A Multi-Layered Model of Financial Markets: Integrating Assets, Agents, Rules, and Dynamic Valuation

Mark Brezina

October 2025

1 Abstract

We propose a comprehensive model of financial markets built on a multi-layered framework that captures the complex interplay of market structure, participant behavior, and valuation dynamics. The framework formalizes the market environment as consisting of: **(1) an assets-and-markets graph** encoding relationships between financial instruments and trading venues (e.g. which exchanges list which assets), correlations among assets (e.g. equity constituents of an index), and fundamental metadata; **(2) an agents graph representing traders and institutions**, including their group affiliations, shared behaviors or incentives, and interaction links; (3) a **hierarchical rule-based system** encoding dynamic principles and constraints (such as risk minimization, concentration limits, and return maximization objectives) that drive market evolution; (4) a **discretized multi-layer value surface** overlaid on the market graph, representing different levels of valuation for each asset – from fundamental value through financial- and information-based adjustments to the final interaction-driven market price; and (5) an explicit **time dimension** along which all these layers co-evolve. We explain how these components work together to simulate realistic market dynamics, where agent decisions and market outcomes influence one another across multiple scales. The paper includes conceptual diagrams (as placeholders) to illustrate the framework, examples of rule types (e.g. variational principles and agent-specific optimization targets), and case studies of practical applications. We demonstrate how this model can be used for **AI-driven trading strategies**, advanced risk modeling (including systemic risk and contagion), and **macro-micro integration** in analysis. We conclude with a discussion of how the multi-layer model provides a rigorous yet flexible foundation for future research, detailed market simulation environments, and production-grade strategy engines.

2 Introduction

Modern financial markets are complex adaptive systems involving networks of instruments, participants, and information flows. Traditional models often focus on isolated aspects – for example, statistical price dynamics, agent behaviors in a simplified setting, or macroeconomic factors – making it difficult to capture emergent phenomena that arise from interactions across scales. Our multi-layered framework is designed to bridge these gaps by integrating multiple representations of the market into a single coherent model. By combining a structural network view of assets and markets with an agent-based view of traders and institutions, governed by explicit rules and layered valuations, the framework aims to reproduce key dynamics observed in real markets (such as contagion, herding, bubbles, and crashes) in a controlled, extensible manner.

In this model, financial instruments and markets are represented as a graph that formalizes how assets are interconnected – through exchange listings, index membership, cross-ownership, or correlation links – and how they are characterized by fundamental attributes. Parallel to this, an agents graph captures the relationships and interactions among market participants, acknowledging that traders and institutions do not act in isolation but are influenced by information flows, competition, and sometimes cooperation (for instance, brokers forming coalitions or investors following common signals). These two graphs (assets and agents) are coupled via a set of dynamic rules: agents act on the market (e.g. placing trades on assets) in pursuit of objectives, and the market's state feeds back to influence agent decisions. The rule-based layer encodes both global principles (like conservation laws or no-arbitrage conditions) and agent-specific objectives/constraints (like utility maximization or risk limits), effectively governing the evolution of the entire system. Crucially, we introduce a multi-layer valuation surface that assigns to each asset multiple value "levels" – such as a fundamental value, a value adjusted for financial conditions (e.g. interest rates, liquidity), a value adjusted for new information or sentiment, and an interaction-adjusted value that accounts for actual supply-demand dynamics and market impact. This set of layered valuations provides a nuanced view of price formation, distinguishing between an asset's intrinsic worth and transient deviations caused by information or trading frictions.

All components of the model are time-varying and interdependent. At each time step, agents receive new information, update their strategies, and interact (e.g. trade or share signals), which leads to transactions and price changes on the asset graph. Those price changes (and any exogenous news or fundamental updates) alter the value surface layers, which in turn feed into agents' future decisions. The market thus evolves through a sequence of feedback loops: top-down influences (e.g. fundamentals or regulations shaping agent behavior) and bottom-up effects (e.g. collective trading causing systemic shifts) work in tandem. This two-way coupling is akin to a multi-layer control system where high-level conditions set the context for lower-level dynamics, and the outcomes at the micro level inform adjustments at the macro level. By capturing these feedback loops, the model can exhibit realistic phenomena such as volatility clustering from herding behavior, liquidity crises from network contagion, or macroeconomic regime shifts impacting all assets.

In the following sections, we detail each component of the framework and describe how they integrate to form a holistic market model. We provide illustrative (placeholder) diagrams to visualize the multi-layer structure and offer concrete examples of the rules and behaviors the system can encode. We then discuss practical applications of this comprehensive model – including the development of AI trading strategies in a richly simulated environment, advanced risk and contagion modeling through multi-layer networks, and the integration of macro-level and micro-level analyses. Finally, we conclude with insights on how this framework can guide future research and be operationalized in simulation tools or live strategy engines.

(Figure 1: Conceptual diagram of the multi-layer market model, showing the asset/market graph, agents graph, valuation layers, and rule-based interactions over time – placeholder.)

3 Assets and Markets Graph

At the foundation of our framework is the Assets and Markets Graph, a network representation of financial instruments and the marketplaces or structures connecting them. Nodes in this graph represent individual assets or asset entities – for example, stocks, bonds, commodities, currencies, indexes, or even derivative contracts. Edges in the graph encode various forms of relationships or couplings between these assets:

- **Market Listing and Trading Relationships:** An edge can connect an asset to an exchange or trading venue node, indicating where that asset is listed and traded. This captures the market structure (for instance, Stock ABC is listed on Exchange X and Y). These edges matter because events on one exchange (like a trading halt) can affect the asset's trading globally.
- **Hierarchical or Compositional Relationships:** Edges can represent membership relationships, such as a stock being part of an index or an ETF, or a subsidiary's equity linking to a parent company. For example, all constituents of the S&P 500 index are connected to the S&P 500 node; this way, movements in the index can propagate to its constituents and vice versa (a coupling through index arbitrage or passive fund flows).
- **Correlation and Causation Links:** Perhaps most importantly, the graph includes weighted edges denoting correlations or causal influences between assets. A high positive weight between two equities might indicate they tend to move together (due to industry relation or common macro factors), whereas a negative weight could indicate an inverse relationship (for instance, stocks vs. bonds in some regimes). These links need not imply direct causation but serve as a statistical encoding of relationships.

In practice, such a network of correlations is used in risk management to understand how shocks can transmit across assets.

Each asset node in the graph carries fundamental metadata as attributes. This can include static information like sector classification, currency, country, and fundamental ratios, as well as time-varying fundamentals like earnings, yield, growth rates, or credit ratings. By structuring asset information in a graph format, we can leverage techniques from network science and knowledge graphs. Indeed, financial knowledge graphs have become a key tool for organizing complex financial information and enabling advanced analytics. For example, a knowledge graph might integrate data from corporate filings, linking a company node to its industry, key executives, risk factors, and market data. In our context, the assets-markets graph is similarly rich: it not only tells us what assets exist and how they are related, but provides a scaffold for modeling how information or value flows through the market.

Why a graph? This network representation allows us to model propagation of shocks and information in a structured way. If a particular asset's fundamental value changes (say a company releases a stellar earnings report), the impact can spread along the graph: edges to an index mean the index's value might shift; edges to sector peers mean those peers might re-rate due to comparative valuation or sector sentiment; edges denoting derivative relationships (e.g. a stock and its futures contract) mean those prices will adjust consistently. By treating these connections formally, one can apply techniques like diffusion equations or network contagion models to simulate the ripple effects of news or trades. For instance, we could define a propagation rule that if asset A's price jumps, assets connected to A experience a proportionate move based on the edge weight (modeling spillover due to arbitrage or reallocations). This is analogous to a graph Laplacian diffusion on the network, and indeed we can draw on the theory of multilayer networks – a standard complex systems approach – to handle multiple types of relations in one framework.

(Figure 2: Assets & Markets Graph schema, depicting nodes for assets, indexes, and exchanges with edges for listings, correlations, and membership – placeholder.)

By encoding the market structure in this way, we lay the groundwork for understanding market-wide dynamics. Many real-world phenomena, like sector-wide rallies or liquidity crises, cannot be explained by looking at one asset in isolation; they are the product of network connections. Regulators and researchers increasingly use network models to gauge systemic risk – for example, representing the financial system as a network of contracts and correlations to see how distress can propagate. Our asset graph serves a similar purpose, capturing how a change or stress in one part of the market might cascade through various channels.

Additionally, this graph-based approach aligns with knowledge-driven AI techniques. Instead of treating market data as unstructured time series, we have an structured representation that algorithms (or agents in our model) can exploit. For example, an AI agent could perform a graph traversal to find all assets related to a certain commodity (oil) and adjust its portfolio when an oil price shock occurs, something that a purely statistical model might miss without explicitly encoded relationships.

In summary, the Assets and Markets Graph provides a skeletal structure of the financial domain: it ties together exchanges, instruments, indices, and fundamental linkages into a single relational map. This not only aids intuitive understanding and visualization of market structure but also enables quantitative modeling of network effects – a critical aspect for capturing co-movement of assets and cross-market dynamics.

4 Agents Graph

Parallel to the asset-centric view, our model explicitly represents the participants in the market through an Agents Graph. Here, nodes represent agents – which could be individual traders, institutional investors, market makers, banks, hedge funds, or even algorithmic trading systems. Edges between agent nodes capture various forms of interactions, influence, or relationships among these agents:

- **Communication and Information Sharing:** An edge might indicate that two agents share information or signals (for example, analysts within the same firm sharing research, or a social network link through which retail investors influence each other's decisions). The weight on such an edge could quantify the degree of influence or correlation in their strategies.
- **Competitive or Adversarial Relationships:** In markets, some participants compete (e.g. high-frequency trading firms racing each other). Edges can represent rivalry or strategic interaction, such as agents playing repeated games (bidding wars, etc.).
- **Common Ownership or Group Membership:** Agents can be grouped (e.g. all traders in a particular hedge fund, or all subsidiaries of a banking conglomerate). We can represent group affiliation by connecting agent nodes to a "group" node or simply by clustering subgraphs. Agents in a group might have aligned incentives (shared profit & loss, common risk management) – effectively shared rewards or utility. In the graph, this could be reflected as very high connectivity (everyone in the group fully shares info) or as a star topology (one central node representing the fund manager distributing strategy to sub-agents).
- **Leader-Follower Dynamics:** Sometimes one agent's actions lead others (for instance, smaller investors follow the trades of a famous investor, or algorithmic strategies follow a market trend set by a large player). Directed edges can encode this by pointing from leader to follower with a certain influence strength.

Each agent node carries a state (portfolio positions, available capital, risk tolerance, current strategy parameters, etc.) and possibly internal variables like sentiment or memory of past events. Agents also have utility or objective functions guiding their behavior. In our framework, we allow each agent i to have a utility $U_i(t)$ (or a loss function L_i to minimize) which they seek to optimize at any given time. For example, a hedge fund agent might have a utility equal to expected profit minus a penalty for risk, whereas a market maker agent might have utility based on earned spread minus inventory holding costs.

The agents graph is dynamic – the connectivity can change over time. For instance, if two traders stop sharing information (relationship breaks) or if a new major player enters the market (introducing new links of influence), the graph updates. We model this by a time-varying adjacency matrix $W(t)$ for agents, where $W_{ij}(t)$ represents the influence of agent j on agent i at time t. This admits well-studied formulations from network game theory and multi-agent systems: essentially, our agents operate in a dynamic networked game where their interactions (e.g. communication or competition) can be represented as a graph that evolves. Prior research in multi-agent finance often assumes either a fully mixed population or very simplified interactions, but here we retain the richness of arbitrary network structure.

What does this enable? Critically, the agents graph allows simulation of phenomena like **herding, coordination, and asymmetric information:**

- If many agents are connected and one starts a buying spree (perhaps based on some private signal), its neighbors in the network may observe or get tipped off and also buy – creating a herding effect that can propagate. This can generate endogenous trends or bubbles purely from the network of agents.

- Clustering of agents (communities in the graph) can lead to sub-market dynamics. For example, a cluster of quant funds might all behave similarly (because they use similar models or share personnel), leading to events like a "quant meltdown" if that cluster experiences stress. Meanwhile, another cluster (say, retail traders on a social platform) might be acting on different information. Our model can capture how these clusters interact or conflict.

- **Leader-follower behavior** can be explicitly modeled: one can designate certain agents as information leaders (e.g. a central bank or a famous investor), whose actions strongly influence others (high outgoing edge weights). This is useful in scenarios like policy announcements or large fund moves that set the tone for the market.

In an agent-based simulation context, each agent will make decisions (buy, sell, hold, adjust portfolios, etc.) based on their utility and the information they have. The information an agent has can come from:

1. **Private Signals or State:** e.g. their own predictive model or fundamental analysis results.

2. **Observations from the Market:** e.g. price changes or volume (which links to the assets graph).

3. **Neighbors (Edges) in the Agent Graph:** e.g. an agent might directly get a signal from a connected agent or infer sentiment from others' actions.

By incorporating (3), we allow social learning or strategic interaction. A well-known result in game theory is that networked agents can coordinate or fall into equilibria that differ from the isolated-agent case. For instance, network effects might amplify volatility: a single agent's random trade could, via the graph of interactions, lead to many agents trading (an information cascade).

It's also important to note that the agents graph connects to the assets graph through the actions: when an agent trades an asset, they effectively create a link between the agent node and the asset node (in an implicit bipartite sense). One could extend the model with a bipartite graph layer linking agents to the assets they trade or hold. This would show, for example, which agents are the key holders of which assets, and could be useful for analyzing concentration risk or the overlap of portfolios. Indeed, regulators sometimes examine networks of institutions to assets (who holds what) to spot crowded trades or vulnerabilities (this is similar to network-based stress testing where you see which institutions hold the same asset – a common holding is effectively a link that can transmit shocks if one agent firesales the asset).

In summary, the Agents Graph brings a multi-agent perspective to our model, making it possible to study micro-level behaviors and interactions. It acknowledges that markets are driven by the decisions of many actors, each with their own goals and connections. By explicitly including these, our framework can capture emergent effects like consensus formation, competition leading to price wars, or cooperation leading to market stability. It also sets the stage for implementing multi-agent reinforcement learning or game-theoretic analysis within the model, treating each agent as an autonomous decision-maker possibly solving an optimization problem amidst others.

5 Rule-Based Dynamic System

At the heart of the model's evolution is a Rule-Based System that encodes how everything changes over time. These rules are essentially the laws of motion for our financial world – analogous to physical laws in a physics model or the equations in an economic model. We structure the rules in a hierarchical, modular fashion, distinguishing different types or tiers of rules:

- **Tier 1: Invariant and Feasibility Rules.** These are fundamental constraints that must hold at all times, akin to conservation laws. In finance, an example could be cash conservation (no money is created or destroyed out of thin air in transactions – every buyer's spend is a seller's receipt) or the conservation of shares (the total shares of a stock remain fixed unless there is an issuance or buyback event, which would be explicitly modeled).

Feasibility constraints also include things like no negative holdings beyond certain limits (no agent can hold negative cash beyond their credit limit, etc.), enforceability of trade (a trade can only happen if there's a buyer and a seller). These rules maintain the basic integrity of the system – e.g., no-arbitrage conditions might be included here, ensuring that certain price relationships are enforced by arbitrage agents if violated (this could be encoded as an agent rule as well, but one can treat some arbitrage conditions as near-invariants in a well-functioning market).

- **Tier 2: Probabilistic Transition Rules.** These govern the general stochastic evolution of the environment in the absence of (or in addition to) agent actions. In a market context, this could include the exogenous processes: e.g., "if no trades occur, asset prices follow a diffusion or mean-reverting random process" or "news events arrive with a certain probability distribution and when they occur, they alter fundamental values according to some probability law". Essentially, this layer encodes how things tend to behave. For instance, one rule could be: "*If an earnings surprise of +X% occurs for a company, then with 80% probability its fundamental value shifts upward by a proportional amount.*" Or "*Daily returns of each asset follow a distribution with mean zero and volatility sigma, unless pushed by other forces.*" These rules might incorporate historical statistical properties and can be viewed as the background "physics" of our financial world, possibly drawn from econometric models.

- **Tier 3: Interaction and Outcome Rules.** These specify what happens when agents interact with each other or with the market. For example, a market clearing rule in an order book: if a buyer and seller submit matching orders, a trade occurs and the price is set to the trade price, volumes are deducted accordingly, etc. Another interaction rule could define auction outcomes (for instance, the opening price of an exchange is determined by an equilibrium auction given all agents' orders, which is a rule mapping orders to a single clearing price). Agent interactions might include: "*If Agent A and Agent B are connected and A's trade is large enough, B reacts by adjusting her own position by a rule (perhaps mimicking or hedging).*" These can be quite complex and are often where one encodes strategic behavior outcomes or institutional mechanisms (like circuit breakers: "*If volatility exceeds threshold, exchange halts trading for 5 minutes*" would be a rule).

The rules can also be seen from another angle: objective functions and constraints guiding optimization. Many dynamic principles in markets are naturally framed as optimization problems:

- **Minimization/Maximization Objectives:** Each agent might explicitly solve an optimization at each time step. For instance, a portfolio manager agent could rebalance by "*maximize expected return minus λ times variance of return*" (a classic Markowitz objective) subject to constraints.

A market maker might have an objective "*minimize inventory risk plus penalty for not fulfilling order flow*". These objectives translate into rules when we simulate the agent's behavior – essentially, the rule is that the agent takes *the action that optimizes its objective given the current state.* This could be deterministic (the optimal solution) or probabilistic (with some randomness or exploration).

- **Optimization Constraints:** There are often constraints like "*don't allocate more than X% to any single asset" (minimize concentration risk), or regulatory constraints like capital requirements that effectively say "if losses exceed Y, agent must de-lever*". These can be encoded as hard rules that override other actions (e.g., a rule might forcibly liquidate positions if a margin call is triggered), or as parts of the agent's optimization (penalty terms). For example, a rule could enforce that no portfolio's Value-at-Risk exceeds a threshold – if an agent tries to take a position that would break this, the rule prevents it or scales it down.

- **Variational Principles:** On a system-wide level, one can sometimes identify a principle like a variational problem that the market tends to solve. For instance, in an efficient market with risk-neutral pricing, prices adjust to eliminate arbitrage – one could view this as the market "minimizing free energy" in analogy to physics, where the free energy corresponds to exploitable arbitrage profit. In our model, we could incorporate such a principle: e.g., a rule that if any price discrepancy exists (say an index future vs. the underlying stocks) beyond a small threshold, arbitrage agent actions will quickly correct it. In effect, the market moves toward an equilibrium that *minimizes arbitrage opportunities* (an analogy to an energy minimum). Similarly, we might state that agents collectively tend to maximize utility, which in equilibrium could relate to some global optimum given constraints (though in a game, it's more of an equilibrium than a single optimization). We include variational ideas to inspire certain rule designs – for example, one could derive a price formation rule by assuming market prices maximize some social welfare function or minimize overall trading "stress".

To make this concrete, here are examples of rules in our system:

- **Price Update Rule:** If net buy volume for an asset is positive in a time step, its price ticks up; if net sell volume, price ticks down, with magnitude proportional to imbalance (this is a simple linear price impact rule, ensuring market clearing with slippage). This encodes basic supply-demand interaction.

- **Risk Limit Rule:** Any agent's portfolio Value-at-Risk (VaR) must remain below a threshold. If an agent's actions would cause VaR to exceed the limit, the rule forces a reduction in position (e.g. the agent must sell some holdings to reduce risk). This implements minimize risk as a hard constraint at agent level.

- **Diversification Rule:** A portfolio construction rule might impose that no single asset can be more than 10% of the portfolio (minimize concentration risk). Agents may have to rebalance if concentrations grow (perhaps due to price changes) – a rule could trigger automatic partial selling of an oversized position.

- **Trend Following Agent Rule:** Suppose an agent type is a trend follower. A simple rule for them: "*If the information-adjusted price of an asset is above its fundamental value by more than X% and rising, buy more; if below by Y%, sell.*" This is a heuristic rule that encodes a strategy objective (maximize momentum gains) with an implicit constraint (don't deviate too far from fundamentals).

- **Market Maker Agent Rule:** "*Keep the asset price around the information-adjusted value by providing liquidity: if price deviates* $+\delta$*, supply liquidity (sell) to push it down; if* $-\delta$*, buy to push up.*" This rule has the market maker minimizing deviation (and maybe maximizing spread capture).

- **Information Shock Rule:** "*If a news event occurs that affects asset A's fundamental value by* Δ*, then over the next N time steps, the information-adjusted value of A will move toward the new fundamental with some decay factor, and correlated assets will also adjust by a fraction of* Δ *according to correlation weights.*" This rule dictates how information propagates and translates into price changes gradually (not all at once, perhaps due to limits to arbitrage or gradual information diffusion).

- **Learning/Adaptation Rule:** Agents can have rules to adapt their behavior: "*If an agent's strategy underperforms (utility falls below expectation for Q periods), then with some probability it switches strategy or adjusts parameters.*" This introduces evolutionary dynamics (agents exploring different strategies to maximize long-term returns).

All rules in the system can be codified as mathematical mappings from conditions to outcomes, possibly with randomness (to allow stochastic events). For example, a rule could be $r : (agentstate, marketstate) \mapsto actiondistribution$ – essentially defining a policy for a class of agents.

Hierarchy and precedence: Often these rules operate at different frequencies or priorities. In implementation, we might first apply Tier 1 rules (ensuring no invariants are violated), then apply Tier 2 (simulate exogenous stochastic moves), and then Tier 3 (agent interactions and decisions) within each time step. This ensures, for example, that an agent can't violate a hard constraint because the Tier 1 rule would prevent or correct it.

It's worth noting that rule-based modeling of markets has precedents in agent-based modeling (ABM) literature, where researchers define sets of if-then behaviors for agents and observe emergent phenomena. Declarative rule-based agent models (like those by Lotzmann & Meyer) have shown that even simple rule sets can lead to complex dynamics. Our approach extends this by embedding the rules in a multi-layer structure with explicit graphs and by incorporating optimization principles directly.

Finally, these rules are modifiable and extensible, which is crucial for practical use. For instance, if a user of this model wants to test a new market policy (say a transaction tax), one would introduce a rule at the appropriate tier (e.g., an interaction rule: "for each trade, a tax of X% is deducted from proceeds"). The flexibility of a rule-based system means our framework can act as a sandbox for scenario analysis: turning rules on or off can simulate different market designs (continuous trading vs. call auction, presence or absence of certain kinds of agents, etc.). It also means the model can be calibrated: rules have parameters (like reaction speeds, risk thresholds) which can be tuned to fit historical data or plausible scenarios.

6 Discretized Multi-Layer Value Surface

One of the novel components of our framework is the multi-layer value surface that lies on top of the assets graph. Think of this as a set of transparent overlays on the network of assets, where each overlay (layer) represents a different notion of "value" or "price" for each asset. While a real market ultimately has one traded price per asset at a time, in modeling it is useful to distinguish between various value concepts that influence that price. We define (at least) four layers:

Fundamental Value Layer. This layer encodes the intrinsic or fundamental value of each asset. For a stock, this could be based on discounted cash flows, book value, or some model of fair value given economic conditions. It changes relatively slowly, driven by fundamentals like earnings reports, economic data, or structural changes in the company or asset. Fundamental value can be thought of as the price around which long-term investors believe the asset should trade. In our model, each asset node i has a fundamental value $F_i(t)$ at time t. This could be set by a fundamental model or given exogenously. It's like a baseline evaluation layer in the sense of a scalar field of utility or worth over assets.

Financially Adjusted Value Layer. This second layer takes the fundamental value and adjusts it for financial market conditions. This includes factors like interest rates (discount rates), inflation, liquidity premium, or risk premia required by the market. For example, if interest rates drop significantly, the financially adjusted value of equities might rise above the raw fundamental (since future earnings are discounted at a lower rate). Conversely, if an asset is fundamentally worth 100$ but is very illiquid or high-risk, the financially adjusted value might be lower (investors demand a discount). This layer can be thought of as the output of classical financial pricing models (e.g. option pricing, CAPM adjustments, etc.) that take fundamentals and add market-consistent adjustments. It evolves with macro-financial indicators and risk appetite.

Information-Adjusted Value Layer. The third layer incorporates information, news, and sentiment on top of the financially adjusted baseline. This reflects the market's expectation of near-term factors, momentum, or transient informational advantages. For instance, a positive news rumor might push the information-adjusted value above what fundamentals and broad financial conditions alone would suggest, because traders expect short-term outperformance. This layer captures how ongoing information flow (earnings surprises, analyst upgrades, geopolitical news, social media sentiment) modifies the perceived value of each asset. One can imagine this as what the price would be if all agents instantly and rationally processed the latest information – it's closer to a short-term fair value given current knowledge. It is more volatile than the first two layers, because information arrives frequently and can change perceptions quickly.

Interaction-Adjusted Value (Market Price) Layer. The final layer represents the actual market price that results after considering the interactions and frictions among agents. Even if everyone agrees on information-adjusted value, the actual traded price can deviate due to factors like order execution, short-term supply-demand imbalances, strategic trading (someone might push price temporarily), or constraints (not everyone can trade at once, some have liquidity needs). This layer effectively is the real-time transaction price, and it's influenced by microstructure effects. We call it interaction-adjusted because it's the value after accounting for who is interacting with the market at that moment – e.g. if many are trying to buy simultaneously, the price can overshoot above the info-adjusted value; if a large seller is unloading inventory, price might temporarily fall below the consensus value. This corresponds to the concept of market impact and transient price pressure.

These layers are conceptual, but we can formalize them. One way is to treat them as fields $V^{(fundamental)}(i,t)$, $V^{(financial)}(i,t)$, $V^{(info)}(i,t)$, and $V^{(market)}(i,t)$. These fields are attached to each asset i. They are not independent: typically we have. $V^{(fundamental)}(i,t) \to V^{(financial)}(i,t) \to V^{(info)}(i,t) \to V^{(market)}(i,t)$ where each arrow means "informs or sets a baseline for". For example, $V_i^{(financial)}$ might equal $V_i^{(fundamental)}$ adjusted by some function of interest rates and volatility. $V_i^{(info)}$ might start from $V_i^{(financial)}$ and then incorporate recent news sentiment (perhaps via a multiplier or additive term derived from textual analysis or recent returns). Finally, $V_i^{(market)}$ is determined via the trading dynamics considering supply/demand.

One can draw an analogy to signal processing: fundamental value is like a low-frequency signal (slow-moving trend), information-adjusted adds medium-frequency fluctuations (news-driven jumps), and interaction-adjusted adds high-frequency noise and microstructure effects. Over long horizons, one expects the market price to gravitate around the fundamental value (mean reversion to fundamentals), but at any given time, it could be off by a significant amount due to layers 2–4.

We implement these layers in the model by having the value surface evolve on the asset graph nodes. There can be rules for how one layer influences the next. For example:

- **Financial adjustment rule:** If interest rates $r(t)$ change, adjust all $V_i^{(financial)}$ proportionally to their duration or rate sensitivity. If volatility

(VIX) spikes, maybe reduce $V^{(financial)}$ for risky assets to reflect higher risk premium.

- **Information update rule:** On a news event for asset i, instantly adjust $V_i^{(info)}$ by some delta (e.g. +5%). Also, propagate that update to related assets' info layer values: if asset j is correlated or connected to i, update $V_j^{(info)}$ by say $correlation_{weight} \cdot 5\%$ (this mimics info diffusion across the graph). This propagation can be thought of as diffusion on the value surface across the network.

- **Convergence rule:** In absence of new shocks, $V_i^{(info)}$ should slowly drift toward $V_i^{(financial)}$ as hype dies down or lack of news means nothing sustains the divergence. This could be modeled as a mean-reversion process.

- **Market price formation rule:** $V_i^{(market)}$ tends toward $V_i^{(info)}$ but is influenced by order flow. If there's a buy imbalance, $V_i^{(market)}$ may exceed $V_i^{(info)}$ temporarily. One can treat $V^{(info)}$ as an anchor and $V^{(market)}$ as fluctuating around it depending on immediate net demand. Mathematically, one could write: where α is a speed-of-adjustment (to info value) and β translates order flow into price impact. This way, if there's no trading imbalance, price moves gradually toward the info-fair-value; heavy buying pushes it up, heavy selling pushes it down, but it doesn't stray indefinitely because eventually those pressures subside and price mean-reverts toward the info-driven value.

By explicitly keeping track of these layers, our model can distinguish the sources of price movement. For example, suppose we observe a price jump in $V_i^{(market)}$. Is it because the fundamental value changed (earnings surprise)? Or just a trading imbalance (a large fund selling)? In the model, that difference is clear: a fundamental change would alter $V^{(fundamental)}$ and thus all layers above it, whereas a pure technical sell-off would show $V^{(market)}$ dropping below $V^{(info)}$ (temporarily) while $V^{(info)}$ and $V^{(financial)}$ remain unchanged. Agents in the model can even be designed to respond differently in these cases – e.g., a value investor agent might buy when $V^{(market)}$ is significantly below $V^{(fundamental)}$, interpreting it as a bargain due to temporary market distress.

This multi-layer approach resonates with some real-world investment thinking: investors often speak of "price vs. fair value" or "fundamentals vs. technicals". Here we are formalizing those notions. It also aligns with research attempts to combine macro and micro data for better forecasting, since macro fundamentals (affecting layer 1 and 2) combined with micro price patterns (layer 4) can improve predictions. A recent example is an AI model that fused macro-index data with micro-stock data to improve price forecasts. That study showed that integrating broad market context (like index trends, which relate to fundamental and macro-financial layers) with granular stock signals (which include micro dynamics) yielded significantly more accurate predictions – in fact, the integrated model reduced prediction error by over 36% for a stock like Apple compared to models using micro data alone. This underscores the value of our layered approach: by considering both macro-fundamental valuations and micro interaction effects, one can achieve a more comprehensive understanding and potentially better forecasts of market behavior.

Another benefit of the value surface concept is in visualization and diagnostics. One can imagine plotting these layers over time for a given asset: the fundamental value might be a smooth slowly rising line, the financially adjusted value a slightly more wiggly line following interest rates, the info-adjusted value jumping up and down with news, and the market price oscillating around the info line. The gaps between the layers tell stories (e.g. large gap between market and info layers might indicate a liquidity crunch or exuberant trading beyond fundamentals). This can help analysts or AI agents identify trading opportunities (mean reversion trades, arbitrage between an ETF's price and its net asset value which is analogous to a deviation between market layer and fundamental layer).

In implementation, the value surface is discretized, meaning we treat the values on each asset at discrete time steps. The term "surface" is used by analogy to a spatial field, but here the "space" is actually the network of assets. One could literally imagine a 2D visualization where the x-y plane is some representation of asset space (maybe clustered by sector or so) and the vertical dimension has multiple layers for different types of valuation. However, it might be more straightforward to think of it as a table of values: for each asset (row) we have columns for fundamental, financial-adjusted, info-adjusted, market price, at each time.

Under the hood, updating these layers follows specific equations. The initial draft of this framework provided an example of a diffusion-like update for an evaluation field on a grid. In our context, a diffusion term could represent how fundamental value changes at one asset (site) slowly bleed into connected assets' values (like how a shock gets shared across the network), with a coefficient controlling the speed. Additionally, there are source terms from agents: e.g., if agents inject capital or take actions that effectively add "value" (like government bailout increases fundamental of many assets), that can be a source term in the update equation. While the specifics can get technical, the key idea is that each layer's evolution can be described by equations considering both neighbor influences (network diffusion) and external inputs (news, agent actions).

To summarize, the discretized multi-layer value surface equips our model with a rich representation of price and value. It disentangles different contributions to an asset's price and allows our simulated agents to operate with more nuanced information (e.g., an agent could specifically focus on trading the gap between market price and fundamental value, or gauge market sentiment by looking at the gap between info value and fundamental). This layered approach mirrors how real-world market analysis works – combining fundamentals, macro factors, sentiment, and order flow – but here it's systematically built into the model's state. It ultimately helps the model capture the full spectrum from long-term value to short-term price noise, which is critical for a unified macro-micro modeling of financial markets.

7 Temporal Evolution and Layer Interactions

With all pieces of the framework defined (assets graph, agents graph, rules, and value layers), we now describe how the entire system evolves over time in an integrated fashion. We work in discrete time steps (t = 0, 1, 2, ...) for conceptual clarity, although in implementation these could be, say, milliseconds, minutes, or days depending on the desired resolution. At each time step, a sequence of updates occurs that advances the state of the world from time t to t+1.

Global State Representation: We denote the global state at time t as $X(t) = \{G_{assets}(t), G_{agents}(t), V_{surface}(t), S_{agents}(t)\}$, Where $G_{assets}(t)$ includes all asset nodes, their attributes (fundamental values, etc.), and edges (couplings at time t); $G_{agents}(t)$ includes agent nodes and interaction edges at time t; $V_{surface}(t)$ denotes all layers of values for each asset at time t (fundamental, financial, info, market); and $S_{agents}(t)$ denotes the internal state of each agent (portfolio holdings, cash, strategy state, etc.) at time t. The rules then describe how $X(t)$ transforms into $X(t+1)$.

A possible high-level algorithm for each time step:

1. **Apply Exogenous Changes:** Incorporate any external changes that occur at time t. This includes new fundamental information (e.g. economic indicators, earnings announcements) updating the fundamental layer, any changes to the asset graph structure (a new asset IPO, or an asset delisting, or a new correlation emerging – though structure changes may be less frequent), and changes to agent graph structure (e.g. a new agent entering, or communication links shifting). These are like external inputs to the system.

2. **Determine Agent Actions:** Each agent observes the current state (with whatever limitations appropriate – maybe not all agents see all info) and decides on an action. Actions could be: placing orders to buy/sell certain assets, adjusting their portfolio, sending signals to other agents, etc. The decision of each agent is governed by their rules/objectives. In many cases, this is an optimization or heuristic: e.g., maximize U_i given current prices and expectations. If we frame one step as a one-shot game, agents might even strategically anticipate others (though that becomes game-theoretic; a simpler approach is agents react to state assuming others' current actions are known or ignoring others momentarily). In a reinforcement learning interpretation, each agent's policy π_i outputs an action given state. This stage produces a set of intended actions (like a list of orders).

3. **Match and Settle Actions (Market Mechanism):** The agents' actions now interact through market rules. For example, all buy and sell orders go into an order book or matching engine. The rule system then clears the market: matches buyers and sellers, determining new transaction prices for each asset (this updates the interaction-adjusted value layer = market prices). If agents' actions conflict with constraints, some

might be curtailed here (e.g. if total demand far exceeds supply, not all orders fully execute, or a price jumps until equilibrium). Essentially, this is where the Tier-3 interaction rules (like price impact, clearing, and trade execution rules) come into play. The output is the updated holdings (who bought/sold how much) and the new market prices $V^{(market)}(t+1)$.

4. **Update Agent States:** Based on what happened (trades executed, price changes), update each agent's state. This means cash balances go up or down, portfolios change, unrealized P&L is accounted for (which might affect their capital or risk metrics), etc. Also update any agent-specific variables like trust or strategy settings if they have adaptation rules (for instance, if an agent lost a lot of money, it might become more risk-averse – which could be a rule encoded in their utility or decision policy).

5. **Recompute Derived Values:** Now that we have new market prices and any fundamental news, we update the value surface layers. For instance, if $V^{(market)}$ moved, we check if that implies any adjustments to $V^{(info)}$ (one could argue $V^{(info)}$ should actually be an underlying driver, but if the market moved without news, perhaps it creates a feedback where people re-assess something). Usually, it's the other way: $V^{(info)}$ would have been updated before market moves if there was real news. But also consider that large market moves themselves can become information (e.g. a sudden crash might signal some agents know bad news, so others update their info-based valuations downward). Our model can accommodate that reflexivity: a rule can state "if price drops more than X with no known news, reduce info-layer value a bit for others because it might indicate hidden information." We also propagate any shock through the asset graph (diffusion of fundamental or info values as described). Essentially, the fundamental layer might update from exogenous data, the info layer updates from fundamentals plus any observed sentiment or momentum, and financial layer updates if needed (e.g. interest rates moved this step). These calculations could use equations like shown earlier (diffusion, mean reversion, etc.).

6. **Enforce Constraints and End-of-Step Adjustments:** After tentative updates, enforce any Tier-1 invariant rules. For example, if some rounding or discrete effects caused a slight imbalance (e.g., more cash went out than in due to a bug or approximation), correct it. Ensure all conservation laws hold (they should by construction, but this is a safety check). Also, if any agent now violates something (maybe after trades, an agent's leverage is above allowed), one might immediately apply a corrective measure (like margin call liquidation) at this end-of-step or at next step start.

7. **Proceed to Next Time Step:** The state is now consistent at $t+1$, and the cycle repeats.

Throughout this process, stochastic elements (from Tier-2 rules or random agent decisions) can introduce randomness, meaning if we simulate multiple runs, we'd get a distribution of outcomes.

A key point is that all the layers (assets, agents, values, rules) are deeply intertwined in each step. Agents act on the asset graph (via trades that alter market prices). The asset graph's state (prices, correlations) feeds back to the agents as they make decisions. The value layers provide intermediate signals – e.g., an agent might specifically use the difference between $V^{(info)}$ and $V^{(market)}$ as a trigger to act. The rules tie it all together by specifying how one affects the other (e.g., a rule in the agent's decision might be "if market price is below fundamental by ¿20%, buy" – explicitly linking the value surface to agent action).

Two-Way Coupling: We want to emphasize the interplay with an example. Consider a scenario of a market shock:

- At time t, a sudden fundamental news arrives that significantly lowers the fundamental value of Asset A (say a biotech company's drug trial failed). In step 1, we update $V_A^{(fundamental)}$ way down, and accordingly $V_A^{(financial)}$ and $V_A^{(info)}$ also drop (because this is clear news). The asset graph has edges from A to other biotech stocks and perhaps to a biotech index; through a diffusion rule, those related assets' info-layer values also decrease a bit (on anticipation of spillover).

- Now agents at time t see this. Many agents holding Asset A will want to sell (to minimize their losses), and perhaps some short-seller agents jump in to short because they expect others will sell. These intentions are formed in step 2. Meanwhile, some contrarian or market-maker agents might plan to buy if the price overreacts.

- In step 3, all these orders hit the market. The selling likely far outweighs buying initially, so Asset A's market price $V_A^{(market)}$ plunges. Some trades for related stocks might also trigger if their holders panic or if stop-loss rules are in place (so those prices also drop a bit).

- Step 4, agent states update: those who sold take losses but have cash, those who bought (if any) now hold assets that dropped. Some agents might hit risk limits now.

- Step 5, recompute values: After the dust settles, maybe Asset A's market price is below even the new fundamental (because of overshoot). The info-adjusted value might be set equal to the fundamental now (since news fully explains it, there's no additional info premium). But if it's overshot, that might create an information discrepancy – the model might say $V_A^{(info)}$ is actually a bit above $V_A^{(market)}$ if it assumes rational expected value. This could signal a potential rebound. Also, because many assets in the sector fell, their info-layer might recover a bit if the rule says "overreactions correct". Or if the sentiment is truly contagion, maybe all biotechs' info-layer stays depressed.

- Now at time t+1, agents see that Asset A's price is very low relative to either its new fundamental or relative to others. A value investor agent may decide "this is a oversold situation" and buy. Short-term traders might also anticipate a bounce (a technical correction). So at t+1, buy orders come in. Then market clears and price bounces back some.

This example shows how the layers (fundamental info vs market price) and the agent behaviors interact in time. Notably, the model can exhibit reflexivity – where price moves not fully justified by fundamentals can themselves influence perceptions and further moves (as famously described by George Soros). In our framework, reflexivity is handled by rules that allow feedback from $V^{(market)}$ to $V^{(info)}$ (people treating price as information) and from agent behavior back to fundamentals (e.g., a market crash could raise a firm's cost of capital, hurting its fundamental prospects – we could model that too).

The time evolution equations underlying this are mathematically rich. As seen in the initial formalization, one can write state-update functions f for each component. For instance, the state of an asset (like an "object" or an "edge" in the initial draft's language) can be updated by a function f_e that depends on its own last state, neighboring states, the surface values at its location, and some randomness. Concretely, for an asset price: $price_i(t+1) = price_i(t) + g\Big(\, price_i(t),\, V_i^{(info)}(t),\, \{price_j(t) : j \sim i\},\, orderflow_i(t)\Big) + \eta$, where $j \sim i$ denotes neighboring assets connected in the graph (for cross-impact), and η is noise. The exact form of g encodes the rules (like the price impact formula, mean reversion, etc.). Similarly, agent state update can be seen as:

$$portfolio_i(t+1) = h(portfolio_i(t), actions_i(t), prices(t+1)$$

For example, if agent i decided to buy 10 shares of asset A, after execution, its portfolio of A increases by 10 (provided the order filled fully) and cash decreases accordingly – that's a simple accounting update.

One more interesting aspect of temporal evolution is the possibility of different time scales. Some components evolve fast (market prices can change in milliseconds), while others are slow (fundamentals may update quarterly). Our framework can accommodate multiple time scales by either simulating with a fine time step and having some changes occur rarely, or by having sub-stepping for fast dynamics. For example, one could have an inner loop for order-book dynamics (fast ticks) nested inside an outer loop for daily fundamental changes. This is complex to coordinate, but the rule hierarchy helps (fast rules vs. slow rules can be layered).

8 Practical Applications of the Multi-Layer Model

A key strength of this comprehensive model is that it is not merely a theoretical construct – it has direct practical applications in various domains of finance. By capturing a wide range of dynamics (structural, behavioral, informational, etc.), the framework can serve as a foundation for advanced tools and strategies. We highlight a few important use cases:

8.1 AI-Driven Trading Strategies and Market Simulation

Developing and testing AI trading strategies is greatly enhanced by a rich simulation environment. Our multi-layer model can function as a sophisticated simulated market where AI agents (e.g. using reinforcement learning or other algorithms) can be trained and evaluated. Unlike simplistic simulation models that might assume i.i.d. returns or single-agent optimization, our environment provides realistic scenarios: multiple agents interacting, regime shifts when major news hits, network effects causing cascading events, and so forth.

For example, one could deploy a reinforcement learning agent whose goal is to maximize profit over time. In our environment, it must learn not just to trend-follow or mean-revert on a single time series, but to navigate a market of interacting assets and agents. It could learn strategies like detecting when a certain sector is underpriced relative to its fundamentals and taking a long position while hedging others, or identify when a competitor agent is about to cause a price impact and trade in advance. The presence of the agents graph and layered pricing means the AI can also learn to anticipate other agents' behaviors (a step toward game-theoretic intelligence). Indeed, if we incorporate a mechanism for the learning agent to model others (say via a belief system or using the knowledge graph of agent types), it moves into the realm of multi-agent reinforcement learning, possibly approximating a Stackelberg leader-follower scenario where the agent tries to predict reactions of others and optimize accordingly.

The value of such a simulation is evidenced by industry trends: large firms and researchers are building ever more realistic market simulators to train trading algorithms. For instance, Microsoft Research's MarS (Market Simulation) engine uses generative AI models to emulate realistic order flow and market microstructure. They create a "large market model (LMM)" which aims to reproduce both fine-grained order-level behavior and macro-level dynamics. The goal is to have an environment where AI can be trained on synthetic yet realistic data, overcoming the scarcity of certain market scenarios in historical data. Our framework aligns with that philosophy: by including multiple layers, we ensure that the simulation isn't one-dimensional. A well-calibrated version of our model could generate scenarios of market panic, gradual bull runs supported by fundamentals, sector rotations, liquidity squeezes, etc., providing a rich training ground for algorithms.

Additionally, because our model is grounded in transparent rules and structures, it can be used for what-if analysis of trading strategies. A portfolio manager could plug in a proposed algorithm and simulate how it performs not just in a historical replay but in counterfactual situations: What if many agents follow a similar strategy? Will there be crowding (the agents graph can simulate that)? What if a new regulation comes (we modify a rule and see the impact)? This goes beyond backtesting – it's more akin to stress-testing strategies under simulated conditions.

In short, for any firm or researcher working on AI-based trading, having a multi-layer market model is like having a "market gym" – an environment to safely test and hone strategies. This could accelerate development of robust trading bots that can handle complex real-world conditions. It also helps avoid overfitting to simplistic assumptions, since the model can be as unpredictable in some ways as real markets (especially if stochastic rules are included to introduce variation).