Can LLMs Discover Novel Economic Theories?

Alejandro Lopez-Lira* University of Florida

Sina Seyfi† Aalto University

Yuehua Tang‡ University of Florida

March 16, 2026

Abstract

We propose AutoTheory, a system for discovering economic theories using large language models and evolutionary search. The system combines LLM-based theory generation with mathematical auditing, pseudocode-to-code translation with review, calibration, quantitative scoring, and simulated peer review. To promote diversity, each run draws a random combination of expert personas from 20 methodological perspectives. We apply the system to two empirical puzzles: the price-multiplier scaling puzzle and the equity term structure (dividend strip) puzzle. For the price-multiplier puzzle, the system discovers multiple distinct mechanisms, spanning attention costs, limited diversification, portfolio constraints, and liquidity channels, with the best models matching all three empirical targets with as few as two free parameters.

Keywords: LLMs, theory generation, evolutionary algorithms, asset pricing, price multipliers, automated discovery

*alejandro.lopez-lira@warrington.ufl.edu

†sina.seyfi@aalto.fi

‡yuehua.tang@warrington.ufl.edu

1 Introduction

The development of economic theory traditionally relies on creativity, intuition, and domain expertise to identify mechanisms that explain empirical patterns. Much of this process is informal and hard to formalize: researchers search for promising hypotheses and theories using intuition and other "soft" knowledge. Still, advances in machine learning show that algorithms can systematically support tasks previously viewed as informal. In this spirit, Ludwig and Mullainathan (2024) show that machine learning can discipline hypothesis generation. This raises a natural question: can the generation of economic theories be formalized and automated using large language models (LLMs)?

We address this question by building a *general-purpose system* for LLM-based theory discovery in economics. The system is puzzle-agnostic: it takes as input a specification of an empirical puzzle (target values, baseline model, calibration parameters) and outputs candidate theories that are economically plausible, parsimonious, and fit the data. It accompanies these theories with mathematical formulations, executable code, optimized parameters, and simulated referee assessments. The system makes theory search scalable: it generates many candidate mechanisms, imposes feasibility and internal-consistency checks, disciplines them using empirical targets, and retains the strongest proposals. Numerical targets (observed empirical moments the theory must match) serve as an objective, LLM-independent quality filter. A candidate theory either reproduces the data or it does not, regardless of how persuasively the LLM describes it. This numerical anchor is what makes automated theory search viable: it grounds the evaluation in data rather than in the LLM's own judgment.

Why should LLMs be able to discover novel economic theories? A classic view of creativity and scientific discovery is that progress emerges through a generate–test–retain process: produce many ideas, evaluate them, and keep the best (Campbell, 1960). Simonton (1997) likewise argues that creative success is partly a numbers game, with higher output increasing the likelihood of major contributions. LLMs fit naturally into this frame-

work because they can cheaply generate many diverse candidate theories by recombining existing concepts and mechanisms in new ways. Repeated sampling from an LLM lets researchers search more broadly over plausible explanations; if useful theories occur with non-zero probability, more draws increase the chance of finding one. Our system implements this generate–evaluate–select logic at scale.

For economists, automating this creative search stage changes the production function of theory. Instead of relying on a small number of hand-crafted mechanisms tailored to a single puzzle, researchers can systematically explore a broader model space, record failures as well as successes, and compare competing explanations using common quantitative and qualitative criteria. In practice, economists often learn by surprise: a regularity appears in the data, the current model does not explain it, and researchers iteratively revise mechanisms until a coherent explanation emerges. Heckman and Singer (2017) describe this as an abductive process. LLMs can support this process as a scalable engine for proposing candidate mechanisms that humans can scrutinize.

Methodologically, we build a puzzle-agnostic system for theory search, using OpenAI's open-weight reasoning model `gpt-oss-120b` (OpenAI et al., 2025) for all LLM-based stages. The pipeline starts from a puzzle specification and generates candidate theories with LLM-guided evolution. Each candidate undergoes an independent mathematical audit, translation through structured pseudocode into executable code (with review at each stage), and calibration against empirical targets. A scorer then grades each model on quantitative fit, economic plausibility, and parsimony, and a referee LLM produces a final score and feedback.

To promote diversity, each run draws a random combination of expert personas from 20 methodological perspectives (e.g., bayesian, behavioral, physicist, minimalist). This structure keeps creativity and discipline together: broad generation at the front, hard filters and transparent evaluation at the back. Unlike recent AI-discovery systems in mathematics and the natural sciences, which optimize over fully formal objects and verifiable

objectives, our setting involves judgment-laden models of human behavior; our contribution is to make this search disciplined and reproducible despite that ambiguity.

We opt for three different strategies for the LLM-based theory generation. These three strategies are *random* (propose a mechanism from scratch), *mutate* (modify a previously successful theory, guided by its referee report), and *crossover* (recombine elements of two successful theories). A broad literature motivates this design: impactful discovery often comes not from wholly unprecedented ideas but from unusual recombinations of existing knowledge, methods, or data (Fleming, 2001; Nerkar, 2003; Uzzi et al., 2013; Fortunato et al., 2018; Teplitskiy et al., 2022; Park et al., 2023; Yu and Romero, 2024). Random search preserves exploration; mutate and crossover let the system exploit and recombine partial insights discovered earlier in the search.

The price-multiplier puzzle illustrates the system. Gabaix and Koijen (2023) define the price multiplier M as how much an asset's price changes when demand shifts by one dollar. Li and Lin (2022) find that the price multiplier is not constant across levels of aggregation. They show that the price multiplier is larger at the style-level than at the idiosyncratic level, and that the ratio of style-level to idiosyncratic price multipliers observed in data ($M_{\text{style}}/M_{\text{idio}} \approx 3.6$) differs dramatically from the prediction of the standard CARA-normal model (≈ 14.5). This factor-of-four gap is economically important and has a clear quantitative target, making it suitable for automated evaluation.

For the price-multiplier puzzle, we run the system for 1,000 iterations (458 completing all pipeline stages) and find multiple distinct mechanisms in the top tier (four models tied at referee score 84/100). These include attention-cost models, dual-agent limited-diversification frameworks, liquidity-adjusted CRRA models, and Bayesian learning with portfolio-breadth costs. For example, in the Bayesian-Breadth-Cost CARA model, investors face noisy signals about common factors and quadratic costs of portfolio breadth; in equilibrium, this structure produces a flatter style-to-idiosyncratic multiplier ratio, matching the data with only three parameters. All top mechanisms fit the data with two

to five free parameters. We also apply it to the equity term structure puzzle (van Binsbergen et al., 2012), where short-term dividend strips have roughly twice the Sharpe ratio of the market, a pattern that standard consumption-based models struggle to match. This puzzle, with ten moment conditions including conditional targets, is substantially harder, and the best model scores 58/100 across 2,153 runs (257 completing all stages).

The remainder of the paper proceeds as follows. Section 3 presents a simple model of theory discovery under parallel LLM search. Section 4 describes the architecture of our LLM-based theory-discovery system. Section 5 reports our experimental analysis for both puzzles and compares the results. Section 6 discusses capabilities, limitations, and implications of LLM-based theory discovery. Section 7 concludes.

2 Related Literature

Our work sits at the intersection of three literatures: AI-assisted scientific discovery, psychological and formal models of creativity, and demand-based asset pricing. We contribute to the first by demonstrating LLM-based theory generation in economics; to the second by implementing a formal generator–selector loop consistent with blind variation and selective retention; and to the third by producing theoretical mechanisms that explain empirical puzzles in asset pricing.

2.1 Creativity as Variation and Selection

A longstanding way to formalize creativity in science and mathematics is as a generate–test–retain process: generate many candidate ideas (variation), evaluate them (selection), and keep or elaborate the best (retention). Campbell (1960) articulates this in the "blind variation and selective retention" (BVSR) model of creative thought: wide variation is produced by trial-and-error or fluency of ideas, and a critical or selective function decides which variants are retained. Amabile (1983) offers a complementary componential view:

creativity depends on domain-relevant skills, creativity-relevant processes, and task motivation within a supportive environment, a framing that aligns with treating the LLM as a variation engine operating in a structured evaluation environment. Empirically, creative output behaves like constrained stochastic behavior: Simonton (1997) shows that hits scale with attempts (an "equal-odds" style regularity) and provides a predictive model of career trajectories and landmarks. In technological search, Fleming (2001) shows that recombinant exploration (novel combinations of existing components) increases variance in outcomes: most combinations fail, but occasional combinations generate large advances. This pattern reinforces the intuition that broad exploration can expand the space of viable candidates.

Evidence on scientific practice then highlights a selection bottleneck. Foster et al. (2015) document an "essential tension" in scientists' research strategies: innovative directions are underused relative to their potential impact because rewards are uncertain and delayed. Wang et al. (2017) show that bibliometric indicators are biased against novelty: highly novel papers receive delayed recognition and are systematically disadvantaged by short-term citation metrics despite often achieving top long-term impact. Together, these findings motivate systematic exploration at scale: when human and institutional incentives favor tradition, automated generation can broaden the set of candidate theories that receive serious evaluation. The standard definition of creativity in psychology requires both originality and effectiveness: an idea or product must be novel and also useful or appropriate (Runco and Jaeger, 2012). We adopt this criterion by requiring candidate theories to be novel relative to a reference corpus and to pass verification (e.g., consistency checks and empirical targets). Our pipeline instantiates BVSR with the LLM as the variation engine and automated evaluators (code verification, calibration, referee) as the selection function.

2.2 AI-Assisted Scientific Discovery

The use of AI systems to accelerate scientific discovery has advanced dramatically in recent years. Jumper et al. (2021) demonstrate that deep learning can solve the protein structure prediction problem, achieving atomic-level accuracy on a challenge that had resisted solution for over 50 years. AlphaFold shows that AI can make genuine scientific contributions in domains requiring complex pattern recognition.

More recently, LLM-based (or LLM-component) systems produce *new* mathematical results or formally verified proofs when paired with evaluation. Romera-Paredes et al. (2024) pair a pretrained LLM with an automated evaluator in an evolutionary loop and surpass prior best-known results in combinatorial problems (e.g., new cap set constructions). Trinh et al. (2024) present a geometry theorem-proving system that synthesizes theorems and proofs at Olympiad level without human demonstrations. Hubert et al. (2025) describe a reinforcement-learning and language-model system that finds formal proofs in Lean and achieves medal-level performance. For scientific discovery beyond mathematics, Boiko et al. (2023) present an LLM-driven agent that designs, plans, and executes complex chemical experiments using tools and automation; Luo et al. (2025) show that LLMs can surpass human experts in predicting experimental outcomes in neuroscience, supporting a role for LLMs in hypothesis generation and discovery workflows. These demonstrations instantiate the BVSR loop: LLM as variation engine, verifier or experiment as selector, archive or population as retention.

An emerging economics literature has begun to document closely related LLM use cases. On measurement and text-based inference, recent work develops and validates LLM pipelines for coding qualitative content and constructing large-scale research measures (Ludwig et al., 2025; Asirvatham et al., 2026; Hansen et al., 2023; Ottonello et al., 2024; Goetzmann et al., 2024; Galiani et al., 2026). On research production and scientific workflow, studies show how LLMs can automate parts of economic research and, when combined with external tools or multi-agent structures, execute richer economic-analysis

tasks (Korinek, 2023; Korinek, 2024; Korinek, 2025; Tranchero et al., 2024). A complementary line studies behavior, bias, and deployment: LLM-based survey instruments, adoption and use patterns, behavioral-bias diagnostics, and field decision-support interventions (Wu et al., 2025; Chatterji et al., 2025; Bini et al., 2026; Horton, 2023; Abaluck et al., 2026; Jabarian and Imas, 2025). Collectively, these papers show broad feasibility for LLM-assisted economic inquiry while also highlighting concerns about validation, bias, and context dependence.

Novy-Marx and Velikov (2026) demonstrate that LLMs can generate hundreds of complete finance papers with distinct theoretical justifications for data-mined signals, and caution that generating post-hoc theoretical justifications for data-mined patterns can industrialize hypothesizing after results are known (HARKing). HARKing is fundamentally a concern about statistical inference: presenting post-hoc hypotheses as ex-ante predictions inflates false-positive rates. Our setting is different. We start from a well-established quantitative failure of a workhorse structural model, not from a data-mined anomaly, and search for economic mechanisms that resolve it. This research design mirrors canonical theory papers that target known puzzles; for example, the equity premium puzzle motivates both Campbell and Cochrane (1999) and Bansal and Yaron (2004). Our system automates the search over candidate mechanisms while keeping the process fully transparent: all candidates are reported, not just the best.

AlphaEvolve (Novikov et al., 2025) extends this paradigm to general-purpose algorithm discovery. Using an ensemble of Gemini models with evolutionary search, AlphaEvolve discovers algorithms for matrix multiplication that improve upon Strassen's 1969 result, optimizes Google's data center scheduling (recovering 0.7% of worldwide compute resources), and accelerates AI training kernels. A parallel literature explores machine learning for hypothesis generation in social science: Ludwig and Mullainathan (2024) propose using ML algorithms to surface patterns in data that human researchers might miss, generating novel hypotheses about human behavior.

Our work differs from these predecessors in important ways. Unlike AlphaFold, which operates on well-defined physical constraints, economic theory requires modeling human behavior with inherent uncertainty about "ground truth." Unlike FunSearch, AlphaGeometry, AlphaProof, and AlphaEvolve, which optimize for formally verifiable objectives (proofs, algorithm performance), our evaluation combines quantitative fit with subjective assessments of economic plausibility. And unlike ML-based hypothesis generation, we generate complete theoretical models with analytical pricing expressions, not just statistical patterns. LLMs can generate theories with new mechanisms, mathematical formulations, and testable predictions. The evaluation pipeline applies round-trip consistency verification, parameter optimization, plausibility scoring, and simulated peer review to filter LLM proposals into calibrated economic theories.

2.3 The Price Multiplier Literature

Price multipliers quantify how much an asset's market value changes when demand shifts by one dollar. Understanding these multipliers is central to demand-based asset pricing.

Gabaix and Koijen (2023) propose the "inelastic markets hypothesis," finding that every $1 invested in the aggregate market increases market value by approximately $5, far larger than standard models predict. Using granular instrumental variables, they document price multipliers in the range of 3 to 8 for market-level flows, while individual stock multipliers are closer to 1. This aggregate-versus-individual gap suggests that diversifiable risk is priced differently than systematic risk.

Li and Lin (2022) provide the direct motivation for our puzzle. They decompose demand into style-level and idiosyncratic components and estimate separate price multipliers for each. They find that price multipliers form a continuum: approximately 1.6 at the idiosyncratic level, 2.5 at the granular style level, and 5.9 at the coarse style level. They show that the standard CARA-normal model *qualitatively* predicts this ordering

(higher multipliers for less diversifiable risk) but *quantitatively* fails: the model predicts $M_{\text{style}}/M_{\text{idio}} \approx 14.5$, while the data show ≈ 3.6.

This quantitative gap, a factor of four, is our target puzzle. In a subsequent revision (January 2026, posted after the release of the LLM used in our experiments), Li and Lin (2022) add a back-of-the-envelope calibration showing that when only a fraction ($\approx 29\%$) of investors actively provide liquidity, risk aversion alone can generate multipliers of the correct order of magnitude. However, treating their point estimates literally "would be heroic," and they do not provide a fully specified equilibrium model. The quantitative discrepancy thus remains an open target for theory.

Related work on market microstructure provides foundations for understanding price impact. Kyle (1985) derives equilibrium price impact from informed trading, while Glosten and Milgrom (1985) model bid-ask spreads with heterogeneous information. Hasbrouck (2009) develops methods for estimating effective trading costs from daily data. These papers focus primarily on information-driven price impact; our puzzle concerns risk-based price impact where the Li and Lin (2022) evidence suggests information plays a limited role.

The contribution to this literature is methodological rather than empirical. LLM-based evolutionary search can generate multiple distinct theoretical mechanisms that quantitatively match the empirical puzzle. These mechanisms, including attention-cost models, dual-agent limited-diversification frameworks, and Bayesian learning with portfolio-breadth costs, match empirical moments and admit testable predictions. Multiple mechanisms can fit the same data; the system generates candidate explanations at a scale that would be infeasible for human researchers alone.

3 A Model of Theory Discovery

This section formalizes theory discovery as a stochastic search problem under verification constraints, characterizing when large-scale LLM generation should outperform conventional sequential theorizing and how to organize search when generation is cheap but evaluation is not.

3.1 Setup

Let $\mathcal{T}$ denote the finite but large space of candidate theories for a puzzle specification s. A candidate $\tau \in \mathcal{T}$ can be represented as text, equations, and executable code. The pipeline maps each τ to a quality score $q(\tau) \in [0, 1]$ (normalized from a 0–100 scale) that aggregates implementation validity, empirical fit, economic plausibility, and parsimony. For a publication threshold $\bar{q}$, define the acceptable set $\mathcal{A} = \{\tau \in \mathcal{T} : q(\tau) \geq \bar{q}\}$.

Following the creativity literature, we distinguish effectiveness from originality. Let $\mathcal{K}$ be a reference corpus of known mechanisms. The set of creative theories is

$$\mathcal{C} = \{\tau \in \mathcal{T} : \tau \text{ is novel relative to } \mathcal{K} \text{ and } q(\tau) \geq \bar{q}\}. \tag{1}$$

Novelty here encompasses genuinely new mechanisms or novel combinations of known building blocks, consistent with the recombinant view of creativity in Fleming (2001). This operationalizes the originality-and-effectiveness criterion in Runco and Jaeger (2012) within a fully auditable evaluation pipeline.

3.2 Generator-selector structure

Conditional on s, the LLM induces a distribution $P_L(\cdot \mid s)$ over $\mathcal{T}$. One pipeline run draws a candidate $\tau \sim P_L(\cdot \mid s)$ and then subjects it to deterministic and model-based checks that produce $q(\tau)$. In the language of Campbell (1960), the pipeline is a generate–

test–retain process: generation is stochastic variation, scoring is selection, and the model registry is retention.

Let

$$p_L^A = \Pr_{\tau \sim P_L(\cdot|s)}(\tau \in \mathcal{A}) \tag{2}$$

denote the probability that one draw lands in the acceptable set, and let

$$p_L^C = \Pr_{\tau \sim P_L(\cdot|s)}(\tau \in \mathcal{C}) \tag{3}$$

denote the probability of a creative hit. If draws were independent, the probability of at least one creative hit after n attempts would be

$$1 - (1 - p_L^C)^n, \tag{4}$$

which increases monotonically in n and converges to one as $n \to \infty$ whenever $p_L^C > 0$. Under the same benchmark, the expected number of creative hits is np_L^C. More generally, regardless of the dependence structure, the expected best quality $\mathbb{E}[\max_{1 \le i \le n} q(\tau_i)]$ is weakly increasing in n.

In practice, draws from a single LLM prompt are not independent: the model's training distribution, prompt structure, and decoding temperature induce positive correlation across candidates, so the effective sample size $n_{\text{eff}} < n$. We address this directly through a persona mechanism. Let $\Pi = \{\pi_1, \ldots, \pi_K\}$ be a set of K expert persona configurations. We define 20 base personas (methodological perspectives such as "bayesian," "physicist," "minimalist"); each run either uses a default perspective (with probability 0.2) or randomly samples a combination of one to three base personas (with probability 0.8), yielding $K = \binom{20}{1} + \binom{20}{2} + \binom{20}{3} + 1 = 1{,}351$ distinct configurations. Conditioning the generator on configuration π_k shifts its distribution to $P_L(\cdot \mid s, \pi_k)$, which concentrates on a

different region of the theory space. The unconditional generator is then a mixture

$$P_L(\cdot \mid s) = \sum_{k=1}^{K} w_k \, P_L(\cdot \mid s, \pi_k), \tag{5}$$

where w_k is the probability of assigning configuration π_k. When the component distributions have limited overlap (that is, when different configurations tend to propose qualitatively different mechanisms), draws from different components are approximately independent even if draws within a component are correlated. The effective sample size then scales with the number of distinct components explored, not just the raw number of draws. This observation is the formal motivation for our persona system: it increases n_{eff} by ensuring that the generator's support spans multiple modes of the theory space.

The independence benchmark should therefore be read as an upper bound that becomes approximately tight when the persona mechanism and evolutionary operators are effective at reducing inter-draw correlation. Increasing throughput and diversity raises the likelihood of discovering a high-quality or creative theory when such theories have non-zero probability mass under the generator. This scaling property is the same statistical logic behind the empirical regularity that high-impact discoveries scale with attempts in stochastic models of creativity (Simonton, 1997).

Why scale matters in symbolic search. Three structural features make this scaling argument relevant for theory generation. First, outcomes in symbolic domains are heavy tailed: a small share of candidates accounts for most value, so best-of-n sampling has large returns. Second, the search space is combinatorial: even a limited library of primitives can be recombined into a very large set of coherent mechanisms. Third, sampling controls provide a direct exploration-exploitation margin, allowing one to trade off local refinement against novelty at low marginal cost. Taken together, these properties imply that discovery rates can rise sharply with throughput even when per-draw quality is modest.

3.3 Economic comparison with human-only search

Suppose a human researcher draws from P_H at expected cost c_H per attempt and success probability

$$p_H^{\mathcal{A}} = \Pr_{\tau \sim P_H}(\tau \in \mathcal{A}). \tag{6}$$

The expected cost per acceptable theory is $c_H / p_H^{\mathcal{A}}$. For an LLM system with per-attempt cost c_L and acceptable-hit probability $p_L^{\mathcal{A}}$, the analogous object is $c_L / p_L^{\mathcal{A}}$. LLM search has a cost advantage whenever

$$\frac{c_L}{p_L^{\mathcal{A}}} < \frac{c_H}{p_H^{\mathcal{A}}}. \tag{7}$$

This condition can hold even if $p_L^{\mathcal{A}} < p_H^{\mathcal{A}}$, because c_L is orders of magnitude smaller and generation can be parallelized. An analogous comparison can be made for creative hits by replacing $\mathcal{A}$ with $\mathcal{C}$ for both humans and LLMs; the qualitative point is the same, but acceptable theories are the more directly observable object in our pipeline. In Section 5, we proxy novelty using cosine distance between proposal embeddings.

3.4 Parallel generation as filtered discovery

With parallel compute, discovery shifts from sequential search to filtered discovery. Instead of producing one theory at a time, the system generates a batch $\{\tau_1, \ldots, \tau_N\}$, evaluates all candidates, and retains those above threshold. Under the independent-draw benchmark, expected accepted output in a batch is $N p_L^{\mathcal{A}}$. If generation is fully parallel and evaluation is partly parallel, wall-clock time is driven mainly by the slowest evaluation stage rather than by the number of generated candidates.

This yields a simple parallel-first principle. When generation cost is low relative to evaluation cost, and generation is more parallelizable than adaptive refinement, it is typically advantageous to allocate substantial budget to broad independent exploration before concentrating on local search. Independent draws expand the support explored per unit wall-clock time, whereas sequential refinement conditions each new proposal on

previous outcomes and consumes serial decision time. If evaluation is the bottleneck, the opportunity cost of serial adaptation is high. This principle does not imply that mutation or refinement is useless. It implies that adaptation should be layered on top of broad parallel exploration, rather than replacing it.

3.5 Hybrid policy and testable implications

A practical policy has two stages. The first stage allocates substantial mass to independent random draws to map the space and identify multiple promising basins. The second stage applies directed operators (mutation, crossover) to the best candidates, while preserving some ongoing random exploration to prevent premature concentration. If p_r is the success rate of random generation and p_m the conditional success rate of mutation around strong parents, a simple decomposition is

$$\mathbb{E}[\text{successful random draws}] = N_r p_r, \tag{8}$$

$$\mathbb{E}[\text{successful directed draws}] = N_m p_m. \tag{9}$$

The optimal split between N_r and N_m depends on the empirical gap between p_m and p_r, the diversity of discoveries generated by each operator, and evaluation capacity.

This framework yields two empirical predictions that are directly testable in our data. First, if the theory landscape is strongly multi-modal, random draws should contribute disproportionately to top discoveries relative to local mutation. Second, crossover should improve average quality when it recombines complementary mechanisms from already strong parents. Section 5 reports evidence consistent with both patterns, and therefore supports a hybrid strategy in which parallel exploration remains central throughout the run.

The following sections describe the pipeline that implements this logic: how candidate theories are generated and translated into code, how scores are computed, and how

strategy selection is applied across a large number of runs.

4 System Design

Our system consists of ten stages executed in sequence, organized into three phases: *generation* (Stages 0–1), *verification* (Stages 2–4), and *evaluation* (Stages 5–9). Each stage saves its output incrementally, so partial results are preserved even if later stages fail. The system is puzzle-agnostic: each puzzle defines its own target outputs, calibration inputs, and prompts through a modular interface, allowing the same pipeline to be applied to different empirical puzzles without modification.

4.1 Generation Phase

Stage 0: Problem Analysis with Expert Personas. Given a puzzle definition, an LLM analyzes the problem and identifies the empirical facts that need explaining. The *problem analyzer*, an LLM equipped with the problem description and baseline model, describes the empirical facts, explains why the baseline model fails, and highlights both promising directions for new theoretical explanations and likely dead ends. The *problem analyzer* also determines the constraints that any successful theory must satisfy and highlights one unconventional angle worth exploring.

To promote diversity in the generated theories, each run randomly assigns an *expert persona* to the problem analyzer. We define 20 personas spanning different methodological perspectives: *mathematician* (fixed-point theorems, duality), *bayesian* (priors, posteriors, value of information), *behavioral* (bounded rationality, heuristics), *physicist* (conservation laws, symmetries), *minimalist* (fewest assumptions possible), *contrarian* (challenges conventional wisdom), and others.[1] With probability 0.2, the run uses a default perspec-

[1]The full list of personas includes: default, mathematician, game theorist, information theorist, decision theorist, bayesian, econometrician, robust statistician, macro theorist, micro theorist, financial economist, behavioral, institutional, monetary economist, network theorist, dynamic programmer, minimalist, con-

tive; with probability 0.8, it randomly samples one to three personas and combines their perspectives by including all selected persona descriptions in the LLM's system prompt simultaneously. This mechanism ensures that the system explores qualitatively different theoretical directions across runs, reducing the tendency of LLMs to converge on the same class of solutions. See Appendix A.1 for the prompts used in the problem analyzer.

Stage 1: Theory Evolution. The *theory evolution* stage receives the problem description and baseline model from the problem analyzer. Inspired by the output of the problem analyzer, it proposes a new theoretical mechanism that potentially explains the empirical facts.

For *theory evolution*, we implement three strategies. The first is *Random*, which generates a new theory from scratch. Many scientific discoveries evolve from combinations of previous theories (Uzzi et al., 2013), so we add two strategies that leverage existing theories. Once successful theories exist, the *theory evolution* LLM uses them as parents to generate new theories. One option is *Mutate*, which modifies an existing successful theory. The mutate prompt includes the parent's formulas, parameters, calibration results, and the full referee report (strengths, weaknesses, suggestions), directing the LLM to address at least one identified weakness. Examples of mutation include simplifying the parent model, changing parameter mappings, or adding interactions and new effects.

The third strategy is *Crossover*, which combines mechanisms from two successful theories and creates a hybrid model using the best elements of both. The crossover prompt directs the LLM to aim for fewer total parameters than the sum of the two parents by finding overlap and eliminating redundancy.

Once at least one model has completed the full pipeline, strategies are selected probabilistically: 35% *random*, 35% *mutate*, 30% *crossover*. The system draws parents from models with referee score ≥ 35, with selection probability proportional to score (linearly shifted so that the lowest-ranked model in the top 10 receives a small but positive weight),

trarian, physicist, and pragmatist.

balancing exploitation of the best theories with exploration of lower-ranked alternatives.

The output of the *theory evolution* stage is a new theoretical mechanism that potentially explains the puzzle. See Appendix A.2 for the prompts used in theory evolution.

4.2 Verification Phase

Stage 2: Mathematical Audit. Before translating a proposal into code, we subject it to an independent mathematical review. A separate LLM call, operating at low temperature (0.1), audits every derivation step in the proposal. The auditor checks that each equation follows rigorously from the previous one, that matrix algebra and first-order conditions are correct, and that market-clearing conditions hold. The auditor returns a PASS or FAIL verdict with detailed feedback.

If the audit fails, the system sends the original proposal together with the auditor's feedback to another LLM call that attempts to fix the mathematical errors while preserving the economic mechanism. This fix-and-reaudit cycle repeats up to two times. Proposals that fail all audit attempts are discarded. This stage catches a substantial fraction of algebraic errors that the generation LLM introduces, errors that would otherwise propagate into incorrect code and misleading calibration results.

Stage 3: Pseudocode Generation and Review. Rather than translating proposals directly into Python, we introduce an intermediate pseudocode stage that forces the LLM to think through the complete derivation chain before coding. The *pseudocode agent* LLM converts the audited proposal into structured pseudocode organized into four sections: parameters (with types and bounds), derivation steps (from economic primitives to pricing equations), output computation (mapping derived quantities to target moments), and conditional moments (if the puzzle requires state-dependent predictions).

A separate *pseudocode reviewer* LLM then verifies that the pseudocode faithfully captures the proposal's economic mechanism, that all key equations are present, and that outputs are derived from the model rather than hardcoded. If the reviewer identifies

issues, the pseudocode agent regenerates with the reviewer's feedback. This review-and-retry cycle runs up to two times.

Stage 4: Code Translation and Review. The *coding agent* LLM translates the pseudocode into executable Python. It receives both the pseudocode specification and the original proposal, and produces a self-contained function that takes model parameters as input and returns computed target outputs. The coding agent uses a nested retry strategy: up to two fresh starts, each with up to three retries, for a maximum of six attempts. A fresh start discards all prior context and calls the LLM from scratch (with slightly higher temperature); a retry within the same fresh start feeds the previous error message back to the LLM so it can correct the specific failure.

After code generation, a *code-vs-pseudocode reviewer* LLM verifies that the Python code correctly implements the pseudocode specification. It checks that parameters, derivation equations, output formulas, and conditional logic match. If the review fails, the coding agent regenerates with the reviewer's feedback.

This two-stage translation (proposal to pseudocode to code, with review at each step) structures the translation into smaller, verifiable steps. See Appendix A.3 for the prompts used in code translation.

4.3 Evaluation Phase

Stage 5: Calibration Agent. Once the *coding agent* generates Python code, the *calibration agent* determines parameter specifications for the subsequent optimization stage. It takes the model proposal and the generated code as input and suggests parameter bounds for optimization, default values, test values for robustness checking, and parameter types. This separation keeps calibration concerns distinct from code generation. See Appendix A.4 for the prompts used in the calibration agent.

Stage 6: Parameter Optimization. The *optimizer* is a Python component (not an LLM) that: (i) executes the generated code and runs a *robustness check* at multiple parameter

values (supplied by the calibration agent) to ensure the model runs without errors or invalid outputs; (ii) uses *differential evolution* to minimize the sum of relative squared errors $L(x) = \sum_i \left((y_i(x) - t_i)/t_i \right)^2$, where x is the parameter vector, $y_i(x)$ the model's i-th output, and t_i the corresponding empirical target; and (iii) reports whether the model achieves the targets and the resulting outputs for use in scoring and the referee. See Appendix A.9 for the objective function and optimization details.

Stage 7: Quantitative Scoring. After the optimizer returns optimized parameters and model outputs, the *scorer* computes a quantitative score (0–100) for each model. The scorer is a Python component that combines three dimensions with configurable weights: *fit* (50%) measures how closely the model's outputs match the empirical targets, using percentage errors and an exponential decay; *plausibility* (25%) checks whether the optimized parameters lie within economically reasonable bounds (e.g., risk aversion γ in a standard range, fractions in $[0, 1]$) and penalizes out-of-bounds values; *parsimony* (25%) rewards fewer free parameters. The weighted total is passed to the referee and used to rank models and select parents for mutation and crossover. See Appendix A.10 for the exact formulas.

Stage 8: Simulated Peer Review (Referee). The *referee* is an LLM that acts as a senior journal referee. It receives the model proposal, the quantitative scores from the scorer, and the optimization results. It produces a single score (0–100), a recommendation (reject, major revision, minor revision, or accept), and structured feedback: strengths, weaknesses, suggestions, novelty assessment, and economic plausibility. This feedback is stored in the model registry and used to guide the *mutate* strategy in later theory evolution, so that the system can address specific weaknesses identified by the referee. See Appendix A.11 for the prompts used in the referee.

Stage 9: Auto-Save. The system saves successful models (those that pass optimization and receive a referee score) to a persistent registry along with all metadata: the proposal, code, pseudocode, calibration bounds, optimized parameters, model outputs, quantita-

tive scores, referee feedback, strategy used, parent model(s), generation number, and persona. This registry serves as the population from which future runs select parents for mutation and crossover.

5 Experimental Analysis

We apply the system to two empirical puzzles in asset pricing. The first is the price-multiplier scaling puzzle, where our system discovers multiple distinct top-tier mechanisms across 1,000 pipeline runs (458 completing all stages). The second is the equity term structure (dividend strip) puzzle, a harder target with ten moment conditions, where 2,153 pipeline runs produce 257 scored theories. Together, these experiments illustrate the system's capabilities and the factors that determine puzzle tractability for LLM-based search.

5.1 Setup

All LLM stages use `gpt-oss-120b` (OpenAI et al., 2025), OpenAI's open-weight reasoning model released under the Apache 2.0 license. The model is a 120B-parameter mixture-of-experts reasoning model, trained with large-scale distillation and reinforcement learning; it achieves near-parity with OpenAI's o4-mini on standard reasoning benchmarks. We access it through a university API proxy, with a training and knowledge cutoff of June 1, 2024; temperature and task design vary by stage. Table 1 gives the parameters for each pipeline stage. Theory-facing stages (problem analysis, evolution, referee) use higher temperatures (0.5–0.7) to encourage diverse proposals; verification and code stages use lower temperatures (0.1–0.2) for consistency. The coding agent uses temperature 0.2 on the first attempt and slightly higher on fresh retries (0.2 + 0.1 per fresh start). The optimizer and scorer are deterministic Python components.

The quantitative score for each model is $0.50 \times \text{fit} + 0.25 \times \text{plausibility} + 0.25 \times \text{parsimony}$,

Table 1: LLM and optimization parameters by pipeline stage

Stage	Component	Model	Temperature
0	Problem analyzer (with personas)	120B	0.7
1	Theory evolution (random / mutate / crossover)	120B	0.7
1	Theory evolution (mutate / crossover, with referee feedback)	120B	0.5
2	Mathematical auditor	120B	0.1
3	Pseudocode agent	120B	0.2
3	Pseudocode reviewer	120B	0.2
4	Coding agent	120B	0.2
4	Code-vs-pseudocode reviewer	120B	0.2
5	Calibration agent	120B	0.2
6	Parameter optimizer	(Python)	—
7	Quantitative scorer	(Python)	—
8	Referee	120B	0.3

Notes: Model refers to the autoregressive language model used for that stage: "120B" denotes `gpt-oss-120b` (120 billion parameters), called via the API with a training and knowledge cutoff of June 1, 2024. Temperature is the LLM sampling temperature (0 = deterministic; higher values yield more diverse outputs). Stages 6 (Optimizer) and 7 (Scorer) are deterministic Python components, so model and temperature are not applicable.

with each component on a 0–100 scale. See Appendix A.10 for the formulas for fit, plausibility, and parsimony.

5.2 Puzzle 1: Price Multiplier Scaling

5.2.1 The puzzle and calibration targets

Definition of the puzzle. The puzzle we target is the *scaling of price multipliers across aggregation levels*: how much an asset's market value changes per dollar of demand shift depends on whether that demand is decomposed as style-level (common to many stocks) or idiosyncratic (stock-specific). The price multiplier M is the elasticity of market value to demand; higher M means prices respond more to a given flow. Li and Lin (2022) estimate these multipliers from U.S. stock data by decomposing order flow into coarse style (3×3 size–book-to-market portfolios), granular style (6×6), and idiosyncratic components. They find that multipliers form a monotonic gradient: coarse style has the

largest price impact, granular style is in between, and idiosyncratic is the smallest. Their baseline estimates are $M_{\text{style}} \approx 5.917$ (coarse style), $M_{6\times6} \approx 2.525$ (granular style), and $M_{\text{idio}} \approx 1.635$ (idiosyncratic). The empirical ratio is

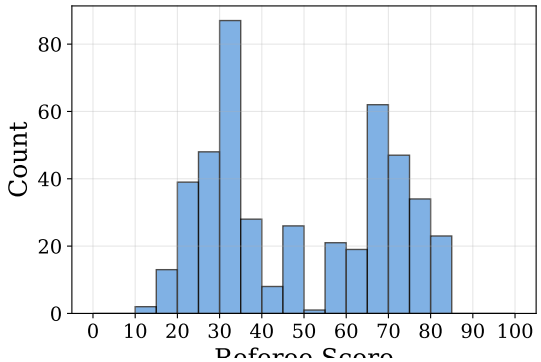
$$\left(\frac{M_{\text{style}}}{M_{\text{idio}}}\right)_{\text{data}} = \frac{5.917}{1.635} \approx 3.62. \tag{10}$$

A natural risk-sharing benchmark implies a much larger ratio. If the marginal investor has CARA–normal preferences and absorbs flow-induced inventory, the required risk premium scales with the variance of the position. A style-level flow shock hits all N_s stocks in a style in the same direction, so the variance of the absorbed position scales with $N_s\sigma^2_{\text{style}}$; an idiosyncratic shock affects one stock, so the relevant variance is σ^2_{idio}. Under this aggregation, the benchmark predicts (see Appendix B for the derivation)

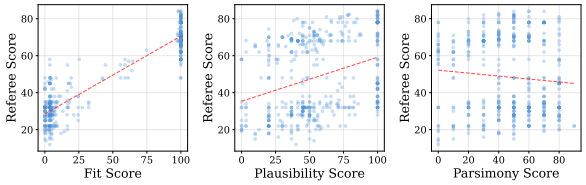
$$\frac{M_{\text{style}}}{M_{\text{idio}}} = \frac{N_s\sigma^2_{\text{style}}}{\sigma^2_{\text{idio}}}. \tag{11}$$

With plausible calibration inputs ($N_s = 253$, $\sigma_{\text{style}} = 0.104$, $\sigma_{\text{idio}} = 0.435$), this ratio equals approximately 14.5. The *puzzle* is that the data show the correct qualitative ordering (style $>$ idiosyncratic) but a far flatter scaling: the empirical ratio is about 3.6, while the benchmark predicts about 14.5, a factor-of-four gap.

Calibration and targets used in the experiment. Following Li and Lin (2022), we use $N = 2{,}278$ total stocks, $S = 9$ styles, and $N_s = 253$ stocks per style. Style volatility is $\sigma_{\text{style}} = 0.104$ (annual) and idiosyncratic volatility is $\sigma_{\text{idio}} = 0.435$ (annual). The empirical targets are $M_{\text{style}} = 5.917$, $M_{\text{idio}} = 1.635$, and the target ratio $M_{\text{style}}/M_{\text{idio}} = 3.62$. The optimizer minimizes the sum of squared relative errors across all three outputs (equal weights); see Appendix A.9. We define a run as *successful* if and only if (i) the robustness check passes, (ii) the verification stages certify that the code implements the proposal, and (iii) the post-optimization average percentage error $\bar{e}$ is at most 10%.



(a) Distribution of referee scores across all models.



(b) Fit, plausibility, and parsimony vs. referee score.

Figure 1: Score distributions of generated models (price multiplier puzzle).

Notes: Panel (a) reports the distribution of final LLM referee scores on a 0–100 scale. Panel (b) plots each quantitative component score (fit, plausibility, parsimony) against the referee score, with linear trend lines. The best referee score is 84, achieved by four models.

5.2.2 Experiment design

We execute 1,000 pipeline invocations using up to 128 concurrent workers, of which 458 complete all stages and enter the model registry. The remainder fail at the math audit (41%), optimizer (9%), code generation (3%), or time out (1%); see Table 12. Once at least one model completes the pipeline, each run uses the mixed strategy: random 35%, mutate 35%, crossover 30%. Parents for mutate and crossover are drawn from models with referee score ≥ 35, selected from the top 10 with probability proportional to score. See Appendix C.2 for the exact procedure.

5.2.3 Score distributions

Four models with distinct mechanisms reach the top referee score of 84. Figure 1 shows the distributions of referee scores and quantitative component scores.

Table 2 reports summary statistics. The mean referee score is 48.5 with a standard deviation of 20.9; 185 models (40%) score at least 60, and 23 models (5%) score at least 80. The evolutionary search reaches generation 14, indicating that mutation and crossover build on earlier discoveries.

Table 2: Summary Statistics (Price Multiplier)

Statistic	Value
Total models	458
Mean score	48.5
Median score	45.0
Std. dev.	20.9
Best score	84
Models ≥ 60	185
Models ≥ 80	23
Max generation	14

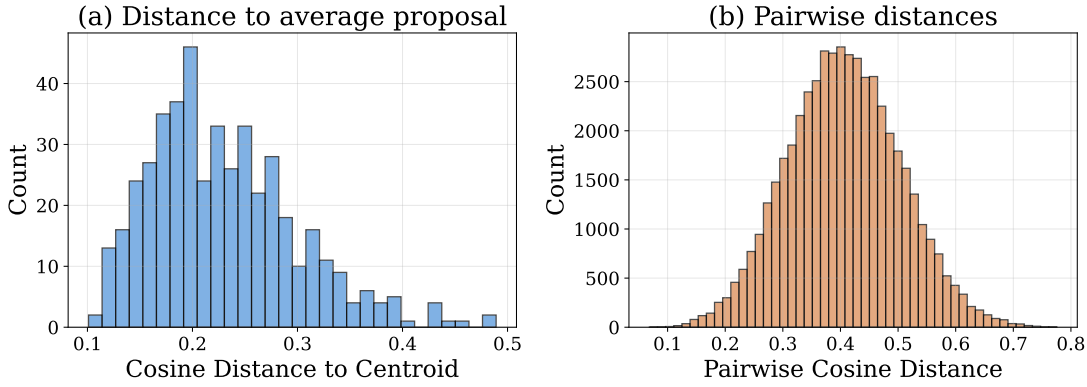
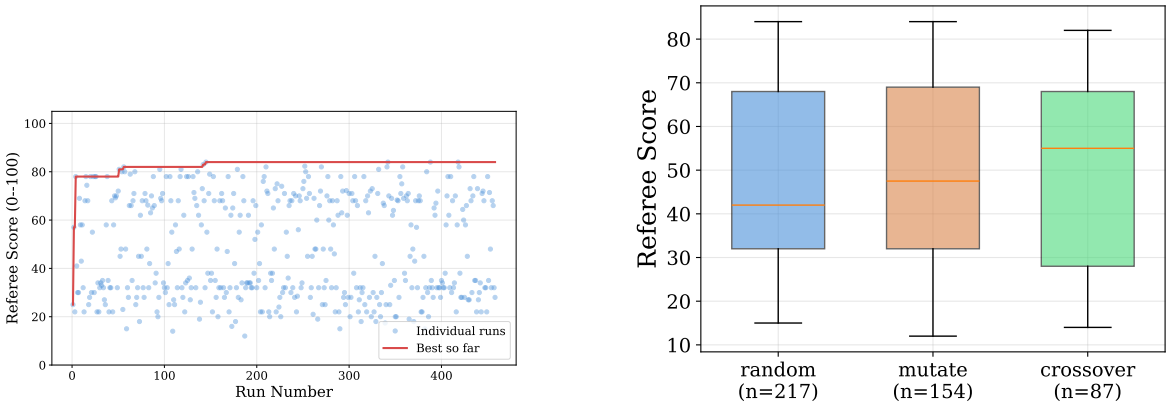


Figure 2: Variation in generated proposals (price multiplier puzzle).

Notes: Panel (a) shows cosine distance between each proposal embedding and the average proposal embedding (using all-MiniLM-L6-v2 sentence embeddings); larger values indicate proposals that are farther from the center of the generated set. Panel (b) shows pairwise cosine distances across random proposal pairs; larger values indicate stronger dispersion in the explored idea space.

5.2.4 Heterogeneity of generated proposals

To assess whether the system explores diverse theoretical directions rather than producing minor variants of the same idea, we embed all 458 proposals using a sentence transformer (all-MiniLM-L6-v2) and compute cosine distances. Figure 2 shows the results. The mean cosine distance from each proposal to the centroid is 0.23, and the mean pairwise cosine distance is 0.41 (the absolute scale depends on the embedding model, but the pairwise distances indicate that proposals differ meaningfully in content rather than clustering around a single mechanism). The combination of persona diversity and evolutionary operators generates proposals that are semantically distinct, not just parameter



(a) Score evolution.

(b) Score by strategy.

Figure 3: Score evolution across 458 runs (price multiplier puzzle).

Notes: Panel (a) plots referee scores by run index, with the running maximum shown as a red line. Panel (b) reports score distributions by strategy using box plots. The figure visualizes the search dynamics over the full experiment horizon.

variations.

5.2.5 Score evolution

Figure 3 shows the score evolution over the experiment.

Table 3 reports strategy-level performance. Mutate achieves the highest mean referee score (49.6), while random generation produces models across the widest score range and ties for the best score (84). Crossover has a higher median (55.0) than either random or mutate, suggesting it tends to produce above-average models by combining successful ideas. However, crossover has fewer observations ($n = 87$ vs. 217 for random and 154 for mutate) and by construction requires two parents with referee scores ≥ 35, so its higher median partly reflects starting from stronger raw material.

5.2.6 Correlates of referee scores

To understand what drives the LLM referee's evaluation, we regress $\log(1 + \text{referee score})$ on a set of observable features in four specifications (Table 4). Columns (1) and (2) exclude the quantitative fit score; columns (3) and (4) include it. Columns (2) and (4) add dummies

Table 3: Strategy Performance Comparison (Price Multiplier)

Strategy	n	Mean	Median	Max
Random	217	48.0	42.0	84
Mutate	154	49.6	47.5	84
Crossover	87	47.7	55.0	82
All	458	48.5	45.0	84

for the evolutionary strategy (mutate, random; crossover is the omitted category). All specifications control for the length of the proposal text and the lengths of the referee's written feedback fields (strengths, weaknesses, suggestions). Lengths enter as $\log(1 + \text{length})$ to allow for diminishing effects and to reduce the influence of outliers.

The quantitative fit score is the mechanical score computed by the pipeline before the referee stage. When we include this score in columns (3) and (4), it is by far the strongest predictor: the R^2 jumps from about 0.09 in specifications (1)–(2) to about 0.89 in (3)–(4). The referee's final assessment is largely aligned with the objective fit-plausibility-parsimony composite. Among the feedback lengths, the "strengths" section is consistently positive for the referee score (significant at 1% in columns (1)–(2)), while "weaknesses" length correlates negatively. Strategy dummies add negligible explanatory power.

Table 4: Determinants of referee scores (Price Multiplier)

	(1)	(2)	(3)	(4)
Intercept	4.139***	4.118***	2.990***	2.978***
	(0.196)	(0.201)	(0.084)	(0.084)
Quantitative fit score			0.018***	0.018***
			(0.0004)	(0.0004)
Proposal length	-0.175	-0.186	-0.019	-0.019
	(0.183)	(0.184)	(0.064)	(0.064)
Strengths length	0.751***	0.764***	0.043	0.036
	(0.123)	(0.124)	(0.046)	(0.047)
Weaknesses length	-0.412***	-0.409***	-0.010	-0.005
	(0.139)	(0.140)	(0.056)	(0.056)
Suggestions length	-0.174	-0.180	-0.020	-0.018
	(0.154)	(0.155)	(0.054)	(0.055)
Mutate (strategy)		0.026		0.019
		(0.064)		(0.022)
Random (strategy)		0.049		0.002
		(0.061)		(0.022)
Strategy dummies	No	Yes	No	Yes
N	458	458	458	458
R^2	0.086	0.087	0.888	0.889

Notes: OLS regressions of $\log(1 + \text{referee score})$ on text length features (log character counts of proposal, referee strengths, weaknesses, and suggestions), quantitative fit score, and (where indicated) strategy dummies. Heteroskedasticity-robust (HC1) standard errors are in parentheses. Significance levels are $^{***}p < 0.01$, $^{**}p < 0.05$, and $^{*}p < 0.10$.

5.2.7 Top-Scoring Mechanisms

Table 5 lists the ten highest-scoring models. Four models tie at the top score of 84: the Bayesian-Breadth-Cost CARA, the Dual-Agent Limited-Diversification with Unified Risk-Aversion (DALD-UR), the Liquidity-Adjusted CRRA, and the Constrained-CRRA No-Short Idiosyncratic model. Complete equilibrium derivations for two of the four top-scoring models follow.

1. Bayesian-Breadth-Cost CARA Model (BBC).

Table 5: Top 10 Discovered Models (Price Multiplier)

Rank	Model	Strategy	Key Insight	Score
1	Liquidity-Adjusted CRRA (LCRRA) Model	random	The incorporation of a liquidity-adjusted marginal utility that links the pricing kernel to idiosyncratic order flow is a novel twist on the standard CRRA framework, distinguishing the paper from prio…	84
2	DALD-UR (Unified-Risk-Aversion) Model	mutate	The paper offers a novel closed-form solution for price multipliers in a dual-agent CARA framework with a breadth constraint, which has not been presented in the literature before.	84
3	Constrained-CRRA No-Short Idiosyncratic Model	random	The idea of generating an idiosyncratic risk price via a binding long-only constraint is not entirely new, but the paper's clean two-group CRRA formulation and the explicit closed-form link between th…	84
4	Bayesian-Breadth-Cost CARA Model	random	The model combines existing ideas—Bayesian learning about a common factor and quadratic portfolio-breadth costs—into a coherent closed-form solution. While the algebraic correction is valuable, the core mechanisms are not new, so the contribution is incremental rather than groundbreaking.	84
5	Attention-Cost Integrated CARA-Normal Model	mutate	The incorporation of a quadratic limited-attention cost into a CARA-normal pricing model is a novel twist that directly links a behavioural friction to the idiosyncratic risk premium, distinguishing i…	83
6	DALD-U (Unified-Risk-Aversion Dual-Agent Limited-Diversification)	mutate	The incorporation of a hard-cap breadth constraint into an exact exponential-SDF framework is a novel extension of the standard CARA-normal literature and addresses a known empirical puzzle.	82
7	Network-Liquidity Low-Rank Factor Model	random	The introduction of a market-wide common shock together with a participation constraint to generate a non-zero idiosyncratic multiplier is a fresh angle, though related ideas appear in the limited-par…	82
8	Participatory-Attention CARA-Normal Model	crossover	The combination of a quadratic attention cost with a participation threshold that creates a mixture SDF is a fresh angle on the limited-attention literature, though related ideas have appeared in rece…	82
9	Binary-Attention CARA-Normal Model	mutate	The mixture SDF generated by a binary limited-attention rule is a novel contribution to the CARA-Normal literature and has not been presented in prior work.	82
10	Unspanned Stochastic-Volatility CRRA Model	random	Introducing an unspanned stochastic-volatility factor into a CRRA consumption-based framework is a novel twist on the standard one-factor pricing kernel and yields new closed-form multiplier expressio…	82

Notes: Top 10 models by referee score from the price multiplier experiment. "Strategy" indicates whether the model was generated from scratch (random), by modifying a parent (mutate), or by combining two parents (crossover). "Key Insight" summarizes the referee's novelty assessment.

Setup. A continuum of identical investors have CARA utility $U(W) = -\exp(-\gamma W)$. There are N assets with returns

$$R_i = \alpha_i S + \varepsilon_i, \quad i = 1, \ldots, N, \tag{12}$$

where $S \sim N(0, \sigma^2_{\text{style}})$ is a common style factor, $\varepsilon_i \overset{\text{i.i.d.}}{\sim} N(0, \sigma^2_{\text{idio}})$ are idiosyncratic shocks, and $\alpha_i = 1$ for N_s style stocks and $\alpha_i = 0$ otherwise. Two frictions modify the standard model:

1. *Noisy public signal*: investors observe $s = S + \eta$, where $\eta \sim N(0, \sigma^2_\eta)$.
2. *Quadratic holding cost*: a cost $\lambda \sum_{i=1}^N \theta_i^2$ penalizes portfolio breadth.

Bayesian updating. The posterior variance of the style factor given the signal is

$$\sigma^2_{S|s} = \left(\frac{1}{\sigma^2_{\text{style}}} + \frac{1}{\sigma^2_\eta} \right)^{-1} \equiv \phi\, \sigma^2_{\text{style}}, \quad \phi \in (0, 1). \tag{13}$$

The information-premium parameter is $\psi = (1 - \phi)/\phi \geq 0$, and the residual idiosyncratic-variance fraction under the holding cost is $\kappa = 2\lambda/(\gamma \sigma^2_{\text{idio}} + 2\lambda) \in (0, 1)$.

Investor optimization. Each investor solves

$$\max_{\boldsymbol{\theta}} \left\{ \boldsymbol{\theta}'(\boldsymbol{\mu} - R_f \mathbf{1}) - \frac{\gamma}{2} \boldsymbol{\theta}' \Sigma \boldsymbol{\theta} - \lambda \boldsymbol{\theta}' \boldsymbol{\theta} \right\}, \tag{14}$$

where $\Sigma = \sigma^2_{\text{idio}} I_N + \sigma^2_{\text{style}} \boldsymbol{\alpha}\boldsymbol{\alpha}'$ is the return covariance matrix. The first-order condition is

$$(\gamma \Sigma + 2\lambda I_N) \boldsymbol{\theta} = \boldsymbol{\mu} - R_f \mathbf{1}. \tag{15}$$

Optimal demand via Sherman–Morrison. Since Σ contains the rank-one component $\sigma^2_{\text{style}} \boldsymbol{\alpha}\boldsymbol{\alpha}'$, the Sherman–Morrison formula yields closed-form demands. For a style stock

($\alpha_i = 1$):

$$\theta_i^{\text{style}} = \frac{\mu_i - R_f}{\gamma\sigma^2_{\text{idio}} + 2\lambda} - \frac{\gamma\sigma^2_{\text{style}}}{(\gamma\sigma^2_{\text{idio}} + 2\lambda)(\gamma\sigma^2_{\text{idio}} + 2\lambda + \gamma\sigma^2_{\text{style}} N_s)} \sum_{j=1}^{N_s} (\mu_j - R_f), \tag{16}$$

and for an idiosyncratic stock ($\alpha_i = 0$):

$$\theta_i^{\text{idio}} = \frac{\mu_i - R_f}{\gamma\sigma^2_{\text{idio}} + 2\lambda}. \tag{17}$$

Stochastic discount factor. The CARA SDF is $m \propto \exp(-\gamma W^*)$, where $W^* = R_f + \boldsymbol{\theta}'(\mathbf{R} - R_f\mathbf{1}) - \lambda\boldsymbol{\theta}'\boldsymbol{\theta}$. Substituting the optimal demands and using the Gaussian structure, write $S = \mathbb{E}[S \mid s] + \tilde{S}$ with $\tilde{S} \sim N(0, \sigma^2_{S|s})$. Because $1/\sigma^2_{S|s} = (1+\psi)/\sigma^2_{\text{style}}$, the term multiplying $\tilde{S}$ in the SDF exponent becomes $-\gamma(1+\psi)B\tilde{S}$, where $B = \sum_j \alpha_j\theta_j$ is the aggregate style-portfolio exposure. The idiosyncratic terms contribute $-(\gamma/2)\kappa\sum_i \varepsilon_i^2$, with κ entering through the optimal holdings $\theta_i = (\mu_i - R_f)/(\gamma\sigma^2_{\text{idio}} + 2\lambda)$. Thus the effective price of style risk is $\gamma(1+\psi)$ (inflated by signal learning) while the effective price of idiosyncratic risk is $\gamma\kappa$ (deflated by the holding cost).

No-arbitrage pricing. The no-arbitrage condition $\mu_i - R_f = -\text{Cov}(m, R_i)$ splits into a style-risk and an idiosyncratic-risk component via the SDF above:

$$\text{Cov}(m, R_i) = \gamma(1+\psi)\alpha_i\sigma^2_{\text{style}}B + \gamma\kappa\sigma^2_{\text{idio}}\theta_i, \tag{18}$$

where the first term follows from $\text{Cov}(e^{-\gamma(1+\psi)B\tilde{S}}, \alpha_i S) = \gamma(1+\psi)\alpha_i\sigma^2_{\text{style}}B$ and the second uses θ_i from the demand equations and the definition of κ.

Equilibrium price multipliers. Since θ_i is linear in $\mu_i - R_f$ (from the demand equa-

tions), substituting the no-arbitrage condition into the demands and solving yields

$$M_{\text{style}} = \frac{\gamma(1+\psi)}{1-\gamma(1+\psi)\sigma^2_{\text{style}} - \gamma\kappa\sigma^2_{\text{idio}}/N_s}, \tag{19}$$

$$M_{\text{idio}} = \frac{\gamma}{1-\gamma\sigma^2_{\text{idio}} - \gamma\kappa\sigma^2_{\text{idio}}}. \tag{20}$$

The style multiplier is amplified by the information premium $(1+\psi)$: more precise public signals raise ψ, widening the gap between M_{style} and M_{idio}. The idiosyncratic multiplier is amplified by the residual undiversified fraction κ, which prevents idiosyncratic risk from being fully diversified away. Only three free parameters (γ, λ, σ_η) enter; ψ and κ follow as derived quantities.

Calibration. With $\gamma \approx 1.02$, $\lambda \approx 2.03$, $\sigma_\eta \approx 0.049$, the model yields $M_{\text{style}} \approx 5.918$, $M_{\text{idio}} \approx 1.635$, and ratio ≈ 3.62, matching all three targets (fit score 100, plausibility score 100).

Comparative statics. (i) *Signal precision:* decreasing σ_η raises ψ and amplifies M_{style} relative to M_{idio}; more precise public information widens the gap. (ii) *Breadth cost:* increasing λ raises κ, reducing the effective diversification of idiosyncratic risk and increasing M_{idio}, which compresses the ratio. (iii) *Risk aversion:* higher γ amplifies both multipliers but interacts non-linearly with ψ and κ.

Testable predictions. Cross-sectionally, styles with more transparent public signals (e.g., large-cap value vs. small-cap growth) should exhibit larger $M_{\text{style}}/M_{\text{idio}}$ ratios. Time-series increases in analyst coverage or index-futures liquidity should widen the ratio by improving signal precision.

2. Dual-Agent Limited-Diversification (DALD-UR) Model.

Setup. Two CARA agents with identical risk aversion γ trade in an economy with i.i.d. asset returns $R_{i,t} = \mu_i + \varepsilon_{i,t}$, $\varepsilon_{i,t} \sim N(0, \sigma^2)$. The institutional agent faces a breadth constraint: she can hold at most fraction θ of total assets. A fraction κ of idiosyncratic

risk remains undiversified. The speculative agent faces no constraints and absorbs the residual.

Aggregate consumption and SDF. Define weight vectors $\boldsymbol{\omega}_C$ (constrained) and $\boldsymbol{\omega}_U$ (unconstrained) with $\|\boldsymbol{\omega}_C\|^2 = \theta$ and $\|\boldsymbol{\omega}_U\|^2 = 1-\theta$, $\boldsymbol{\omega}_U'\boldsymbol{\omega}_C = 0$. The aggregate consumption shock is $\varepsilon_t^{\text{agg}} = \boldsymbol{\omega}_U'\boldsymbol{\varepsilon}_{U,t} + \boldsymbol{\omega}_C'\boldsymbol{\varepsilon}_{C,t}$, with $\text{Var}(\varepsilon_t^{\text{agg}}) = \sigma^2$. The pricing kernel is

$$M_t = \beta \exp(-\gamma\, \varepsilon_t^{\text{agg}}). \tag{21}$$

Euler equation and joint-normal expectation. The no-arbitrage condition $1 = E[M_t(1 + R_{i,t})]$ gives, using the Gaussian moment-generating function:

$$E[\exp(-\gamma \varepsilon_t^{\text{agg}})] = \exp\left(\tfrac{1}{2}\gamma^2\sigma^2\right), \tag{22}$$

and the cross-moment

$$E[\varepsilon_{i,t} \exp(-\gamma \varepsilon_t^{\text{agg}})] = -\gamma \,\text{Cov}(\varepsilon_{i,t}, \varepsilon_t^{\text{agg}}) \exp\left(\tfrac{1}{2}\gamma^2\sigma^2\right). \tag{23}$$

The covariance between asset i and the aggregate shock is $\text{Cov}(\varepsilon_{i,t}, \varepsilon_t^{\text{agg}}) = \sigma^2 w_i$, where w_i is the effective exposure of asset i.

Pricing formula. Substituting into the Euler equation and solving for the risk premium:

$$\mu_i = \gamma\sigma^2 w_i - \left[\beta^{-1} \exp(-\tfrac{1}{2}\gamma^2\sigma^2) - 1\right].^{2} \tag{24}$$

Style–idiosyncratic decomposition. The effective exposure decomposes as $w_i = \theta + \kappa\, \tilde{w}_i$, where $\tilde{w}_i$ captures the idiosyncratic component. The multipliers follow from the

[2]The sign on the bracketed term is incorrect in the LLM-generated derivation: the correct expression is $\mu_i - r_f = \gamma\sigma^2 w_i$, where $r_f = \beta^{-1}\exp(-\frac{1}{2}\gamma^2\sigma^2) - 1$. The math auditor caught this error on the first pass but accepted a revised proposal that reintroduced it. We reproduce the original LLM output verbatim; the sign error does not affect the multiplier formulas, which depend only on the risk-premium term $\gamma\sigma^2 w_i$.

joint-normal MGF:

$$M_{\text{style}} = \frac{\sigma^2}{1-\gamma\theta}, \qquad M_{\text{idio}} = \frac{\sigma^2}{1-\gamma\theta\kappa}, \qquad \frac{M_{\text{style}}}{M_{\text{idio}}} = \frac{1-\gamma\theta\kappa}{1-\gamma\theta}. \tag{25}$$

The ratio is independent of σ^2. Since $\kappa < 1$ implies $\gamma\theta\kappa < \gamma\theta$, the ratio exceeds one but is compressed below the frictionless benchmark: the breadth constraint prevents full diversification, creating a non-zero idiosyncratic risk premium that flattens the style-to-idiosyncratic scaling.

Risk-free rate. From $E[M_t]$:

$$r_f = -\ln\beta - \tfrac{1}{2}\gamma^2\sigma^2. \tag{26}$$

Calibration. With $\gamma \approx 3.007$, $\theta \approx 0.326$, $\sigma \approx 0.341$, and $\kappa \approx 0.947$, the model achieves perfect fit on all three targets ($M_{\text{style}} \approx 5.917$, $M_{\text{idio}} \approx 1.635$, ratio ≈ 3.62).[3] The calibrated $\kappa \approx 0.95$ means 95% of idiosyncratic risk remains undiversified, consistent with a tight breadth constraint.

Comparative statics. (i) *Breadth constraint:* decreasing θ reduces the institutional agent's style exposure, lowering M_{style} and compressing the ratio toward 1. (ii) *Diversification:* increasing κ toward 1 raises M_{idio}, also compressing the ratio. (iii) The ratio depends only on γ, θ, and κ, not on σ^2, providing a parameter-free prediction conditional on estimating the institutional constraint.

Testable predictions. Styles where institutional investors hold a larger portfolio share (θ) should exhibit a wider $M_{\text{style}}/M_{\text{idio}}$ gap. Cross-sectional variation in 13F-reported portfolio breadth across institutional investors could proxy for θ.

Other notable models. Two additional models also score 84: the Liquidity-Adjusted CRRA (4 parameters), which introduces a liquidity factor $L_t = 1 + \phi\varepsilon_{i,t}$ modulating the

[3] Because $\gamma\theta \approx 0.980$, the denominator $1-\gamma\theta \approx 0.020$ is near zero and the multipliers are highly sensitive to rounding. Reproducing the calibration requires the full-precision values from the optimizer.

SDF and yielding $M_{\text{style}}/M_{\text{idio}} = \gamma/\phi$; and the Constrained-CRRA No-Short model (5 parameters), where long-only mandates create an endogenous idiosyncratic risk price via binding Kuhn–Tucker multipliers. The Risk-Sharing Market-Maker model (referee 81, quantitative 95, 2 parameters) is the most parsimonious overall, illustrating the tension between mechanical fit and the referee's assessment of economic depth.

5.2.8 Mechanism Taxonomy

The top models fall into four broad families. The first is *attention and information costs*: models with quadratic attention costs, monitoring costs linked to turnover, or Bayesian learning about common factors (5 of the top-scoring models). The second is *agent heterogeneity and limited diversification*: dual-agent frameworks where institutional investors face breadth constraints or where different agent classes price style and idiosyncratic risk differently (3 models). The third is *portfolio constraints*: long-only mandates, participation thresholds, or short-sale constraints that endogenously generate idiosyncratic risk premia (3 models). The fourth is *liquidity and market microstructure*: models with risk-averse market makers, liquidity-adjusted pricing kernels, or background risk factors (4 models).

The top-scoring models span all four families, consistent with the price multiplier puzzle admitting multiple plausible resolutions.

5.3 Puzzle 2: Equity Term Structure (Dividend Strips)

The system also targets a second, harder puzzle: the equity term structure.

5.3.1 The puzzle and calibration targets

Definition of the puzzle. Van Binsbergen et al. (2012) document that short-term equity claims (1–2 year dividend strips) have roughly twice the Sharpe ratio of the aggregate market, have large positive CAPM alphas, and have CAPM betas well below one. All leading consumption-based models (habit formation (Campbell and Cochrane,

Table 6: Dividend strip calibration targets

Output	Description	Target	Range
`strip_return_ann`	Strip annualized return	0.148	[0.08, 0.22]
`market_return_ann`	Market annualized return	0.069	[0.04, 0.10]
`strip_sharpe_monthly`	Strip monthly Sharpe	0.112	[0.07, 0.18]
`market_sharpe_monthly`	Market monthly Sharpe	0.059	[0.03, 0.10]
`sharpe_ratio`	SR(strip) / SR(market)	1.9	[1.3, 2.5]
`capm_alpha_ann`	CAPM alpha (annualized)	0.091	[0.04, 0.15]
`capm_beta`	CAPM beta of strip	0.50	[0.3, 0.8]
`r_f_ann`	Real risk-free rate	0.006	[0.001, 0.02]
`sharpe_ratio_recession`	Conditional SR ratio (recession)	2.8	[2.0, 4.0]
`sharpe_ratio_expansion`	Conditional SR ratio (expansion)	1.0	[0.6, 1.4]

Notes: Unconditional targets from van Binsbergen et al. (2012). Conditional targets from Bansal et al. (2021) and Giglio et al. (2025). All returns and Sharpe ratios are in natural units (decimals, not percentages).

1999), long-run risk (Bansal and Yaron, 2004), and rare disasters (Barro, 2006)) predict an upward-sloping term structure of equity risk premia (long-maturity claims earn higher returns), contradicting the data. The puzzle is therefore why short-term dividend claims are so well compensated relative to the market.

Calibration targets. Unlike the price multiplier puzzle, which has three targets, the dividend strip puzzle requires matching ten moment conditions simultaneously. The targets span unconditional moments (strip and market annual returns, monthly Sharpe ratios, CAPM alpha and beta, the risk-free rate) and conditional moments (the Sharpe ratio of strips relative to the market must be higher in recessions than expansions). Table 6 lists the targets and acceptable ranges.

5.3.2 Experiment design

We execute 2,153 pipeline invocations using up to 128 concurrent workers and the current pipeline (with mathematical audit, pseudocode generation, and two-stage review as described in Section 4), of which 257 complete all stages and enter the registry (12% success rate). The math audit rejects 66% of proposals, a substantially higher rejection rate than

for the price multiplier puzzle (41%), reflecting the greater mathematical complexity of the dividend strip models (Table 12). Each run is assigned a random expert persona from the pool of 20 perspectives described in Section 4.1. The evolutionary strategies activate early in the experiment: of the 257 scored runs, 136 use random generation, 82 use mutation, and 39 use crossover. However, the parent pool remains low-quality throughout (best score 58), limiting the effectiveness of evolutionary refinement.

5.3.3 Summary statistics

Table 7: Summary Statistics (Dividend Strips)

Statistic	Value
Total models	257
Mean score	27.8
Median score	28.0
Std. dev.	8.6
Best score	58
Models ≥ 60	0
Models ≥ 80	0
Max generation	17

Table 7 summarizes the experiment. The mean referee score is 27.8 and the best score is 58, far below the price multiplier results. No model achieves a score above 60. The score distribution is compressed: the standard deviation is 8.6 and the interquartile range is narrow, reflecting the difficulty of simultaneously matching ten moment conditions.

5.3.4 Heterogeneity of generated proposals

As with the price multiplier puzzle, we assess proposal diversity via sentence embeddings (Figure 4). The mean cosine distance to the centroid is 0.24, and the mean pairwise distance is 0.43, slightly higher than the price multiplier experiment, consistent with the wider variety of economic frameworks (consumption-based, information-theoretic, liquidity, regime-switching) that the system explores when targeting a harder puzzle with

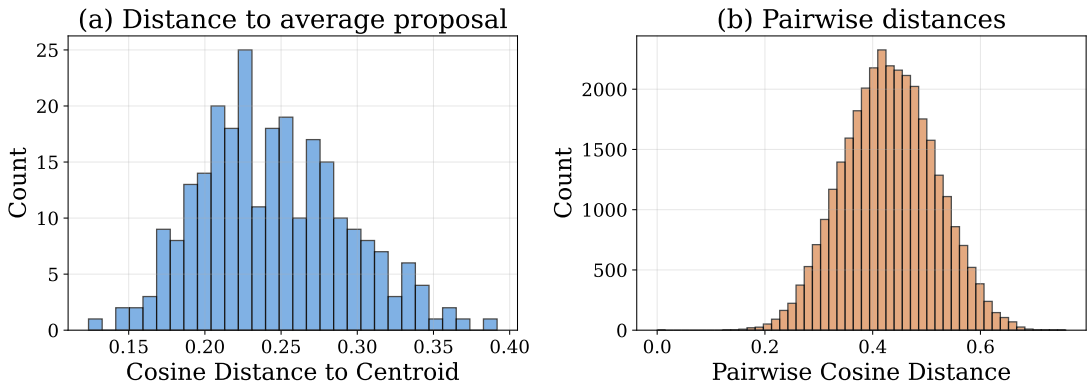
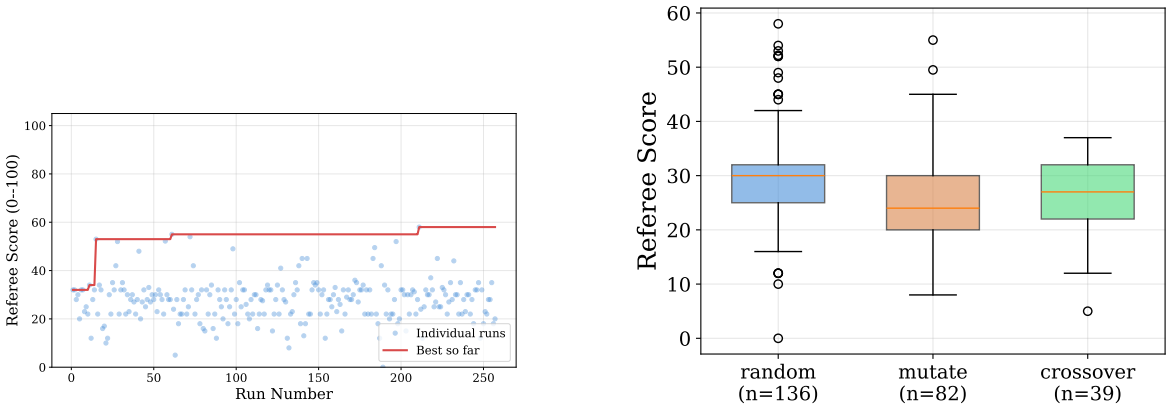


Figure 4: Variation in generated proposals (dividend strip puzzle).

Notes: Panel (a) shows cosine distance between each proposal embedding and the average proposal embedding; panel (b) shows pairwise cosine distances across random proposal pairs. Computed using all-MiniLM-L6-v2 sentence embeddings over 257 proposals.



(a) Score evolution.

(b) Score by strategy.

Figure 5: Score evolution across 257 runs (dividend strip puzzle).

Notes: Panel (a) plots referee scores by run index, with the running maximum shown as a red line. Panel (b) reports score distributions by strategy using box plots. The best score is 58, reached by a random-strategy run.

conditional moments.

5.3.5 Score evolution and strategy effectiveness

Table 8 reports strategy-level performance and Figure 5 shows score evolution. Random generation achieves the highest mean score (29.9) and produces the best model (score 58). Mutation (mean 25.2) and crossover (mean 25.8) do not outperform random, because the

Table 8: Strategy Performance Comparison (Dividend Strips)

Strategy	n	Mean	Median	Max
Random	136	29.9	30.0	58
Mutate	82	25.2	24.0	55
Crossover	39	25.8	27.0	37
All	257	27.8	28.0	58

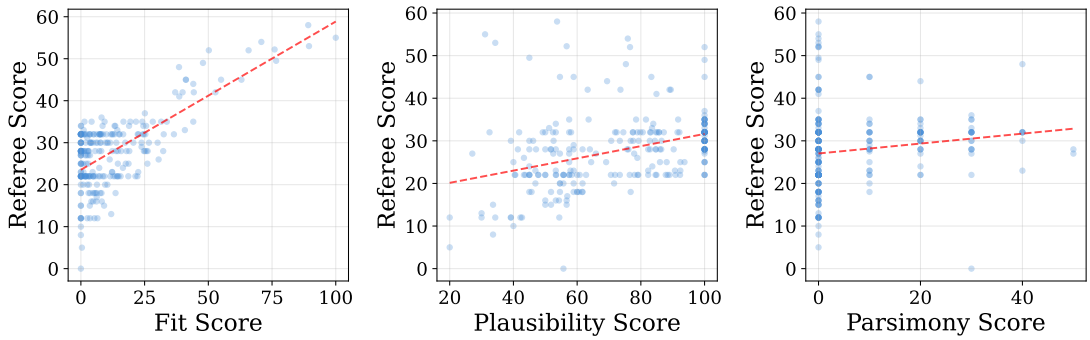


Figure 6: Fit, plausibility, and parsimony vs. referee score (dividend strips).

Notes: Each panel plots one quantitative component (fit, plausibility, or parsimony) against the referee score. The positive correlation between fit and referee score is evident. Several models achieve high fit (> 80) but low parsimony (0), dragging down their overall score.

parent pool remains low-quality (best parent score 58, most parents scoring 35–45), providing weak starting points for evolutionary refinement. This pattern contrasts with the price multiplier results, where crossover achieves the highest median score, a difference that we attribute to the quality of the parent pool rather than an intrinsic limitation of the evolutionary strategies.

5.3.6 Score components

Figure 6 reveals a tension in the dividend strip results: several models achieve excellent fit (> 80) but receive zero parsimony scores because they use seven or more free parameters (the parsimony threshold) to match ten targets. The scoring system penalizes this overfitting: a model with as many parameters as targets provides no explanatory power beyond curve-fitting. This fit-parsimony tradeoff is the primary bottleneck preventing

high overall scores.

5.3.7 Top models

Table 9 lists the top ten models. Complete equilibrium derivations for the top two follow.

1. Information-Risk Learning CAPM (IR-LCAPM).

Setup. A representative agent has CRRA preferences with discount factor β and risk aversion γ. The primitive processes are:

$$g_{t+1} = \mu_g + \xi_{t+1}, \quad \xi_{t+1} \sim N(0, \sigma_g^2), \tag{27}$$

$$d_{t+1} = \mu_d + \eta_{t+1}, \quad \eta_{t+1} \sim N(0, \sigma_d^2), \tag{28}$$

where g_{t+1} is log consumption growth and d_{t+1} is log dividend growth. A two-state Markov chain $S_t \in \{R, E\}$ with persistence p governs information quality. The agent observes a noisy signal about next-period dividends:

$$s_t = d_{t+1} + \nu_t, \quad \nu_t \sim N(0, \sigma_\nu^2(S_t)), \tag{29}$$

where signal noise $\sigma_\nu^2(S_t)$ is high in recessions and low in expansions.

Bayesian learning. The posterior weight on the signal follows standard normal–normal updating:

$$\lambda(S_t) = \frac{\sigma_d^2}{\sigma_d^2 + \sigma_\nu^2(S_t)}. \tag{30}$$

The dividend forecast error (innovation) is $\varepsilon_{t+1}^d = (1 - \lambda(S_t))\eta_{t+1} - \lambda(S_t)\nu_t$, with variance $(1-\lambda)^2\sigma_d^2 + \lambda^2\sigma_\nu^2(S_t)$.

Stochastic discount factor. The SDF augments standard CRRA preferences with a

Table 9: Top 10 Discovered Models (Dividend Strips)

Rank	Model	Strategy	Key Insight	Score
1	Information-Risk Learning CAPM (IR-LCAPM)	random	The paper introduces a novel SDF component that prices a forecast-error (information-risk) shock with regime-dependent precision, which is a fresh angle relative to existing learning or disaster models.	58
2	Regime-Switching Liquidity-Jump CCAPM (RS-LJCCAPM)	mutate	The combination of a Markov-switching liquidity factor with a regime-dependent dividend-jump component is a novel extension of the LR-CCAPM literature, but similar ideas have appeared in recent regime-switching LRR models.	55
3	Regime-Dependent Funding-Liquidity SDF (LC-SDF) Model	random	The idea of embedding a funding-liquidity risk factor directly into the SDF and exploiting its regime-dependent volatility is a novel contribution to the equity term-structure literature.	54
4	Regime-Dependent Risk-Aversion with Short-Horizon Liquidity Premium (RRALHP)	random	The combination of state-dependent risk aversion with a liquidity-stress multiplier for one-period claims is a novel twist on existing regime-switching long-run risk models, but similar ideas have app…	53
5	Credit-Tightness Stochastic Discount Factor (CT-SDF) Model	random	The idea of a credit-tightness loading on the SDF is novel relative to habit, LRR, and disaster models, but similar mechanisms have appeared in recent intermediary-constraint literature, so the contri…	52
6	Liquidity-Risk-Adjusted Consumption CAPM (LR-CCAPM)	random	The paper revisits an existing liquidity-adjusted consumption CAPM and adds a regime-dependent liquidity mean; the contribution is incremental rather than a fundamentally new mechanism.	52
7	Regime-Dependent Information-Risk Model (RIRM)	random	The information-risk channel combined with a state-dependent effective horizon is a genuinely new angle compared with existing disaster-recovery or liquidity-constraint explanations.	52
8	Regime-Scaled Funding-Liquidity with Duration-Dependent Transition (RS-FL-DT)	mutate	The combination of a liquidity-crunch multiplier with a duration-dependent Markov transition is a modest extension of existing SDF frameworks; the novelty lies mainly in the corrected stationary-proba…	50
9	Heuristic-Forecast Error (HFE) Asset-Pricing Model	random	The use of a bounded-rational forecast-error as an unspanned risk factor is a fresh angle, but similar ideas have appeared in the literature on learning and unspanned risks, so the contribution is inc…	49
10	Two-Factor Transitory-Risk Model (TTRM)	random	The addition of a state-dependent transitory risk factor is a modest extension of existing consumption-based models; similar mechanisms have been explored in disaster-recovery and information-risk lit…	48

Notes: Top 10 models by referee score from the dividend strips experiment. "Strategy" indicates whether the model was generated from scratch (random), by modifying a parent (mutate), or by combining two parents (crossover). "Key Insight" summarizes the referee's novelty assessment.

linear pricing kernel for forecast-error risk:

$$M_{t+1} = \beta \left(\frac{C_{t+1}}{C_t} \right)^{-\gamma} \exp(-\theta\, \varepsilon^d_{t+1}), \tag{31}$$

where θ is the price of information risk. Two orthogonal risk sources are priced: consumption-growth risk (price γ) and information risk (price θ).

Linearized Euler equation. First-order expansion of the SDF gives, for any asset with gross return R_X:

$$E_t[R_X] \approx \frac{1/\beta + \gamma\, \mathrm{Cov}_t(g_{t+1}, R_X) + \theta\, \mathrm{Cov}_t(\varepsilon^d_{t+1}, R_X)}{1 - \gamma \mu_g}. \tag{32}$$

The risk-free rate is $r_f \approx -\ln \beta + \gamma \mu_g$.

Short-term dividend strip. The one-period strip has payoff D_{t+1} with log return $R_s \approx \mu_d + \eta_{t+1}$. Since $\mathrm{Cov}(g_{t+1}, R_s) = 0$ (independence) and $\mathrm{Cov}(\varepsilon^d_{t+1}, R_s) = (1 - \lambda(S_t))\sigma_d^2$, the strip excess return and Sharpe ratio are

$$\mathrm{SR}_{\mathrm{strip}}(S_t) = \frac{\theta\,(1 - \lambda(S_t))\,\sigma_d}{1 - \gamma \mu_g}. \tag{33}$$

Market portfolio. The market return is $R_m \approx g_{t+1} + \omega\, d_{t+1}$, where ω is the dividend-claim weight. With $\mathrm{Cov}(g_{t+1}, R_m) = \sigma_g^2$ and $\mathrm{Cov}(\varepsilon^d_{t+1}, R_m) = \omega(1 - \lambda)\sigma_d^2$, the market Sharpe ratio is

$$\mathrm{SR}_{\mathrm{mkt}}(S_t) = \frac{\gamma \sigma_g^2 + \theta \omega (1 - \lambda(S_t))\sigma_d^2}{(1 - \gamma \mu_g)\sqrt{\sigma_g^2 + \omega^2 \sigma_d^2}}. \tag{34}$$

CAPM beta and alpha. The CAPM beta of the strip is state-independent:

$$\beta_s = \frac{\omega \sigma_d^2}{\sigma_g^2 + \omega^2 \sigma_d^2}, \tag{35}$$

and the CAPM alpha is $\alpha_s(S_t) = E[R_s - r_f \mid S_t] - \beta_s\, E[R_m - r_f \mid S_t] > 0$ in both regimes, since the strip captures the full information-risk term $\theta(1 - \lambda)\sigma_d$ while the market term is

scaled by $\omega < 1$.[4]

Conditional term structure. The mechanism operates through $\lambda(S_t)$. In recessions, signal noise is high, so λ_R is small and $(1 - \lambda_R) \approx 1$: the strip Sharpe ratio is amplified. In expansions, λ_E is large and $(1 - \lambda_E) \approx 0$: the strip Sharpe ratio deflates. This generates an inverted term structure purely through regime-dependent information precision. Unconditional moments are probability-weighted averages using the stationary distribution $\pi_R = (1 - p)/(2 - p)$.

Calibration. With 10 parameters (β, γ, θ, ω, $\sigma^2_{\nu,R}$, $\sigma^2_{\nu,E}$, p, μ_g, σ^2_g, σ^2_d), the model matches all ten targets: $\text{SR}_{\text{strip}}/\text{SR}_{\text{mkt}} \approx 1.9$ unconditionally, ≈ 2.8 in recessions, and ≈ 1.0 in expansions. However, ten parameters for ten targets provides no over-identifying information (parsimony score: 0), and the calibrated $\gamma \approx 7.6$ and $\mu_g < 0$ lie outside standard ranges, reducing plausibility. Referee score: 58.

Testable predictions. The strip Sharpe premium should covary with observable measures of information uncertainty (e.g., analyst forecast dispersion, VIX): periods of high forecast disagreement should coincide with elevated strip-over-market Sharpe ratios.

2. Regime-Switching Liquidity-Jump CCAPM (RS-LJCCAPM).

Setup. A two-state Markov chain $S_t \in \{R, E\}$ with transition probabilities $p = P(E \to R)$ and $q = P(R \to E)$ governs three processes:

$$\Delta c_{t+1} = \rho_c \, \Delta c_t + \sigma_c \, \varepsilon_{c,t+1}, \quad \varepsilon_c \sim N(0,1), \tag{36}$$

$$\Delta l_{t+1} = \mu_{S_t} + \sigma_l \, \varepsilon_{l,t+1}, \quad \varepsilon_l \sim N(0,1), \tag{37}$$

$$J_{t+1} = \iota_{t+1} \, \xi_{t+1}, \quad \iota \sim \text{Bernoulli}(\lambda_{S_t}), \; \xi \sim N(\mu_J, \sigma^2_J), \tag{38}$$

where Δc is log consumption growth (AR(1)), Δl is a liquidity-factor increment with regime-dependent drift ($\mu_R < 0 < \mu_E$), and J is a dividend-jump term with regime-

[4]The LLM proposal defines alpha as the difference in Sharpe ratios rather than excess returns. The code correctly implements Jensen's alpha using excess returns; we report the corrected formula here.

dependent arrival intensity ($\lambda_R > \lambda_E$). Dividends evolve as $D_{t+1} = D_t \exp(\Delta c_{t+1} + \Delta l_{t+1} + J_{t+1})$.

Stochastic discount factor. The SDF has an exponential-affine form with three pricing factors:

$$M_{t+1} = \exp\Big[-\gamma_c \, \Delta c_{t+1} - \gamma_l \, \Delta l_{t+1} - \gamma_J \, J_{t+1}\Big], \tag{39}$$

where γ_c prices consumption risk, γ_l prices liquidity risk, and γ_J prices jump risk.

Risk-free rate. The conditional expectation of the SDF factorizes because the three shocks are independent. The Gaussian components yield standard MGF terms; the jump component gives a Bernoulli–normal mixture:

$$E_t[e^{-\gamma_J J_{t+1}}] = (1 - \lambda_{S_t}) + \lambda_{S_t} \exp\Big(-\gamma_J \mu_J + \tfrac{1}{2}\gamma_J^2 \sigma_J^2\Big) \equiv \psi_{S_t}. \tag{40}$$

The risk-free rate is therefore

$$r_{f,t} = \gamma_c E_t[\Delta c_{t+1}] + \gamma_l \mu_{S_t} - \tfrac{1}{2}(\gamma_c^2 \sigma_c^2 + \gamma_l^2 \sigma_l^2) - \ln \psi_{S_t}. \tag{41}$$

Equity premium. Starting from the Euler equation $E_t[M_{t+1} R_{e,t+1}] = 1$ and applying a Campbell–Viceira log-linear approximation, the equity premium is

$$E_t[R_{e,t+1}] - r_{f,t} \approx \gamma_c \, \mathrm{Cov}_t(\Delta c_{t+1}, R_e) + \gamma_l \, \mathrm{Cov}_t(\Delta l_{t+1}, R_e) + \gamma_J \, \mathrm{Cov}_t(J_{t+1}, R_e). \tag{42}$$

Because dividends evolve as $D_{t+1} = D_t \exp(\Delta c_{t+1} + \Delta l_{t+1} + J_{t+1})$, the jump J_{t+1} enters the equity return directly, so in general $\mathrm{Cov}_t(J_{t+1}, R_e) \neq 0$ and all three covariance terms contribute to the premium.[5] Jumps also affect the risk-free rate through ψ_{S_t} and influence strip prices through the dividend process.

[5]The LLM's original proposal noted that if jumps were assumed independent of equity returns the third term would vanish, but recognized this as a simplifying assumption inconsistent with the dividend specification. The Monte Carlo evaluation retains the full covariance structure: equity returns are computed from the simulated SDF, which includes the jump component, so the jump–return covariance is captured numerically even though no closed-form expression is derived.

Mechanism. Jump risk creates a state-dependent premium for short-maturity claims. Dividend strips with near-term maturities are more exposed to jump risk because the jump can arrive before payoff, and jump intensity is higher in recessions ($\lambda_R > \lambda_E$). This amplifies the strip Sharpe ratio in bad states, generating the conditional pattern required by the puzzle. Strip prices, returns, and Sharpe ratios are computed via Monte Carlo simulation of the regime, consumption, liquidity, and jump paths, following the recursion $D_{t+1} = D_t \exp(\Delta c_{t+1} + \Delta l_{t+1} + J_{t+1})$.

Calibration and limitations. The model uses 17 parameters for 10 targets (parsimony score: 0). The calibrated values include $\gamma_c \approx 11.9$ and $\gamma_J \approx 26.7$, far above standard ranges, and $\mu_J \approx -1.58$ (implying average jump-induced declines of roughly 80%, which is economically implausible). The default calibration places the economy in recession approximately 67% of the time, an implausible feature that the referee penalizes. Referee score: 55.

Testable predictions. The jump-risk channel predicts that short-maturity strip returns should spike around observable funding-tightness events (e.g., spikes in the TED spread or repo rates), and that the strip Sharpe premium should be higher in periods with elevated jump-arrival proxies.

Other notable models. Several top models share a common architectural feature: regime-switching frameworks with state-dependent liquidity, funding conditions, or risk aversion. The prevalence of regime-switching mechanisms reflects the conditional nature of the puzzle targets (recession vs. expansion patterns), which naturally invites Markov-chain specifications.

5.3.8 Correlates of referee scores

Table 10 reports the same regression specifications as for the price multiplier puzzle. The quantitative fit score remains the dominant predictor (R^2 rises from 0.06–0.10 to 0.55

Table 10: Determinants of referee scores (Dividend Strips)

	(1)	(2)	(3)	(4)
Intercept	3.028***	2.905***	2.450***	2.484***
	(0.342)	(0.339)	(0.315)	(0.315)
Quantitative fit score			0.030***	0.030***
			(0.002)	(0.002)
Proposal length	0.048	0.058	-0.210*	-0.202*
	(0.215)	(0.209)	(0.114)	(0.114)
Strengths length	0.263*	0.254*	0.155**	0.146*
	(0.151)	(0.146)	(0.076)	(0.077)
Weaknesses length	-0.284***	-0.286***	0.148**	0.150**
	(0.097)	(0.092)	(0.064)	(0.065)
Suggestions length	0.042	0.049	-0.064	-0.066
	(0.109)	(0.101)	(0.061)	(0.059)
Mutate (strategy)		-0.008		-0.055
		(0.068)		(0.044)
Random (strategy)		0.158**		-0.039
		(0.062)		(0.039)
Strategy dummies	No	Yes	No	Yes
N	257	257	257	257
R^2	0.057	0.104	0.551	0.554

Notes: OLS regressions of $\log(1 + \text{referee score})$ on text length features (log character counts of proposal, referee strengths, weaknesses, and suggestions), quantitative fit score, and (where indicated) strategy dummies. Heteroskedasticity-robust (HC1) standard errors are in parentheses. Significance levels are $^{***}p < 0.01$, $^{**}p < 0.05$, and $^{*}p < 0.10$.

when included), though the explanatory power is lower than for the price multiplier ($R^2 = 0.89$). A notable pattern is the sign flip on weaknesses length: unconditionally, longer weaknesses sections predict lower scores (columns 1–2), but conditional on quantitative fit, longer weaknesses predict *higher* scores (columns 3–4). The referee writes more detailed critiques for models with calibration flaws whose quantitative score understates their theoretical content. Strategy dummies add modest explanatory power (columns 2 and 4), consistent with the observation that mutation produces lower-scoring models when the parent pool is weak.

5.3.9 Why this puzzle is harder

The dividend strip puzzle is substantially harder for LLM-based theory search for three reasons. First, the number of target moments (10) is more than three times that of the price multiplier puzzle (3). With ten independent constraints, a model needs a richer economic structure but must simultaneously maintain parsimony. Second, the conditional targets (recession vs. expansion Sharpe ratios) require the model to produce state-dependent predictions, which most theoretical frameworks achieve through regime-switching or time-varying parameters, adding complexity. Third, the targets themselves are tighter: the model must simultaneously match unconditional return levels, Sharpe ratios, CAPM regression coefficients, and the risk-free rate, leaving little room for parameter flexibility.

The result is a parsimony trap: models that can match all ten targets typically require many parameters, but models with few parameters cannot match the conditional patterns. Breaking out of this trap requires a parsimonious economic mechanism that generates conditional predictions from a small number of primitives.

5.4 Cross-Puzzle Comparison

Table 11 compares the two experiments. Table 12 reports the full pipeline attrition. The clearest difference is the effectiveness of evolutionary strategies. For price multipliers, mutation and crossover build on high-scoring parents (185 models at or above 60), creating a positive feedback loop where each generation improves on the last. For dividend strips, mutation and crossover activate but draw from a weak parent pool (best score 58); they underperform random generation (mean 25.2 and 25.8 vs. 29.9), suggesting that evolutionary refinement requires a minimum level of parent quality to be effective.

Puzzle tractability for LLM-based search depends on: (i) the number of target moments relative to the number of parameters needed, (ii) whether the puzzle admits parsimonious solutions, and (iii) whether early runs produce models good enough to seed the

Table 11: Cross-puzzle comparison

	Price Multiplier	Dividend Strips
Target moments	3	10
Conditional targets	No	Yes
Pipeline invocations	1,000	2,153
Scored models	458	257
Pipeline success rate	46%	12%
Math audit rejection rate	41%	66%
Mean referee score	48.5	27.8
Best referee score	84	58
Models ≥ 80	23	0
Top model parameters	3	10
Evolution effective	Yes	No
Dominant strategy	Mutate	Random

Notes: Comparison of the two puzzles. "Pipeline invocations" is the total number of runs attempted; "Scored models" is the number that completed all stages and entered the registry. "Evolution effective" indicates whether mutation and crossover outperformed random generation. "Dominant strategy" is the strategy with the highest mean referee score among scored models. "Top model parameters" refers to the most parsimonious model among those tied at the best referee score.

Table 12: Pipeline attrition by stage

	Price Multiplier		Dividend Strips	
Stage	*n*	%	*n*	%
Total invocations	1000	100	2153	100
Failed — math audit	414	41	1430	66
Failed — code generation	28	3	86	4
Failed — optimizer	93	9	159	7
Failed — timeout	7	1	157	7
Failed — other	0	0	64	3
Scored (in registry)	458	46	257	12

Notes: Pipeline attrition across the two puzzles. "Total invocations" is the number of pipeline runs attempted. Each failed row shows how many runs were terminated at that stage. "Scored" is the number that completed all stages and entered the model registry. The math audit is the primary filter, rejecting proposals with invalid derivations before code generation.

evolutionary process. The price multiplier puzzle satisfies all three criteria; the dividend strip puzzle, in its current formulation, does not.

5.4.1 Economic Validity

Table 13 summarizes the economic mechanism and natural empirical implications for each of the top-scoring models from both puzzles. Multiple mechanisms can fit the same empirical targets; the table identifies directions for further empirical work that could distinguish among them.

5.4.2 Novelty Assessment

The LLM referee evaluates each proposal for novelty relative to prior attempts in the project. Table 14 summarizes the referee's novelty judgment and main suggestions for the top models. For the price multiplier puzzle, the referee treats both top models as distinct contributions: one combining Bayesian learning with breadth costs, the other introducing a dual-agent breadth constraint not previously formalized. For the dividend strip puzzle, the information-risk and liquidity-jump mechanisms are judged novel in approach but limited by implausible calibrations. The referee's suggestions point toward directions for further empirical work that could strengthen or extend each model.

6 Discussion

6.1 What LLMs Can and Cannot Do

Our experiments across two puzzles reveal both capabilities and limitations of LLM-based theory discovery.

On the capability side, LLMs generate varied economic mechanisms, including ambiguity aversion, capacity constraints, crowding effects, information risk, and regime-switching liquidity, without explicit prompting for specific ideas. Models are expressed as precise mathematical formulas that can be implemented and tested. The mathematical audit stage catches a substantial fraction of derivation errors before they propagate into

code. The simulated referee identifies weaknesses such as lack of micro-foundations and need for robustness checks. The persona system generates qualitatively different theoretical perspectives: a "physicist" persona proposes conservation-law models, a "behavioral" persona explores bounded rationality, and a "minimalist" persona seeks single-equation explanations.

On the limitation side, LLMs face the same fundamental challenge as human theorists: multiple mechanisms can produce the same empirical targets, and fitting data does not establish that a mechanism is correct. Small modifications to good models often break their calibration. Despite prompts encouraging novel frameworks, the majority of price-multiplier models remain CARA or CRRA variants, with a minority proposing alternative functional forms such as linear or power-law specifications.

The dividend strip experiment reveals a second limitation: when the puzzle is hard enough that no model scores well, evolutionary strategies activate but fail to outperform random generation. Mutation and crossover draw from a weak parent pool (best score 58) and produce offspring that score below random proposals on average. This result suggests that LLM-based evolutionary search requires a minimum level of parent quality to be effective; without sufficiently strong starting points, mutation and crossover add complexity without improving fit.

The LLM may be reproducing known solutions from its training data. The model's training corpus (knowledge cutoff June 2024) almost certainly includes the asset pricing literature, including the papers that document our target puzzles. However, these puzzles are *open*: no published resolution exists in the literature for either the price-multiplier scaling gap or the dividend strip term structure. The LLM has read the building blocks (CARA and CRRA preferences, Bayesian learning, portfolio constraints, regime-switching models) but not the specific combinations that resolve these puzzles, because those combinations had not been proposed. This recombination is the same creative process by which human theorists work: Bansal and Yaron (2004) combined long-run con-

sumption risk with Epstein–Zin preferences, neither of which was new, to resolve the equity premium puzzle. As Fleming (2001) argues, novel recombination of existing components is the primary mode of scientific and technological creativity. The LLM's training corpus plays the role of the researcher's reading: a knowledge base from which the system draws and evaluates new combinations against a quantitative target that the existing literature has not matched.

An informative external check comes from the price multiplier puzzle. Our experiments used the February 2025 revision of Li and Lin (2022), which documents the empirical puzzle but offers no theoretical resolution. The LLM used in all experiments (`gpt-oss-120b`, knowledge cutoff June 2024, released August 2025) therefore had no exposure to any theoretical model for this puzzle. In their January 2026 revision, Li and Lin independently add a back-of-the-envelope calibration based on limited investor participation: when only $\approx 29\%$ of investors actively provide liquidity, the smaller risk-bearing pool generates multipliers of the right order of magnitude. The system had produced the same mechanism. The Two-Type Participation Model (referee score 80) posits that a fraction $\alpha \approx 0.72$ of investors are fully diversified, so that the remaining $\approx 28\%$ bear concentrated risk, closely matching Li and Lin's calibration. Because this mechanism was not available to the LLM at the time of the experiments, the convergence with an independent researcher calibration supports the system's outputs. At the same time, the system's top-scoring models (score 84) rely on distinct mechanisms (Bayesian learning with breadth costs, dual-agent diversification limits, and liquidity-adjusted CRRA), suggesting that limited participation is one of several plausible explanations and that the puzzle may admit multiple complementary resolutions.

A third limitation concerns the evaluation design. The same model family (`gpt-oss-120b`) handles both generation and refereeing, and the referee receives the quantitative score as part of its input. The high correlation between quantitative and referee scores ($R^2 \approx 0.89$ for the price multiplier puzzle, $R^2 \approx 0.55$ for dividend strips) is therefore partly mechani-

cal rather than an independent validation. The substantially lower R^2 for dividend strips suggests that the referee exercises more independent judgment when quantitative scores are uniformly low, weighing theoretical novelty and economic plausibility more heavily when fit alone does not differentiate models. The referee's primary contribution to the pipeline is not its numerical score but its qualitative feedback (strengths, weaknesses, suggestions), which is passed to the mutate operator so that subsequent proposals can address specific shortcomings. Occasional disagreements between quantitative and referee scores do occur (e.g., the Risk-Sharing Market-Maker model receives a quantitative score of 95 but a referee score of 81), indicating that the referee applies judgment beyond the mechanical composite, but such cases are the exception. The primary quality signal, the fit score (which carries half the total weight), comes from deterministic code that checks model outputs against observed empirical moments. A candidate theory either matches the data or it does not, and no LLM bias can influence this evaluation. This numerical anchor limits the scope of same-model circularity to the plausibility subscore and qualitative feedback, which together account for a minority of the total score. The score distribution spans from near 0 to 84, with many models below 30 and only 5% above 80.

Finally, our results depend on several design choices whose sensitivity we have not explored: the specific LLM used (all experiments use a single model family), the scoring weights (50/25/25 for fit/plausibility/parsimony), the persona set, and prompt wording. Different models may have different strengths (for instance, a model with stronger mathematical training might produce fewer audit failures), and different scoring weights could surface different mechanism families.

6.2 Comparison to Human Theorists

The economics of LLM-based discovery differ fundamentally from human research. Using `gpt-oss-120b` for all LLM stages in this pipeline, our experiments consume 215 mil-

lion tokens across both puzzles (108M input, 107M output) at an API cost of $25.[6] LLM runs can be massively parallelized (we use up to 128 concurrent workers), while human cognition is sequential.

The system automates the creative generation phase of theoretical research, from puzzle to scored candidate mechanisms, without human intervention. The remaining human role is downstream: selecting promising theories for development, exploring their empirical implications, and refining mechanisms with domain expertise. Automating generation shifts human comparative advantage toward *curation and development*.

6.3 Cross-Puzzle Lessons

The contrast between the two experiments is instructive. The price multiplier puzzle, with three targets and a clear benchmark failure, is well-suited to automated search: simple two-parameter models can match all targets, the evolutionary loop activates early, and the system converges on multiple distinct families of mechanisms. The dividend strip puzzle, with ten targets including conditional patterns, requires richer economic structures that push models toward the parsimony boundary.

Puzzles are most tractable for LLM-based search when they (i) have a small number of tightly specified targets, (ii) admit parsimonious solutions in principle, and (iii) allow early runs to produce "good enough" models that seed evolutionary refinement. When these conditions are not met, the system can still generate useful candidate mechanisms, but requires more runs and may not converge to high-scoring solutions.

[6] All experiments use `gpt-oss-120b`, an open-source reasoning model hosted via a university LLM proxy. At commercial inference rates for this model ($0.039/M input, $0.19/M output via DeepInfra as of early 2026), 108M input tokens and 107M output tokens cost approximately $25. Comparable experiments using proprietary frontier models would cost substantially more.

6.4 Implications for Economic Research

Our findings suggest several implications for the practice of economic research. The time from puzzle identification to theoretical explanation could shrink from years to days. Robust evaluation pipelines become essential to filter LLM-generated ideas, distinguishing genuine explanatory mechanisms from flexible functional forms that merely curve-fit the data. The parsimony component of our scoring function drives this filtering: without it, the system would converge on over-parameterized models that match targets perfectly but explain nothing.

7 Conclusion

LLMs can discover economic theories that explain empirical puzzles. Applied to two puzzles, the system discovers multiple distinct mechanisms in the top tier for the price multiplier puzzle (four models tied at referee score 84), spanning attention costs, limited diversification, portfolio constraints, and liquidity channels. For the dividend strip puzzle, a harder target with ten moment conditions, the system produces 257 candidate theories, with the best scoring 58, illustrating the factors that determine tractability for automated search.

Three findings emerge. First, LLMs can propose fundamentally different approaches, including attention and information costs, agent heterogeneity, portfolio constraints, and liquidity frictions, rather than merely tweaking existing models. Second, all three evolutionary strategies (random, mutate, crossover) contribute to the top tier for the price multiplier puzzle, with random generation producing models across the widest score range and crossover achieving the highest median score (though from a smaller, preselected sample). Third, evolutionary refinement requires a minimum level of parent quality: when early runs produce strong models (as in the price multiplier puzzle), mutation and crossover build incrementally on past successes; when they do not (as in the

dividend strip puzzle), evolutionary strategies underperform random generation.

The pipeline, not the LLM itself, does the heavy lifting: mathematical auditing, pseudocode-to-code translation with review, parameter optimization, and simulated peer review filter LLM proposals into calibrated economic theories.

The approach applies to any empirical puzzle with well-defined quantitative targets. As LLMs improve and evaluation pipelines mature, automated theory search can expand the set of candidate explanations that receive serious scrutiny.

References

Abaluck, Jason, Robert Pless, Nirmal Ravi, Anja Sautmann, and Aaron Schwartz (Jan. 2026). *Does LLM Assistance Improve Healthcare Delivery? An Evaluation Using On-site Physicians and Laboratory Tests*. DOI: 10.3386/w34660. National Bureau of Economic Research: 34660. URL: https://www.nber.org/papers/w34660 (visited on 03/13/2026). Pre-published.

Amabile, Teresa M. (1983). "The Social Psychology of Creativity: A Componential Conceptualization". In: *Journal of Personality and Social Psychology* 45.2, pp. 357–376. ISSN: 1939-1315. DOI: 10.1037/0022-3514.45.2.357.

Asirvatham, Hemanth, Elliott Mokski, and Andrei Shleifer (Feb. 2026). *GPT as a Measurement Tool*. DOI: 10.3386/w34834. National Bureau of Economic Research: 34834. URL: https://www.nber.org/papers/w34834 (visited on 03/13/2026). Pre-published.

Bansal, Ravi, Shane Miller, Dongho Song, and Amir Yaron (Dec. 1, 2021). "The Term Structure of Equity Risk Premia". In: *Journal of Financial Economics* 142.3, pp. 1209–1228. ISSN: 0304-405X. DOI: 10.1016/j.jfineco.2021.05.043. URL: https://www.sciencedirect.com/science/article/pii/S0304405X21002361 (visited on 03/13/2026).

Bansal, Ravi and Amir Yaron (2004). "Risks for the Long Run: A Potential Resolution of Asset Pricing Puzzles". In: *The Journal of Finance* 59.4, pp. 1481–1509. ISSN: 1540-6261. DOI: 10.1111/j.1540-6261.2004.00670.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-6261.2004.00670.x (visited on 03/13/2026).

Barro, Robert J. (Aug. 1, 2006). "Rare Disasters and Asset Markets in the Twentieth Century*". In: *The Quarterly Journal of Economics* 121.3, pp. 823–866. ISSN: 0033-5533. DOI: 10.1162/qjec.121.3.823. URL: https://doi.org/10.1162/qjec.121.3.823 (visited on 03/13/2026).

Bini, Pietro, Lin William Cong, Xing Huang, and Lawrence J. Jin (Jan. 2026). *Behavioral Economics of AI: LLM Biases and Corrections*. DOI: 10.3386/w34745. National Bureau of Economic Research: 34745. URL: https://www.nber.org/papers/w34745 (visited on 03/13/2026). Pre-published.

Boiko, Daniil A., Robert MacKnight, Ben Kline, and Gabe Gomes (Dec. 2023). "Autonomous Chemical Research with Large Language Models". In: *Nature* 624.7992, pp. 570–578. ISSN: 1476-4687. DOI: 10.1038/s41586-023-06792-0. URL: https://www.nature.com/articles/s41586-023-06792-0 (visited on 03/13/2026).

Campbell, Donald T. (1960). "Blind Variation and Selective Retentions in Creative Thought as in Other Knowledge Processes". In: *Psychological Review* 67.6, pp. 380–400. ISSN: 1939-1471. DOI: 10.1037/h0040373.

Campbell, John Y. and John H. Cochrane (Apr. 1999). "By Force of Habit: A Consumption-Based Explanation of Aggregate Stock Market Behavior". In: *Journal of Political Economy* 107.2, pp. 205–251. ISSN: 0022-3808. DOI: 10.1086/250059. URL: https://www.journals.uchicago.edu/doi/10.1086/250059 (visited on 03/13/2026).

Chatterji, Aaron, Thomas Cunningham, David J. Deming, Zoe Hitzig, Christopher Ong, Carl Yan Shan, and Kevin Wadman (Sept. 2025). *How People Use ChatGPT*. DOI: 10.3386/w34255. National Bureau of Economic Research: 34255. URL: https://www.nber.org/papers/w34255 (visited on 03/13/2026). Pre-published.

Fleming, Lee (Jan. 2001). "Recombinant Uncertainty in Technological Search". In: *Management Science* 47.1, pp. 117–132. ISSN: 0025-1909. DOI: 10.1287/mnsc.47.1.117.10671. URL: https://pubsonline.informs.org/doi/10.1287/mnsc.47.1.117.10671 (visited on 03/13/2026).

Fortunato, Santo, Carl T. Bergstrom, Katy Börner, James A. Evans, Dirk Helbing, Staša Milojević, Alexander M. Petersen, Filippo Radicchi, Roberta Sinatra, Brian Uzzi, Alessandro Vespignani, Ludo Waltman, Dashun Wang, and Albert-László Barabási (Mar. 2, 2018). "Science of Science". In: *Science* 359.6379, eaao0185. DOI: 10.1126/science.aao0185. URL: https://www.science.org/doi/10.1126/science.aao0185 (visited on 03/13/2026).

Foster, Jacob G., Andrey Rzhetsky, and James A. Evans (Oct. 1, 2015). "Tradition and Innovation in Scientists' Research Strategies". In: *American Sociological Review* 80.5, pp. 875–908. ISSN: 0003-1224. DOI: 10.1177/0003122415601618. URL: https://doi.org/10.1177/0003122415601618 (visited on 03/13/2026).

Gabaix, Xavier and Ralph S. J. Koijen (Dec. 23, 2023). *In Search of the Origins of Financial Fluctuations: The Inelastic Markets Hypothesis*. DOI: 10.2139/ssrn.3686935. Social Science Research Network: 3686935. URL: https://papers.ssrn.com/abstract=3686935 (visited on 03/13/2026). Pre-published.

Galiani, Sebastian, Ramiro H. Gálvez, Franco Mettola La Giglia, and Raul A. Sosa (Jan. 2026). *Measuring Efficiency and Equity Framing in Economics Research: LLM-Based Evidence from 1950 to 2021*. DOI: 10.3386/w34714. National Bureau of Economic Research: 34714. URL: https://www.nber.org/papers/w34714 (visited on 03/13/2026). Pre-published.

Giglio, Stefano, Dacheng Xiu, and Dake Zhang (2025). "Test Assets and Weak Factors". In: *The Journal of Finance* 80.1, pp. 259–319. ISSN: 1540-6261. DOI: 10.1111/jofi.13415. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/jofi.13415 (visited on 03/13/2026).

Glosten, Lawrence R. and Paul R. Milgrom (Mar. 1, 1985). "Bid, Ask and Transaction Prices in a Specialist Market with Heterogeneously Informed Traders". In: *Journal of Financial Economics* 14.1, pp. 71–100. ISSN: 0304-405X. DOI: 10.1016/0304-405X(85)90044-3. URL: https://www.sciencedirect.com/science/article/pii/0304405X85900443 (visited on 03/13/2026).

Goetzmann, William N., Dasol Kim, and Robert J. Shiller (June 2024). *Emotions and Subjective Crash Beliefs*. DOI: 10.3386/w32589. National Bureau of Economic Research: 32589. URL: https://www.nber.org/papers/w32589 (visited on 03/13/2026). Pre-published.

Hansen, Stephen, Peter John Lambert, Nicholas Bloom, Steven J. Davis, Raffaella Sadun, and Bledi Taska (Mar. 2023). *Remote Work across Jobs, Companies, and Space*. DOI: 10.3386/w31007. National Bureau of Economic Research: 31007. URL: https://www.nber.org/papers/w31007 (visited on 03/13/2026). Pre-published.

Hasbrouck, Joel (2009). "Trading Costs and Returns for U.S. Equities: Estimating Effective Costs from Daily Data". In: *The Journal of Finance* 64.3, pp. 1445–1477. ISSN: 1540-6261. DOI: 10.1111/j.1540-6261.2009.01469.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-6261.2009.01469.x (visited on 03/13/2026).

Heckman, James J. and Burton Singer (May 2017). "Abducting Economics". In: *American Economic Review* 107.5, pp. 298–302. ISSN: 0002-8282. DOI: 10.1257/aer.p20171118. URL: https://www.aeaweb.org/articles?id=10.1257/aer.p20171118 (visited on 03/13/2026).

Horton, John J. (2023). "Large Language Models as Simulated Economic Agents: What Can We Learn from Homo Silicus?" In: *SSRN Electronic Journal*. ISSN: 1556-5068. DOI: 10.2139/ssrn.4413859. URL: http://dx.doi.org/10.2139/ssrn.4413859.

Hubert, Thomas, Rishi Mehta, Laurent Sartran, Miklós Z. Horváth, Goran Žužić, Eric Wieser, Aja Huang, Julian Schrittwieser, Yannick Schroecker, Hussain Masoom, Ottavia Bertolli, Tom Zahavy, Amol Mandhane, Jessica Yung, Iuliya Beloshapka, Borja Ibarz, Vivek Veeriah, Lei Yu, Oliver Nash, Paul Lezeau, Salvatore Mercuri, Calle Sönne,

Bhavik Mehta, Alex Davies, Daniel Zheng, Fabian Pedregosa, Yin Li, Ingrid von Glehn, Mark Rowland, Samuel Albanie, Ameya Velingker, Simon Schmitt, Edward Lockhart, Edward Hughes, Henryk Michalewski, Nicolas Sonnerat, Demis Hassabis, Pushmeet Kohli, and David Silver (Nov. 12, 2025). "Olympiad-Level Formal Mathematical Reasoning with Reinforcement Learning". In: *Nature*, pp. 1–3. ISSN: 1476-4687. DOI: 10.1038/s41586-025-09833-y. URL: https://www.nature.com/articles/s41586-025-09833-y (visited on 03/13/2026).

Jabarian, Brian and Alex Imas (Sept. 2025). *Artificial Writing and Automated Detection*. DOI: 10.3386/w34223. National Bureau of Economic Research: 34223. URL: https://www.nber.org/papers/w34223 (visited on 03/13/2026). Pre-published.

Jumper, John, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis (Aug. 2021). "Highly Accurate Protein Structure Prediction with AlphaFold". In: *Nature* 596.7873, pp. 583–589. ISSN: 1476-4687. DOI: 10.1038/s41586-021-03819-2. URL: https://www.nature.com/articles/s41586-021-03819-2 (visited on 03/13/2026).

Korinek, Anton (Feb. 2023). *Language Models and Cognitive Automation for Economic Research*. DOI: 10.3386/w30957. National Bureau of Economic Research: 30957. URL: https://www.nber.org/papers/w30957 (visited on 03/13/2026). Pre-published.

— (Nov. 2024). *Generative AI for Economic Research: LLMs Learn to Collaborate and Reason*. DOI: 10.3386/w33198. National Bureau of Economic Research: 33198. URL: https://www.nber.org/papers/w33198 (visited on 12/26/2024). Pre-published.

Korinek, Anton (Sept. 2025). *AI Agents for Economic Research*. DOI: 10.3386/w34202. National Bureau of Economic Research: 34202. URL: https://www.nber.org/papers/w34202 (visited on 03/13/2026). Pre-published.

Kyle, Albert S. (Nov. 1985). "Continuous Auctions and Insider Trading". In: *Econometrica* 53.6, p. 1315. ISSN: 00129682. DOI: 10.2307/1913210.

Li, Jiacui and Zihan Lin (June 16, 2022). *Price Multipliers Are Larger for Less Diversifiable Order Flows*. DOI: 10.2139/ssrn.4038664. Social Science Research Network: 4038664. URL: https://papers.ssrn.com/abstract=4038664 (visited on 03/13/2026). Pre-published.

Ludwig, Jens and Sendhil Mullainathan (May 1, 2024). "Machine Learning as a Tool for Hypothesis Generation*". In: *The Quarterly Journal of Economics* 139.2, pp. 751–827. ISSN: 0033-5533. DOI: 10.1093/qje/qjad055. URL: https://doi.org/10.1093/qje/qjad055 (visited on 03/13/2026).

Ludwig, Jens, Sendhil Mullainathan, and Ashesh Rambachan (Jan. 2025). *Large Language Models: An Applied Econometric Framework*. DOI: 10.3386/w33344. National Bureau of Economic Research: 33344. URL: https://www.nber.org/papers/w33344 (visited on 04/07/2025). Pre-published.

Luo, Xiaoliang, Akilles Rechardt, Guangzhi Sun, Kevin K. Nejad, Felipe Yáñez, Bati Yilmaz, Kangjoo Lee, Alexandra O. Cohen, Valentina Borghesani, Anton Pashkov, Daniele Marinazzo, Jonathan Nicholas, Alessandro Salatiello, Ilia Sucholutsky, Pasquale Minervini, Sepehr Razavi, Roberta Rocca, Elkhan Yusifov, Tereza Okalova, Nianlong Gu, Martin Ferianc, Mikail Khona, Kaustubh R. Patil, Pui-Shee Lee, Rui Mata, Nicholas E. Myers, Jennifer K. Bizley, Sebastian Musslick, Isil Poyraz Bilgin, Guiomar Niso, Justin M. Ales, Michael Gaebler, N. Apurva Ratan Murty, Leyla Loued-Khenissi, Anna Behler, Chloe M. Hall, Jessica Dafflon, Sherry Dongqi Bao, and Bradley C. Love (Feb. 2025). "Large Language Models Surpass Human Experts in Predicting Neuroscience Results". In: *Nature Human Behaviour* 9.2, pp. 305–315. ISSN: 2397-3374. DOI: 10.1038/

s41562-024-02046-9. URL: https://www.nature.com/articles/s41562-024-02046-9 (visited on 03/13/2026).

Nerkar, Atul (Feb. 2003). "Old Is Gold? The Value of Temporal Exploration in the Creation of New Knowledge". In: *Management Science* 49.2, pp. 211–229. ISSN: 0025-1909. DOI: 10.1287/mnsc.49.2.211.12747. URL: https://pubsonline.informs.org/doi/10.1287/mnsc.49.2.211.12747 (visited on 03/13/2026).

Novikov, Alexander, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco J. R. Ruiz, Abbas Mehrabian, M. Pawan Kumar, Abigail See, Swarat Chaudhuri, George Holland, Alex Davies, Sebastian Nowozin, Pushmeet Kohli, and Matej Balog (June 16, 2025). *AlphaEvolve: A Coding Agent for Scientific and Algorithmic Discovery*. DOI: 10.48550/arXiv.2506.13131. arXiv: 2506.13131 [cs]. URL: http://arxiv.org/abs/2506.13131 (visited on 03/13/2026). Pre-published.

Novy-Marx, Robert and Mihail Velikov (Mar. 2026). "Artificial Intelligence–Powered (Finance) Scholarship". In: *Journal of Economic Literature* 64.1, pp. 5–37. ISSN: 0022-0515. DOI: 10.1257/jel.20251821. URL: https://www.aeaweb.org/articles?id=10.1257/jel.20251821 (visited on 03/13/2026).

OpenAI et al. (Aug. 8, 2025). *Gpt-Oss-120b & Gpt-Oss-20b Model Card*. DOI: 10.48550/arXiv.2508.10925. arXiv: 2508.10925 [cs]. URL: http://arxiv.org/abs/2508.10925 (visited on 03/13/2026). Pre-published.

Ottonello, Pablo, Wenting Song, and Sebastian Sotelo (Sept. 2024). *An Anatomy of Firms' Political Speech*. DOI: 10.3386/w32923. National Bureau of Economic Research: 32923. URL: https://www.nber.org/papers/w32923 (visited on 03/13/2026). Pre-published.

Park, Michael, Erin Leahey, and Russell J. Funk (Jan. 2023). "Papers and Patents Are Becoming Less Disruptive over Time". In: *Nature* 613.7942, pp. 138–144. ISSN: 1476-4687. DOI: 10.1038/s41586-022-05543-x. URL: https://www.nature.com/articles/s41586-022-05543-x (visited on 03/13/2026).

Romera-Paredes, Bernardino, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli, and Alhussein Fawzi (Jan. 2024). "Mathematical Discoveries from Program Search with Large Language Models". In: *Nature* 625.7995, pp. 468–475. ISSN: 1476-4687. DOI: 10.1038/s41586-023-06924-6. URL: https://www.nature.com/articles/s41586-023-06924-6 (visited on 03/13/2026).

Runco, Mark A. and Garrett J. Jaeger (2012). "The Standard Definition of Creativity". In: *Creativity Research Journal* 24.1, pp. 92–96. ISSN: 1532-6934. DOI: 10.1080/10400419.2012.650092.

Simonton, Dean Keith (1997). "Creative Productivity: A Predictive and Explanatory Model of Career Trajectories and Landmarks". In: *Psychological Review* 104.1, pp. 66–89. ISSN: 1939-1471. DOI: 10.1037/0033-295X.104.1.66.

Teplitskiy, Misha, Hao Peng, Andrea Blasco, and Karim R. Lakhani (Nov. 22, 2022). "Is Novel Research Worth Doing? Evidence from Peer Review at 49 Journals". In: *Proceedings of the National Academy of Sciences* 119.47, e2118046119. DOI: 10.1073/pnas.2118046119. URL: https://www.pnas.org/doi/10.1073/pnas.2118046119 (visited on 03/13/2026).

Tranchero, Matteo, Cecil-Francis Brenninkmeijer, Arul Murugan, and Abhishek Nagaraj (Oct. 2024). *Theorizing with Large Language Models*. DOI: 10.3386/w33033. National Bureau of Economic Research: 33033. URL: https://www.nber.org/papers/w33033 (visited on 12/26/2024). Pre-published.

Trinh, Trieu H., Yuhuai Wu, Quoc V. Le, He He, and Thang Luong (Jan. 2024). "Solving Olympiad Geometry without Human Demonstrations". In: *Nature* 625.7995, pp. 476–482. ISSN: 1476-4687. DOI: 10.1038/s41586-023-06747-5. URL: https://www.nature.com/articles/s41586-023-06747-5 (visited on 03/13/2026).

Uzzi, Brian, Satyam Mukherjee, Michael Stringer, and Ben Jones (Oct. 25, 2013). "Atypical Combinations and Scientific Impact". In: *Science* 342.6157, pp. 468–472. DOI: 10.1126/science.1240474. URL: https://www.science.org/doi/10.1126/science.1240474 (visited on 03/13/2026).

Van Binsbergen, Jules, Michael Brandt, and Ralph Koijen (June 2012). "On the Timing and Pricing of Dividends". In: *American Economic Review* 102.4, pp. 1596–1618. ISSN: 0002-8282. DOI: 10.1257/aer.102.4.1596. URL: https://www.aeaweb.org/articles?id=10.1257/aer.102.4.1596 (visited on 03/13/2026).

Wang, Jian, Reinhilde Veugelers, and Paula Stephan (Oct. 1, 2017). "Bias against Novelty in Science: A Cautionary Tale for Users of Bibliometric Indicators". In: *Research Policy* 46.8, pp. 1416–1436. ISSN: 0048-7333. DOI: 10.1016/j.respol.2017.06.006. URL: https://www.sciencedirect.com/science/article/pii/S0048733317301038 (visited on 03/13/2026).

Wu, Jing Cynthia, Jin Xi, and Shihan Shihan. Xie@gmail.com (Oct. 1, 2025). *LLM Survey Framework: Coverage, Reasoning, Dynamics, Identification*. Social Science Research Network: 5665312. URL: https://papers.ssrn.com/abstract=5665312 (visited on 03/13/2026). Pre-published.

Yu, Yulin and Daniel M. Romero (Oct. 8, 2024). "Does the Use of Unusual Combinations of Datasets Contribute to Greater Scientific Impact?" In: *Proceedings of the National Academy of Sciences* 121.41, e2402802121. DOI: 10.1073/pnas.2402802121. URL: https://www.pnas.org/doi/10.1073/pnas.2402802121 (visited on 03/13/2026).

Appendices

A Prompts Used in the System

The system uses the following prompts for theory generation.

A.1 Problem analyzer prompts

System prompt

```
You are an expert theoretical economist analyzing empirical puzzles.
Your task is to deeply understand why standard models fail to explain observed data, and identify promising directions for new theoretical explanations.
THINK REALLY HARD. This is a genuine research problem that has stumped economists.
Take your time to reason through the mechanics carefully.
Think like a researcher preparing to write a theory paper:
- What are the key facts that need explaining?
- Why EXACTLY do existing models fail? Be mechanistic, not hand-wavy.
- What assumptions might be wrong? Which ones matter most?
- What novel mechanisms could explain the data? Be specific about functional forms.
Be specific and analytical. Avoid vague suggestions like "add frictions" - instead say
exactly what kind of friction and how it would affect the equilibrium.
```

- **Analysis prompt template:** The template below is filled with puzzle-specific content; placeholders `{puzzle_description}`, `{targets_description}`, `{baseline_description}`, and `{additional_context}` are substituted at runtime.

Analysis prompt template

```
# Task
Think really hard about this. Deeply analyze the following empirical
↪  puzzle.
Your analysis will guide the development of new theoretical models to
↪  explain the data.
This is a real research problem - your insights matter. Don't rush.
# The Puzzle {puzzle_description}
# Empirical Targets {targets_description}
# Baseline Model {baseline_description}
# Additional Context {additional_context}
# Your Analysis

Provide a thorough analysis addressing:
1. **Empirical Facts**: What are the 2-4 key facts that need
↪  explaining?
2. **Baseline Failure**: Why exactly does the baseline model fail? Be
↪  mechanistic.
3. **Violated Assumptions**: Which assumptions of the baseline are
↪  likely wrong?
4. **Promising Directions**: What are 3-5 theoretical directions worth
↪  exploring?
   - Be specific (not just "add frictions" but "transaction costs that
   ↪  scale with...")
   - Consider: behavioral biases, market frictions, heterogeneous
   ↪  agents,
     information asymmetry, institutional constraints, etc.
5. **Dead Ends**: What approaches are unlikely to work? Why?
```

```
6. **Key Constraints**: What must any successful theory satisfy?
   - Mathematical constraints
   - Economic plausibility requirements
   - Parsimony considerations
7. **Unconventional Idea**: Suggest one surprising or unconventional
↪  angle that
   might work but isn't obvious.
Respond in the structured JSON format.
```

A.2 Theory evolution prompts

System prompt

```
You are an expert theoretical economist specializing in asset pricing.
You propose novel theoretical models to explain empirical puzzles.
You think creatively but rigorously, and your models are mathematically
↪  precise.

CRITICAL: Parsimony is paramount. You are fitting N target moments.
↪  Models with more than N free parameters are overparameterized and
↪  will be heavily penalized. A simple model that approximately hits
↪  targets is far better than a complex model that fits exactly.

CRITICAL: Before elaborating your model, verify that your proposed
↪  mechanism can mathematically achieve the target moments with
↪  reasonable parameter values. A model that cannot hit the targets
↪  regardless of calibration will fail.
```

- **Random strategy:** Propose a completely new theory. Placeholders {`puzzle_context`}

and {previous_attempts} are filled at runtime.

Random strategy

```
# Task
Propose a completely novel theoretical model to explain the empirical
↪  puzzle.
---
{puzzle_context}
---
# Previous Attempts
{previous_attempts}
---
# Your Task
Propose a NEW theoretical model that matches the empirical targets and
↪  is DIFFERENT from previous attempts.
Provide your response as JSON with these fields:
- model_name: Short descriptive name (e.g., "Risk-Information Hybrid
↪  Model")
- key_insight: 1-2 sentence core insight
- formulas: Dictionary mapping each output name to its mathematical
↪  formula
- parameters: List of parameters with economic interpretation (fewer is
↪  better)
- mechanism_explanation: Why this mechanism produces target values
- full_proposal: Complete detailed proposal with all derivations
```

- **Mutate strategy:** Modify a successful parent model. Placeholders include {parent_name}, {parent_insight}, {parent_formulas}, {parent_params}, {parent_score}, {parent_results}, and referee feedback fields.

Mutate strategy

```
# Task

Take this successful model and propose a VARIATION that might work even
↪  better.

---

# Parent Model: {parent_name}

## Key Insight

{parent_insight}

## Formulas

{parent_formulas}

## Parameters ({n_params} free parameters)

{parent_params}

## Results

- Score: {parent_score}/100

{parent_results}

## Referee Feedback

**Strengths:** {referee_strengths}

**Weaknesses:** {referee_weaknesses}

**Suggestions:** {referee_suggestions}

---

# Mutation Instructions

Propose a modification to this model. Options (in order of preference):

1. **Simplify** - Remove a parameter or combine terms (STRONGLY
↪  PREFERRED - improves parsimony)
2. **Modify mechanism** - Change how a parameter enters the model
3. **Add interaction** - Combine existing parameters in a new way
4. **Add new effect** - Introduce a new economic friction (DISCOURAGED
↪  unless essential)
```

```
The referee noted weaknesses above. Your mutation should address at
↪  least one.
The mutation should:
- Keep the core insight but modify the mechanism
- Be economically plausible
- Address referee concerns
- Improve parsimony if possible
---
# Your Task
Provide your response as JSON with these fields:
- model_name: Short descriptive name for the mutated model
- key_insight: 1-2 sentence core insight (what changed and why)
- formulas: Dictionary with new mathematical formulas for each output
- parameters: List of parameters (note which are new/removed)
- mechanism_explanation: Why this mutation improves the model
- full_proposal: Complete detailed proposal with derivations
```

- **Crossover strategy:** Combine two models into a hybrid. Placeholders `{model_a_name}`, `{model_a_insight}`, `{model_a_formulas}`, `{model_a_score}`, and similarly for model B, are substituted at runtime.

Crossover strategy

```
# Task
Combine ideas from these two models to create a hybrid.
---
# Model A: {model_a_name}
## Key Insight
{model_a_insight}
```

```
## Formulas

{model_a_formulas}

## Score: {model_a_score}/100

---

# Model B: {model_b_name}

## Key Insight

{model_b_insight}

## Formulas

{model_b_formulas}

## Score: {model_b_score}/100

---

# Your Task

Create a hybrid model combining the best elements of both. AIM FOR
↪  FEWER total parameters than the sum - find overlap!

Provide your response as JSON with these fields:

- model_name: Short descriptive name for the hybrid model

- key_insight: 1-2 sentence core insight (what elements are combined)

- formulas: Dictionary with new mathematical formulas for each output

- parameters: Combined parameter list (aim for fewer than sum of both)

- mechanism_explanation: Why this combination works better than either
↪  alone

- full_proposal: Complete detailed proposal with derivations
```

The *mutate* prompt includes the parent model's full referee report (strengths, weaknesses, suggestions), so mutation incorporates referee-driven refinement directly. There is no separate "refine" strategy; referee feedback is integrated into the mutate operator.

A.3 Coding agent prompts

System prompt

```
You are an expert Python programmer specializing in mathematical
↪ modeling.
Your task is to translate theoretical economic models into executable
↪ Python code.

IMPORTANT RULES:
1. Implement the formulas EXACTLY as written in the proposal
2. Do NOT use any specific numerical values claimed by the proposer
3. All parameters should be inputs to the function, not hardcoded
4. Include clear comments mapping code to the mathematical formulas
5. Handle edge cases (division by zero, negative values, etc.)
6. Do NOT use "from __future__ import annotations" - use standard type
↪ hints

You write clean, well-documented code with type hints.
```

- **User prompt template:** The placeholder `{model_proposal}` is substituted with the theoretical proposal at runtime.

User prompt template

```
# Task
Translate the following theoretical model into executable Python code.
---
# Model Proposal
{model_proposal}
---
```

````
# Required Interface

You MUST implement this exact interface:

```python
from dataclasses import dataclass
from typing import Dict, Any


@dataclass
class ModelParams:
    """Parameters for the model."""
    # Standard calibration parameters (from the paper)
    N: int = 2278           # total number of stocks
    S: int = 9              # number of style portfolios
    N_s: int = 253          # stocks per style (N/S)
    sigma_style: float = 0.104   # style shock volatility (annual)
    sigma_idio: float = 0.435    # idiosyncratic volatility (annual)
    sigma_market: float = 0.16   # market volatility (annual)
    # ADD ANY MODEL-SPECIFIC PARAMETERS HERE with sensible defaults


def compute_multipliers(params: ModelParams) -> Dict[str, Any]:
    """
    Compute price multipliers M_style and M_idio for this model.
    Returns:
        Dictionary with keys: M_style, M_idio, ratio, intermediates
    """
    # YOUR IMPLEMENTATION HERE
    pass
```

---
````

```
# Instructions
1. Add any new parameters from the model proposal to ModelParams
2. Implement compute_multipliers() based on the mathematical formulas
3. Include comments showing which formula each line implements
4. Return intermediate values for debugging
5. Do NOT hardcode any numbers - use params.*
---
# Output Format
Return ONLY valid Python code. No explanation before or after.
The code must be directly executable.
```python
# Your complete implementation here
```
```

A.4 Calibration agent prompts

System prompt

- You are an expert econometrician who calibrates theoretical models.

```
Given a theoretical model proposal and its Python implementation, you
↪  determine:
1. Sensible default values for each parameter
2. Valid bounds for optimization (min, max)
3. Test values for robustness checking
4. Parameter types (continuous, count, fraction)

You understand economic constraints:
- Risk aversion (gamma) is typically 1-10
```

```
- Counts (K, n_stocks) must be >= 1 and often <= N (total stocks)
- Fractions (delta, phi, eta) must be in [0, 1]
- Costs are typically small positive numbers

Be conservative with bounds - too wide is better than too narrow.
```

- **Calibration prompt template:** Placeholders `{proposal}` and `{code}` are substituted with the model proposal and generated Python code at runtime.

Calibration prompt template

````
# Task
Analyze this model and specify calibration for each parameter.
---
# Model Proposal
{proposal}
---
# Generated Code
```python
{code}
```
---
# Fixed Calibration Parameters (from the literature)
These are FIXED and should NOT be included in your output:
- N = 2278 (total stocks)
- S = 9 (style portfolios)
- N_s = 253 (stocks per style)
- sigma_style = 0.104 (style volatility)
- sigma_idio = 0.435 (idiosyncratic volatility)
````

```
- sigma_market = 0.16 (market volatility)

---

# Your Task

For each MODEL-SPECIFIC parameter (NOT the fixed ones above), provide:

1. name: exact name from the code

2. default: sensible starting value

3. min_bound, max_bound: valid range for optimization

4. test_values: 3-5 values spanning the range (for robustness testing)

5. param_type: "continuous", "count", or "fraction"

6. description: economic interpretation


IMPORTANT:

- Only include parameters that appear in ModelParams (not the fixed
  ↪ calibration ones)

- Bounds should be economically sensible (e.g., K <= N, fractions in
  ↪ [0,1])

- Test values should include edge cases where appropriate

- If a parameter represents a count, min_bound should be >= 1
```

A.5 Mathematical auditor prompts

The mathematical auditor verifies derivation correctness before code translation.

Audit system prompt

```
• You are a mathematical economist who audits derivations for
  ↪ correctness.

  Your job is to check every step of a theoretical proposal. Be rigorous
  ↪ and precise.
```

```
You must verify:
1. Each equation follows from the previous one (no skipped steps)
2. Matrix algebra is correct (dimensions, inverses, transposes)
3. First-order conditions are correctly derived from the objective
4. Market clearing conditions are properly imposed
5. Final pricing formulas actually follow from the derivation

Do NOT check calibrated numerical values - those are irrelevant.
Focus ONLY on whether the mathematical logic is sound.
```

- **Audit prompt:** The placeholder `{proposal}` is substituted with the model proposal. The auditor must end its response with either `OVERALL: PASS` or `OVERALL: FAIL`.

Fix system prompt

```
• You are a theoretical economist who fixes mathematical errors in model
  ↪ derivations.
  You will receive a proposal and a mathematical audit identifying
  ↪ errors.
  Fix ONLY the errors while preserving the economic mechanism and model
  ↪ structure.
  Output the complete corrected proposal in the same format as the
  ↪ original.
  Do NOT change the economic mechanism - only fix the math.
```

- **Fix prompt:** Receives the original proposal and the audit feedback. The fix-and-reaudit cycle repeats up to two times.

A.6 Pseudocode agent prompts

The pseudocode agent translates audited proposals into structured pseudocode.

System prompt

```
• You are a theoretical economist writing structured pseudocode for asset
  ↪ pricing models.
  Your job is to translate a model proposal into STRUCTURED PSEUDOCODE
  ↪ that a programmer
  can implement exactly. The pseudocode must capture the complete
  ↪ derivation chain from
  economic primitives to numerical outputs.

  CRITICAL RULES:
  1. Every output must be DERIVED from the model's economic equations
  2. No fudge factors, scaling parameters, or tuning knobs that directly
  ↪ encode target values
  3. All returns, risk premia, Sharpe ratios, alphas, and betas must
  ↪ emerge from the
     model's pricing kernel / stochastic discount factor (SDF)
  4. Conditional moments must arise from the model's state variable
  ↪ dynamics
  5. Be FAITHFUL to the proposal - include every equation and mechanism
  ↪ mentioned
  6. Be COMPLETE - every target output must have a clear derivation path

  The pseudocode must make the full chain visible:
    Economic primitives -> SDF/pricing kernel -> Asset prices -> Returns
    ↪ -> Target outputs
```

- **User prompt:** Receives the proposal and a list of target outputs. The pseudocode must be organized into four sections: `PARAMETERS`, `DERIVATION`, `COMPUTE OUTPUTS`, and `CONDITIONAL MOMENTS`.

A.7 Pseudocode reviewer prompts

The pseudocode reviewer verifies that pseudocode faithfully captures the proposal.

System prompt

```
• You are reviewing pseudocode against a theoretical model proposal.
  Your job is to verify that the pseudocode faithfully captures the
  ↪ model's economic
  mechanism and derivation chain.

  PASS the pseudocode if: the core economic mechanism is preserved, all
  ↪ key equations
  appear in the derivation, all target outputs are derived from model
  ↪ equations.

  FAIL the pseudocode if: fudge parameters scale outputs toward targets;
  ↪ outputs are
  stated but not derived from the SDF; the pricing kernel does not match
  ↪ the proposal;
  the core economic channel is absent; conditional moments are assumed
  ↪ rather than derived.

  Key principle: Every numerical output must be traceable back to the SDF
  ↪ and model
  parameters through a chain of economic equations.
```

- **Review prompt:** Receives the original proposal and the generated pseudocode. Returns a PASS/FAIL verdict with a list of issues and a one-sentence summary. If the review fails, the pseudocode agent regenerates with the reviewer's feedback (up to two retries).

A.8 Code-vs-pseudocode reviewer prompts

The code-vs-pseudocode reviewer verifies that generated Python code implements the pseudocode specification.

System prompt

```
• You are reviewing Python code against its pseudocode specification
  for a theoretical asset pricing model.

  Check if the code faithfully implements the pseudocode:
  1. PARAMETERS: Are all pseudocode parameters present as function
  ↪  arguments?
  2. DERIVATION: Does the code implement the SDF / pricing kernel
  ↪  equations?
  3. OUTPUTS: Does the code compute each target output using the derived
  ↪  formulas?
  4. CONDITIONAL MOMENTS: If the pseudocode specifies state-dependent
  ↪  computation,
     does the code implement it?

  PASS the code if the core formulas match the pseudocode.
  FAIL the code only for CRITICAL mismatches: extra fudge parameters,
  ↪  wrong formulas,
  outputs hardcoded rather than derived, or missing key derivation steps.
```

```
Key principle: The code must compute outputs through the same
↪  derivation chain as
the pseudocode. If an output bypasses the SDF, FAIL.
```

- **Review prompt:** Receives the pseudocode specification and the generated Python code. Returns a PASS/FAIL verdict with issues and summary. If the review fails, the coding agent regenerates with feedback.

A.9 Evaluator: robustness check and optimization

The evaluator runs in Python (no LLM) and performs two steps.

Robustness check. The calibration agent supplies test values for each model-specific parameter. The evaluator builds a list of parameter combinations (e.g., default, plus each test value varied in turn) and runs `compute_multipliers(params)` for each. The check passes only if every run completes without exception and all outputs are finite. This check catches division-by-zero, invalid domains, and NaN/Inf.

Parameter optimization. The objective is to minimize the sum of relative squared errors to the empirical targets. Each puzzle defines its own outputs and targets. For the price-multiplier puzzle, the outputs are M_{style}, M_{idio}, and the ratio $M_{\text{style}}/M_{\text{idio}}$. For the dividend strip puzzle, all ten outputs are included. In general, for outputs $y_1, \ldots, y_k$ with targets $t_1, \ldots, t_k$, the objective is

$$L(x) = \sum_{i:t_i \neq 0} \left(\frac{y_i(x) - t_i}{t_i} \right)^2, \tag{43}$$

where x is the vector of model-specific parameters. All outputs receive equal weight. The scorer uses the same (equal-weighted) average percentage error when computing the fit subscore, so the objective and fit score are consistent. The evaluator uses `scipy.optimize.differential`

with bounds from the calibration agent (or defaults such as $\gamma \in [0.1, 10]$, fractions in $[0, 1]$), with `maxiter=100`, `tol=1e-6`, `polish=True`, and `seed=42`. The optimizer stage is intentionally lenient: a run is discarded only if the solver fails to converge *and* the final loss exceeds $L > 5.0$, which corresponds to very large errors. All other models, including those with moderate fit, proceed to the scorer and referee, which judge quality on a continuous scale. The optimal parameters and resulting outputs (e.g., M_{style}, M_{idio}, ratio) are returned for the scorer and referee.

A.10 Scorer: formulas

The scorer combines three subscores with configurable weights. In the reported experiment we use $w_{\text{fit}} = 0.50$, $w_{\text{plaus}} = 0.25$, $w_{\text{pars}} = 0.25$. Denote the weight for dimension d by w_d with $\sum_d w_d = 1$.

Fit. For each output i with value v_i and target $t_i \neq 0$, the percentage error is $e_i = 100 \cdot |v_i - t_i|/|t_i|$. The average error is the simple (equal-weighted) mean: $\bar{e} = \left(\sum_i e_i\right)/k$, where k is the number of outputs with non-zero targets. The fit score is

$$\text{Fit} = \min\bigl(100,\ \max\bigl(0,\ 100\, e^{-\bar{e}/20}\bigr)\bigr). \tag{44}$$

So perfect fit gives 100; score decays exponentially with average percentage error (e.g., $\bar{e} \approx 20\%$ gives about 37).

Plausibility. Each calibrated parameter is checked against predefined bounds (e.g., risk aversion $\gamma \in [0.5, 5]$, fractions in $[0, 1]$). Let n be the number of parameters checked and m the number within bounds. The base score is $(m/n) \times 100$. If any parameter is out of bounds, a penalty of up to 30 points is applied based on the maximum deviation; the final plausibility score is $\max(0, \text{base} - \text{penalty})$.

Parsimony. With n_{free} = number of free (model-specific) parameters,

$$\text{Parsimony} = \max\bigl(0,\ 100 - 10 \cdot n_{\text{free}}\bigr). \tag{45}$$

Each free parameter reduces the score by 10 points; models with 10 or more free parameters score zero.

Total. Total $= w_{\text{fit}} \cdot \text{Fit} + w_{\text{plaus}} \cdot \text{Plausibility} + w_{\text{pars}} \cdot \text{Parsimony}$, rounded to one decimal.

A.11 Referee prompts

System prompt

```
• You are a senior referee for a top economics journal.

  You evaluate theoretical model proposals that attempt to explain
  ↪ empirical puzzles.
  You judge based on:
  1. Empirical fit - does it match the data?
  2. Economic plausibility - does the mechanism make sense?
  3. Novelty - is this a new contribution?
  4. Parsimony - is it elegantly simple?
  5. Implementation quality - is the math correctly implemented?

  You provide constructive, specific feedback that would help improve the
  ↪ paper.
  You are fair but rigorous.
```

- **Referee prompt template:** Placeholders {proposal}, {scoring_section}, {quant_total}, {consistent}, {consistency_verdict}, {discrepancies}, and {optimization_results}

are filled with the model proposal, quantitative scores breakdown, weighted total, consistency result, and optimization summary. The referee returns structured JSON with: `model_name`, `score` (0–100), `recommendation` (reject / major_revision / minor_revision / accept), `summary`, `strengths`, `weaknesses`, `suggestions`, `novelty_assessment`, `economic_plausibility`, `reasoning`. Scoring guidelines: 90–100 excellent; 75–89 good, minor revisions; 60–74 promising, major revisions; 40–59 significant issues; 0–39 reject.

Referee prompt template

(Full template omitted; placeholders and required JSON schema as described in the preceding paragraph.)

B Derivation of the Benchmark Price Multiplier Ratio

This appendix derives from first principles the benchmark ratio $M_{\text{style}}/M_{\text{idio}} = N_s\sigma^2_{\text{style}}/\sigma^2_{\text{idio}}$. Under the calibrated inputs used in the main text, it equals 14.5.

B.1 Setup: CARA-Normal liquidity provider and the multiplier

Let $r \in \mathbb{R}^{N_s}$ denote the vector of (excess) returns of the N_s stocks in one style bucket, and assume $r \sim \mathcal{N}(\mu, \Sigma)$. A representative (marginal) risk-averse investor with CARA-Normal mean–variance preferences chooses a holdings vector $x \in \mathbb{R}^{N_s}$ to maximize

$$\mu^\top x - \frac{\gamma}{2} x^\top \Sigma x, \tag{46}$$

where $\gamma > 0$ is the coefficient of absolute risk aversion. The first-order condition is $\mu = \gamma\Sigma x$. Suppose an order-flow shock forces the market to absorb an exogenous net supply q (e.g., flow-induced inventory). Market clearing for the marginal absorber implies $x = q$,

so

$$\mu = \gamma \Sigma q. \tag{47}$$

Thus the expected return (risk premium) required for the marginal investor to hold q is proportional to the risk of that position. A "price multiplier" is the amount of expected return or discount required per unit of flow shock. In this one-period benchmark, the mapping from q to μ is linear with coefficient proportional to $\gamma\Sigma$; up to a common duration or price-elasticity constant that cancels in ratios, the multiplier is proportional to $\gamma\Sigma$.

B.2 Covariance structure: style and idiosyncratic components

Each stock's return is written as $r_i = s + \varepsilon_i$, where s is the style common shock with $\mathrm{Var}(s) = \sigma^2_{\text{style}}$, and ε_i is the idiosyncratic component with $\mathrm{Var}(\varepsilon_i) = \sigma^2_{\text{idio}}$. The ε_i are mutually independent and independent of s. The covariance matrix is therefore

$$\Sigma = \sigma^2_{\text{style}} \mathbf{1}\mathbf{1}^\top + \sigma^2_{\text{idio}} I, \tag{48}$$

where $\mathbf{1}$ is the $N_s \times 1$ vector of ones. Hence $\mathrm{Cov}(r_i, r_j) = \sigma^2_{\text{style}}$ for $i \neq j$, and $\mathrm{Var}(r_i) = \sigma^2_{\text{style}} + \sigma^2_{\text{idio}}$ for each i.

B.3 Idiosyncratic-shock multiplier M_{idio}

An idiosyncratic flow shock is a supply shock in a single stock: $q^{\text{idio}} = e_k$, where e_k is the unit vector with 1 in position k and 0 elsewhere. Substituting into (47) gives $\mu = \gamma \Sigma e_k$. The k-th component is $\mu_k = \gamma \Sigma_{kk} = \gamma(\sigma^2_{\text{style}} + \sigma^2_{\text{idio}})$. Following Li and Lin (2022), the idiosyncratic multiplier is defined to isolate the idiosyncratic variance contribution (the part due to ε_k), i.e.,

$$M_{\text{idio}} \equiv \gamma \sigma^2_{\text{idio}}. \tag{49}$$

B.4 Style-level shock multiplier M_{style}

A style-level flow shock is a supply shock that hits all N_s stocks in the same direction, e.g., one unit in each: $q^{\text{style}} = \mathbf{1}$. Substituting into (47) yields

$$\mu = \gamma \Sigma \mathbf{1} = \gamma \left(\sigma^2_{\text{style}} \mathbf{1}\mathbf{1}^\top \mathbf{1} + \sigma^2_{\text{idio}} I \mathbf{1} \right). \tag{50}$$

Since $\mathbf{1}^\top \mathbf{1} = N_s$, one has $\mathbf{1}\mathbf{1}^\top \mathbf{1} = N_s \mathbf{1}$, and $I\mathbf{1} = \mathbf{1}$. Thus

$$\mu = \gamma \left(N_s \sigma^2_{\text{style}} \mathbf{1} + \sigma^2_{\text{idio}} \mathbf{1} \right) = \gamma \left(N_s \sigma^2_{\text{style}} + \sigma^2_{\text{idio}} \right) \mathbf{1}, \tag{51}$$

so each stock's required premium is $\mu_i = \gamma(N_s \sigma^2_{\text{style}} + \sigma^2_{\text{idio}})$. Similarly, the style multiplier isolates the systematic scaling term:

$$M_{\text{style}} \equiv \gamma N_s \sigma^2_{\text{style}}. \tag{52}$$

B.5 The ratio

With these definitions,

$$\frac{M_{\text{style}}}{M_{\text{idio}}} = \frac{\gamma N_s \sigma^2_{\text{style}}}{\gamma \sigma^2_{\text{idio}}} = \frac{N_s \sigma^2_{\text{style}}}{\sigma^2_{\text{idio}}}. \tag{53}$$

B.6 Numerical value 14.5

Using the calibrated inputs $N_s = 253$, $\sigma_{\text{style}} = 0.104$, and $\sigma_{\text{idio}} = 0.435$, one obtains $\sigma^2_{\text{style}} = 0.010816$ and $\sigma^2_{\text{idio}} = 0.189225$, so

$$\frac{M_{\text{style}}}{M_{\text{idio}}} = \frac{253 \times 0.010816}{0.189225} = \frac{2.736448}{0.189225} \approx 14.5. \tag{54}$$

The factor N_s in the numerator arises from $\mathbf{1}\mathbf{1}^\top \mathbf{1} = N_s \mathbf{1}$: a correlated (style) position loads on the common shock in every stock, so the marginal investor's required premium

per stock scales with the breadth of the correlated inventory they must absorb.

C System Implementation Details

The system runs on Python 3.12. The `llm_service.py` module provides an LLM client wrapper with retry and exponential backoff. The `evolver.py` module implements evolution strategies (random, mutate, crossover) and the 20-persona system. The `math_auditor.py` module performs independent mathematical auditing with fix-and-reaudit. The `pseudocode_agent.py` module translates proposals to structured pseudocode. The `pseudocode_reviewer.py` module verifies pseudocode fidelity to the proposal. The `coding_agent.py` module handles code translation with nested retry. The `code_vs_pseudocode_reviewer.py` module verifies code fidelity to the pseudocode. The `calibration_agent.py` module extracts parameter bounds and defaults. The evaluator and scorer modules perform parameter optimization and quantitative scoring. The `referee.py` module provides simulated peer review. The `model_registry.py` module handles persistent storage of models. The `orchestrator.py` module coordinates the pipeline. The `run_parallel.py` module runs batch experiments with seed tracking. The `puzzles/` package provides a modular puzzle abstraction, with each puzzle defining its own targets, calibration inputs, and prompts.

C.1 Prompting Strategy

For theory generation, we provide the LLM with the empirical puzzle (target values, baseline model prediction), constraints (must be different from existing approaches), output format (key insight, mathematical formulas, parameters, intuition), and prior model examples and referee feedback for mutation strategies.

For code generation, we use a nested retry loop: 2 fresh starts, each with up to 3 fix attempts. Each attempt receives feedback on syntax errors (from `ast.parse`), robustness test failures (boundary conditions), and numerical issues (NaN, Inf values). After code

generation, the code-vs-pseudocode reviewer checks fidelity, with up to one retry if the review fails.

C.2 Strategy and parent selection

Once at least one model has completed the full pipeline, strategy probabilities are: random 35%, mutate 35%, crossover 30%. Before any model completes, every run uses the random strategy. Parents for mutate and crossover must have referee score ≥ 35; selection draws from the top 10 eligible models, weighted by referee score. The mutate prompt includes the parent's full referee report (strengths, weaknesses, suggestions), so mutation incorporates referee feedback directly. The experiment uses a fixed random seed per run so that strategy and parent selection are reproducible.

C.3 Retry Strategy

Table 15 summarizes the retry policy used by each component in the pipeline.

Representative top-scoring models are presented in full in Section 5.2 (BBC and DALD-UR for price multipliers) and Section 5.3 (IR-LCAPM and RS-LJCCAPM for dividend strips).

D Scoring Methodology

The quantitative score combines three dimensions with weights 0.50 (fit), 0.25 (plausibility), 0.25 (parsimony) in the reported experiment. See Appendix A.10 for the exact formulas. Fit measures how closely the model matches empirical targets (average percentage error, exponential decay). Plausibility evaluates parameter reasonableness against predefined bounds. Parsimony penalizes model complexity (fewer free parameters score higher).

Table 13: Economic validity assessment (top models by puzzle)

Puzzle	Model	Claimed Mechanism	Empirical Test Required
Price Mult.	Bayesian-Breadth-Cost CARA	The assumptions of CARA preferences, Gaussian returns, and a noisy public signal are standard and internally consistent; the quadratic cost captures a plausible friction that limits diversification, making the mechanism economically sensible.	Provide a more thorough discussion of the economic intuition behind the quadratic holding cost and its empirical relevance (e.g., link to transaction costs, limits to arbitrage, or portfolio-construction frictions).; Include out-of-sample or additional testable predictions (such as cross-sectional variation in multipliers across industries or time-varying information precision) to demonstrate the model's explanatory power beyond the calibrated moments.
Price Mult.	DALD-UR (Unified-Risk-Aversion)	Limiting the number of assets an institutional investor can hold is a realistic institutional friction, and the resulting separation of style and idiosyncratic risk premia is economically intuitive.	Explore extensions allowing heterogeneous risk-aversion coefficients across agents to assess robustness of the multiplier formulas.; Derive and empirically test additional predictions (e.g., cross-sectional variation of returns with respect to measured style exposure or time-variation of the breadth cap).
Div. Strips	Information-Risk Learning CAPM (IR-LCAPM)	The idea that poorer information in recessions raises the risk premium on short-dated dividend claims is intuitively appealing, but the required magnitude of risk-aversion and the near-zero market exposure (ω) are not economically credible.	Impose tighter priors on γ (e.g., 1–4) and μ_g (positive) and re-estimate; if the moments cannot be matched, consider alternative specifications for the information-risk shock.; Reduce dimensionality by linking the signal-noise variance to observable macro variables (e.g., recession indicators) or by fixing ω and p to values motivated by external evidence.
Div. Strips	Regime-Switching Liquidity-Jump CCAPM (RS-LJCCAPM)	Linking funding-tightness to higher dividend-cut risk is economically sensible, yet the calibrated intensities and jump magnitudes are far from empirically observed values, casting doubt on the quantitative plausibility.	Re-estimate the model using realistic priors (e.g., $\gamma_c \leq 10$, $\mu_J \in$ [-0.30,-0.05]) and impose identification restrictions to reduce the number of free parameters.; Demonstrate that the model can generate the recession-vs-expansion Sharpe-ratio differential; this may require a tighter link between the jump intensity and observable funding-tightness proxies.; Provide robustness checks (out-of-sample forecasts, alternative proxies for S_t) and report all moments (unconditional and conditional) with confidence intervals rather than nan or explosive values.

Notes: This table lists the claimed economic mechanism and a concrete empirical validation exercise for each top-scoring model from each puzzle. The right column states testable requirements, not results already established by this paper.

Table 14: Novelty assessment and referee suggestions (top models by puzzle)

Puzzle	Model	Novelty (referee)	Main Suggestions
Price Mult.	Bayesian-Breadth-Cost CARA	The model combines existing ideas—Bayesian learning about a common factor and quadratic portfolio-breadth costs—into a coherent closed-form solution. While the algebraic correction is valuable, the core mechanisms are not new, so the contribution is incremental rather than groundbreaking.	Provide a more thorough discussion of the economic intuition behind the quadratic holding cost and its empirical relevance (e.g., link to transaction costs, limits to arbitrage, or portfolio-construction frictions).; Include out-of-sample or additional testable predictions (such as cross-sectional variation in multipliers across industries or time-varying information precision) to demonstrate the model's explanatory power beyond the calibrated moments.
Price Mult.	DALD-UR (Unified-Risk-Aversion)	The paper offers a novel closed-form solution for price multipliers in a dual-agent CARA framework with a breadth constraint, which has not been presented in the literature before.	Explore extensions allowing heterogeneous risk-aversion coefficients across agents to assess robustness of the multiplier formulas.; Derive and empirically test additional predictions (e.g., cross-sectional variation of returns with respect to measured style exposure or time-variation of the breadth cap).
Div. Strips	Information-Risk Learning CAPM (IR-LCAPM)	The paper introduces a novel SDF component that prices a forecast-error (information-risk) shock with regime-dependent precision, which is a fresh angle relative to existing learning or disaster models.	Impose tighter priors on γ (e.g., 1–4) and μ_g (positive) and re-estimate; if the moments cannot be matched, consider alternative specifications for the information-risk shock.; Reduce dimensionality by linking the signal-noise variance to observable macro variables (e.g., recession indicators) or by fixing ω and p to values motivated by external evidence.
Div. Strips	Regime-Switching Liquidity-Jump CCAPM (RS-LJCCAPM)	The combination of a Markov-switching liquidity factor with a regime-dependent dividend-jump component is a novel extension of the LR-CCAPM literature, but similar ideas have appeared in recent regime-switching LRR models.	Re-estimate the model using realistic priors (e.g., $\gamma_c \leq 10$, $\mu_J \in$ [-0.30,-0.05]) and impose identification restrictions to reduce the number of free parameters.; Demonstrate that the model can generate the recession-vs-expansion Sharpe-ratio differential; this may require a tighter link between the jump intensity and observable funding-tightness proxies.; Provide robustness checks (out-of-sample forecasts, alternative proxies for S_t) and report all moments (unconditional and conditional) with confidence intervals rather than nan or explosive values.

Notes: The novelty column summarizes the LLM referee's comparative assessment against prior attempted mechanisms in the project. The suggestions column reports the referee's main next-step recommendations to improve theoretical grounding and empirical validation for each model.

Table 15: Retry Configuration

Component	Type	Attempts
LLM API	Exponential backoff	3 (1s, 2s, 4s)
Mathematical Auditor	Audit + fix-and-reaudit	1 + 2 fixes
Pseudocode Agent	Generate + review-and-retry	1 + 2 retries
Coding Agent	Fresh starts $\times$ retries	$2\times3=6$
Code-vs-Pseudocode Review	Review + retry	1 + 1 retry
Structured Output	Simple retry	3

Notes: This table reports retry policies used by the production pipeline. API retries use exponential backoff. The mathematical auditor runs one audit; if it fails, the proposal is fixed and reaudited up to two times. The pseudocode agent generates pseudocode and, if the reviewer rejects it, regenerates with feedback up to two times. The coding agent allows up to two fresh starts, each with up to three repair attempts. Code-vs-pseudocode review triggers one retry if it fails.