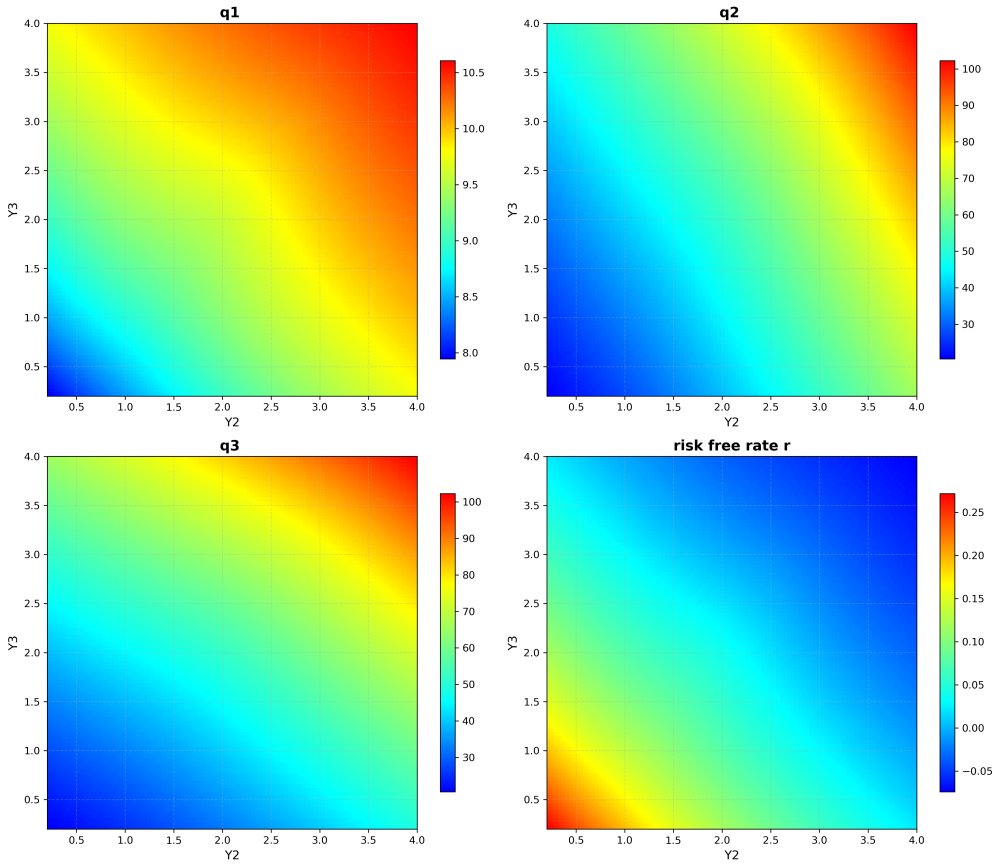
This analysis incorporates the impact of fundamental shocks from different channels on the assets. By measuring the total volatility of the assets, we can observe that when the economy is booming and significantly above average levels, the volatility of the Bitcoin tree is actually suppressed. The volatility of normal assets increases with rising dividends, but under the two normal trees scenario, the maximum volatility is 0.5, which is lower than the 0.8 level seen in the single normal tree case. This could be attributed to investors diversifying their holdings across multiple assets, allowing them to take profits when dividends from a particular tree are exceptionally favorable. Thus, the two normal trees setup not only reduces the formation of bubbles but also dampens the overall volatility amplitude.

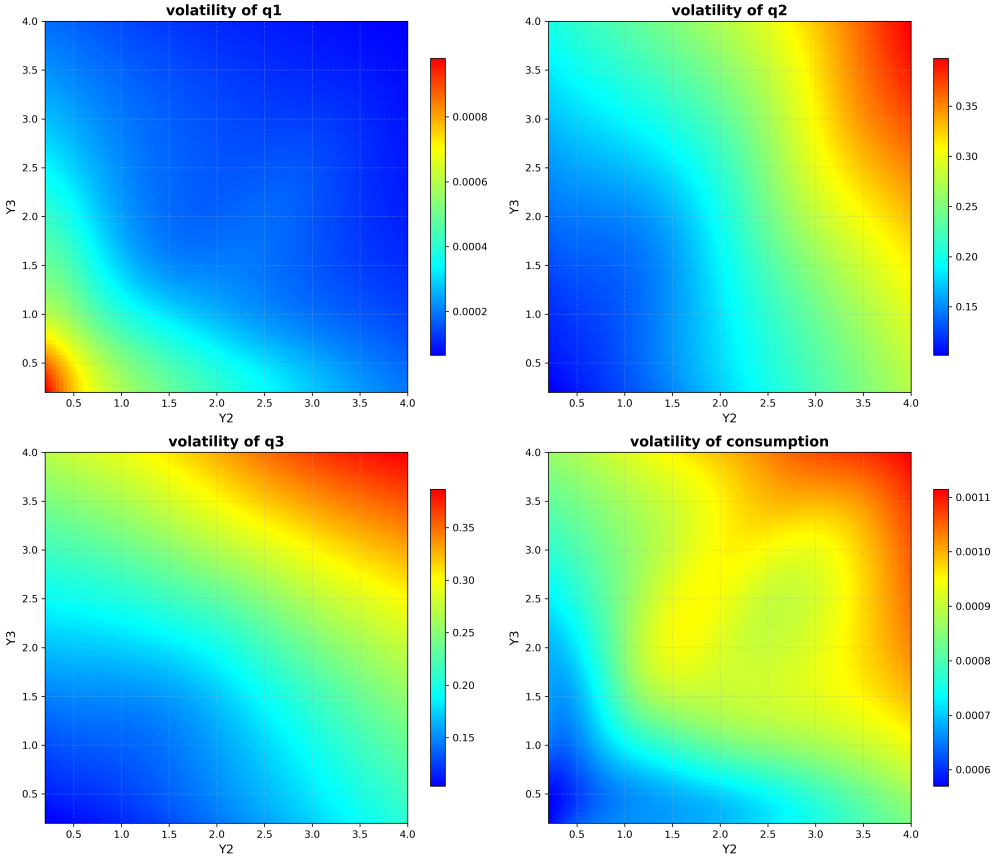


where consumption volatility is calculated by:

$$vol(C_t) = (\sigma_t^{C,1})^2 + (\sigma_t^{C,2})^2) \tag{78}$$

Shifting focus to the results obtained from the BSDE_sig_q method, this approach does not rely on AD support but instead uses the neural network itself to approximate sig_t^q. From the graphical results, the tree prices derived through this method appear to be generally lower, and the errors are more pronounced compared to the previous single normal tree case. I have already abandoned this method.


q1
q2
q3
risk free rate r
Y2
Y3


volatility of q1
volatility of q2
volatility of q3
volatility of consumption
Y2
Y3

References

Richard Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, USA, 1 edition, 1957.

Huang, Ji. Breaking the Curse of Dimensionality in Heterogeneous-Agent Models: A Deep Learning-Based Probabilistic Approach . Working Paper 4649043, SSRN, 2023a. Available at SSRN 4649043.

Huang, Ji. A Probabilistic Solution to High-Dimensional Continuous-Time Macro and Finance Models . Technical report, Working Paper, 2023b.

Bojun Geng. The economics of stablecoin fragility: Run-hazard externalities and macroprudential design. *Available at SSRN*, November 2025. https://ssrn.com/abstract=5834102 or http://dx.doi.org/10.2139/ssrn.5834102.

Ian Martin. The lucas orchard. *Econometrica*, 81(1):55–111, 2013.

Robert E. Lucas. Asset prices in an exchange economy. *Econometrica*, 46(6):1429–1445, 1978.

John H. Cochrane, Francis A. Longstaff, and Pedro Santa-Clara. Two trees. *Review of Financial Studies*, 21(1): 347–385, 2008.

Maxime Sauzet. Projection methods via neural networks for continuous-time models. 49 pages, Posted January 31, 2022, 2021.

T. Santos and P. Veronesi. Labor income and predictable stock returns. *The Review of Financial Studies*, 19(1):1–44, 2006. doi: 10.1093/rfs/hhj002.

Cristina Criddle. Bitcoin consumes 'more electricity than argentina', 2021. URL https://www.bbc.com/news/science-environment-56012952.

Binance. The truth about lost bitcoin: How many are unrecoverable, September 2025. URL https://www.binance.com/en/square/post/29979127932018.

U.S. Department of Justice. Department of Justice files largest ever forfeiture action against approximately $15B in Bitcoin currently in U.S. custody. URL https://www.justice.gov/opa/pr/chairman-prince-group-indicted-operating-cambodian-forced-labor-scam-compounds-engaged.

Markus K. Brunnermeier and Yuliy Sannikov. Macro, money, and finance: A continuous-time approach. In John B. Taylor and Harald Uhlig, editors, *Handbook of Macroeconomics*, volume 2. Elsevier, 2016.

Greg Kaplan, Benjamin Moll, and Giovanni L. Violante. Monetary policy according to HANK. *American Economic Review*, 108(3):697–743, March 2018. doi: 10.1257/aer.20160042. † symbol included in the original title.

John M. Griffin and Amin Shams. Is Bitcoin really untethered? *The Journal of Finance*, LXXV(4):1913–1964, August 2020. doi: 10.1111/jofi.12903. Volume number in Roman numerals as per the journal's style. The typical page range for this article is 1913-1964.

Victor Duarte, Diogo Duarte, and Dejanir H. Silva. Machine learning for continuous-time finance. *The Review of Financial Studies*, 37(11):3217–3271, 2024.

CNN Business. Trump's new 100% tariffs on china triggered an $18 billion crypto sell-off. URL https://edition.cnn.com/2025/10/11/business/trump-tariffs-crypto-selloff.

Jiequn Han, Arnulf Jentzen, and Weinan E. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences*, 115(34):8505–8510, 2018. doi: 10.1073/pnas.1718942115.

Jiequn Han, Arnulf Jentzen, and Weinan E. A brief review of the Deep BSDE method for solving high-dimensional partial differential equations. *Proceedings of the International Congress of Basic Science (ICBS)*, 2024.

Jose A. Scheinkman and Wei Xiong. Overconfidence and speculative bubbles. *Journal of Political Economy*, 111(6): 1183–1220, 2003.

Yukun Liu, Aleh Tsyvinski, and Xi Wu. Common risk factors in cryptocurrency. Working Paper 25882, National Bureau of Economic Research, 2020. http://www.nber.org/papers/w25882.