Scientific Research Publishing

Journal of Software Engineering and Applications, 2026, 19(8), 301-348
https://www.scirp.org/journal/jsea
ISSN Online: 1945-3124
ISSN Print: 1945-3116

Filter before You Solve: A Deterministic-First/Learned-Second Architecture for AI-Driven Portfolio Management with Real-Money Training-Investment Calibration

George Melville[1], Dena Ghiassi[2], Scott Inthathirath[2], Julian Yeomans[1*]

[1]Schulich School of Business, York University, Toronto, Canada
[2]York University & THETAFinance.AI, Toronto, Canada
Email: georgedm@yorku.ca, dghiassi@thetafinance.ai, sinthathirath@thetafinance.ai, *syeomans@yorku.ca

How to cite this paper: Melville, G., Ghiassi, D., Inthathirath, S. and Yeomans, J. (2026) Filter before You Solve: A Deterministic-First/Learned-Second Architecture for AI-Driven Portfolio Management with Real-Money Training-Investment Calibration. *Journal of Software Engineering and Applications*, **19**, 301-348.
https://doi.org/10.4236/jsea.2026.198013

Received: August 1, 2026
Accepted: August 23, 2026
Published: August 26, 2026

Abstract

This study investigates a deterministic-first/learned-second AI/ML framework that is deployed in regulated retail brokerage accounts. A two-stage calibration system is employed that combines historical back-testing with live recalibration via continuous position snapshots. The framework includes a unique explainability layer that employs the global sensitivity analysis method, SimDec, to identify the most influential components. SimDec renders every admissible decision fully attributable to a joint state of the input space. Because the attribution is an input to the trigger rather than a commentary on it, no generative step exists in the decision path and hallucination is structurally unreachable and, therefore, absent by construction. An intraday options trading financial framework is illustrated through a live training-investment cycle on long-call positions using "real money". The major contribution of this research is the overall architectural and methodological framework. The filter-before-you-solve approach enables contributions to be evaluated independently of specific implementations. Beyond financial applications, it is described how the complete architectural pattern can actually generalize to many AI/ML deployment contexts that require auditable deterministic gating prior to learned inference.

Keywords

Explainable AI in Finance, AI-Driven Strategy, Machine Learning
Insight AI in Business, Options Management, Filter-before-You-Solve,

Deterministic-First/Learned-Second Architecture, Hallucination-Free, Interpretability, SimDec, Real-Money Training

1. Introduction

Deploying artificial intelligence (AI) and machine learning (ML) systems into regulated, capital-at-risk business contexts requires architectural discipline and scrutiny that pure-ML systems cannot provide on their own [1] [2]. The three major shortcomings to AI/ML implementation in financial systems are (i) hallucination, (ii) the absence of reproducibility, and (iii) the absence of an audit trail [1]-[3]. *Hallucination* is the production of factually incorrect outputs by generative or transformer-based ML components whose veracity is stated with high confidence. In capital-allocation contexts, hallucinations can produce positions that violate stated risk policies. The *absence of reproducibility* refers to situations in which successive runs of an ML system can generate entirely different outcomes even though identical inputs have been maintained between each instance. The *absence of an audit trail* occurs when the logical chain flowing from the input data through the model decision to the operational outcome is not traceable to primary sources. These failure modes are widely discussed in the AI-explainability (XAI) literature [2] [4], but have proven difficult to eliminate at the algorithm level [3] [5] [6].

This study introduces a different perspective by taking an alternate standpoint. Rather than attempting to eliminate these failure modes within the internal workings of ML algorithms, our approach removes them entirely at the architectural level [5] [6]. This is accomplished by structuring the designed system in such a fashion that all ML components are restricted to operate inside a feasible region defined by deterministic, mathematical, fully-auditable methods [3] [6]. As a consequence, hallucination becomes bounded as a structural property rather than as an algorithmic property. Reproducibility is preserved at the architectural level because the selection work is done in a prior deterministic stage that produces consistent outputs for identical inputs. The audit trail is preserved because every primary-source data input is recorded and traceable at the position level [3].

The interpretability and explainability is operationalized in this framework via the SimDec procedure [7] [8]. SimDec is a global sensitivity-analysis approach that can be used to identify influential components through its *Simple Binning Algorithm* routine [7]-[9]. The binning indices can be calculated from relatively-small, single datasets and have been shown to possess significantly lower computational overheads with more stable accuracy levels than existing "industry-standard" Sobol-styled procedures [10]-[12]. To satisfy the requirements needed for full auditability, small training-corpus snapshots, and real-money decision-loops in the framework's deployment context, the computational properties render SimDec an appropriately attractive methodology. The binning factors contribute three distinct operational features to the architecture. First, they partition the input space into

joint-state bins that locate each evaluated position to a specific corner of the parameter space. This categorical attribution supplies the required position-level for audit-trail traceability. Second, they identify which input combinations possess the largest degree of influence in a form auditable from the training corpus alone. This eliminates dependence on any external libraries and on any specific design-of-experiments sampling structures. Third, the process makes the explainability commitment concrete by attaching every position-level decision to a quantitative attribution value computed via a transparent, open, and inspectable procedure. These methodological components will be described in detail as the use of SimDec results in all subsequent AI-created solutions being both hallucination-free and fully attributable to a joint state of the input space.

The architectural philosophy will be assessed empirically with an application to portfolio management operating under a *filter-before-you-solve* (FBYS) framework. FBYS is a mathematical formalization that enables contributions to be assessed independently of specific implementations [5] [6]. The framework is evaluated against a "real money" options-trading implementation held in two regulated retail brokerage accounts. Position outcomes (moneyness, time-to-expiry, implied volatility) are recorded based upon the operations within these brokerage accounts [13]-[18]. *Moneyness* describes the relationship between the strike price of an option and the current market price of its underlying asset [14]. This indicates whether the option holds intrinsic value if it were exercised immediately and is categorized as *In-The-Money* (ITM), *At-The-Money* (ATM), or *Out-Of-The-Money* (OTM) [13]-[18]. *Time to expiry*, often referred to as *Days to Expiration* (DTE), is the duration between the current date and the date that the option contract becomes void [14]. *Implied Volatility* (IV) is a market forecast showing how much the underlying asset price will fluctuate over the option's lifespan [14]. IV dictates the extrinsic value of an option contract as it is derived directly from current option premiums using pricing models such as Black-Scholes [14] [18]. In traditional finance, theta represents the rate at which an option's price declines over time, while the cliff corresponds to changes in the rate of decay as the expiration of an option's life approaches [14] [19] [20]. While the asymptotic theta acceleration in the final hours of an option's life has been considered previously [13]-[16], the only study that has parameterized sector-conditional asymmetry at the actually-observed decay rate on a position appears in [3]. The framework's complete inference stack runs entirely on a local computer architecture with no cloud services on any layer [3].

Hence, our study will concurrently and independently evaluate both the (i) methodological and (ii) empirical contributions of the architecture. The empirical phase will be assessed in a complete options-trading deployment cycle involving a real-life investment phase. The contributions from the methodological component will consider six specific evaluation primitives that have not been previously considered in combination by the AI-in-finance literature. (1) A two-stage calibration of an AI/ML finance model will be conducted. Stage 1 will derive gate-

way-rule thresholds for a deterministic filter via historical back-testing under a walk-forward back-test protocol. Stage 2 will recalibrate the inference layer using continuous position snapshots from the real-money deployment. (2) Empirical, intraday primary-source documentation of the theta cliff will be conducted. While academic studies of the theoretical asymptotic theta acceleration over the final hours of an option's life are well-established, empirical intraday primary-source documentation of position-level holding traversing is not. In the proposed framework, documentation will show that the edge-decay enforcement architecture can clearly be considered empirically motivated rather than purely theoretical. (3) A training-investment economic model using real-money will be used to stress-test the actual performance of the architecture. Conventional AI/ML treats training as fixed cost and, more specifically, conventional ML in finance views positions as profit-and-loss (P&L). The proposed framework treats deliberate position retention through stress periods as a training-data acquisition cost and tracks the realized benefit when the trained capability subsequently fires. (4) Hallucinations become bounded as overt, structural properties of the architecture, so ML outputs cannot violate rule sets or circumvent constraints. Consequently, hard-filter exclusion plus constraint-enforced ranking means hallucinated outputs cannot select rejected instruments, cannot violate position-size or correlation-group caps, and cannot override the regime-conditional priority structure. (5) Time-decay-aware exit rules become non-discretionary architectural primitives. This, operationalizes the well-established theoretical property that the convexity edge of long-call positions decays through time independent of the path of the underlying. (6) The structure of the live-money training example from a regulated brokerage account enables full audit-trail traceability. Consequently, this structure removes the look-ahead bias, survivorship curation, and simulation-reality gap that have limited previous conventional ML training corpora.

The six primitives will be encapsulated within a deterministic-first/learned-second architecture. Notably, the mathematical filtering stage culls a substantial portion of the input universe prior to any ML component runs. The surviving candidate set is jointly optimized by a deterministic mixed-integer programme (MIP) that enforces the hard portfolio rules at the solver level. As a result, all of the AI/ML components (*i.e.* deep-learning and transformer-based numerical-feature processors) operate solely within the boundaries of a fixed, constrained feasible region, by default. All of the methodological primitives will be evaluated on their relative contributions to these overall architectural merits.

In summary, this study will focus on the use of AI/ML in business across innovations, applications, and impacts. An applications case study at the architectural-pattern level will be provided that documents the AI-driven decision-making process deployed in a regulated financial-account environment that is recursively trained on live transactional data. The paper presents methodological primitives, a recursive training-and-calibration architecture, and a primary-source-anchored deployment record from an AI/ML-driven portfolio management framework.

The overall contribution is framed at the level of architectural pattern, empirical deployment evidence, and the recursive-calibration methodology. Finally, it will be shown that the training-and-calibration architecture in combination with the deterministic-first/learned-second pattern can be straightforwardly generalized beyond options portfolio management to **any** AI/ML context in which regulatory constraints require auditable, explainable, and interpretable solutions.

2. Materials and Methods

This section explains the AI/ML framework for finance, the deterministic-first/learned-second architectural design, the AI/ML components and their bounded role, and the data-engineering underpinning the empirical record.

2.1. Related Work

2.1.1. AI and ML in Portfolio Management

A substantial literature has applied ML methods to asset-pricing and portfolio-construction problems [21]-[23]. The dominant methodological approach is end-to-end learning in which neural-network and reinforcement-learning agents map input features to allocation decisions through learned weights. While the power of these methods has been well-documented, their operational properties (reproducibility, auditability, hallucination removal) remain relatively less developed.

Our framework departs from end-to-end learning at the architectural level. ML components are not eliminated but are not used to make entire allocation decisions. Instead, ML is confined to layers which do not compromise the operational properties that high-stakes investing environments require. Deterministic mathematical methods are used for admissibility filtering and hard-constraint enforcement, while ML is only employed under circumstances which contribute value. This approach is congruent with the inherently-interpretable requirements necessary in high-stakes investing domains [2] and provides an additional safeguard against model overfitting arising from low-probability, market moving events [24].

2.1.2. Hard-Constraint Optimization in Portfolio Management

Mixed-integer programming has provided a consistent foundation for portfolio-construction problems [22] [25] [26]. What distinguishes our framework architecturally is not the MIP, itself, but how it is positioned within the pipeline. In our pipeline, MIP operates exclusively on the post-filter survivor set produced by the deterministic filtering stage (Section 3.2). Increases in input-universe size affect the linear-time filtering pass, but do not affect the MIP runtime in any direct sense as the data-scale stage and the NP-hard solver stage are decoupled.

2.1.3. The Training-Data Problem in Finance ML

Back-tested performance on data does not reliably predict live performance. A persistent methodological impediment in finance ML is that conventional training datasets (historical price series, vendor-curated factor data, simulated option

chains) are all subject to (i) look-ahead bias, (ii) survivorship bias, and (iii) the simulation-reality gap [23]. The data-engineering methodology (Section 2.4) takes training data that is generated continuously by a real-money operation and captured via continuous CSV-format (comma-separated value files) snapshots from the brokerage interface. Each row records the exact position state at a specific timestamp preserving the exact decision context. Look-ahead bias is not possible as the data is recorded in real time at the moment of decision. Survivorship-biased curation is not possible because every position the framework has held appears in the dataset. Analogously, no simulation-reality gap is possible as the actual data is the live trading record. This is called the real-money training-investment economic model and its economic implications are considered in Section 3.2.

2.1.4. Explainable AI and Bounded Hallucination

Hallucination refers to the production of confident but factually incorrect outputs by AI/ML processes. It is an acknowledged failure mode of generative and transformer-based architectures [1] [27]. In capital-allocation contexts, hallucination can produce positions that violate stated risk policy. Algorithmic mitigation strategies (calibration, retrieval-augmented generation, chain-of-thought verification) can reduce the occurrence of hallucinations, but cannot eliminate them. The strategy that prevails in our structure is completely different. AI/ML components are confined to a layer downstream of the deterministic filter and the hard-constraint solver. The deterministic filter excludes precisely those inputs on which the learned ranker has insufficient training-data support (inputs prone to hallucination) from the inference layer by construction. Hallucination-prone inputs cannot reach the layer where AI/ML components operate. Consequently, no hallucination outputs can be produced.

2.1.5. Time-Decay-Aware Exit Rules in Long-Option Strategies

A property that distinguishes our framework from most published portfolio-management approaches is its explicit treatment of time-decay-aware exits for long-call options as a non-discretionary rule, not as a discretionary risk-management overlay. The academic literature on options pricing has long established that long-option positions carry negative theta in which the premium decays over time even when the underlying asset continues to trend favourably [13] [14]. The variance-risk-premium literature further documents that this time-decay cost is structurally compensating short-volatility positions [15] [16]. This indicates that long-option holders pay a persistent premium for convexity that erodes over time independent of the underlying's path. The implication is direct: a long-call position's edge (the convexity advantage that justified entry) decays through time and may have expired before the position has triggered a profit-taking exit on price-based rules alone.

Despite this well-understood theoretical structure, time-decay-aware exit rules are not a standard feature of published AI/ML portfolio-management frameworks [3]. The dominant academic exit-rule structures are price-based (stop-loss thresh-

olds, profit-target thresholds), variance-based (drawdown caps), or rebalancing-based (periodic re-optimization) [14]. Practitioner literature does discuss time-stops in which the convention is to close short-premium positions at fixed days-to-expiry to avoid gamma acceleration [18]. But these are mechanical date-based rules, not model-derived edge-expiry signals. Our edge-decay enforcement (Section 2.2.5) departs from both. It forces position closure when the modelled advantage of the trade has expired, regardless of mark-to-market state or calendar position. No published AI/ML portfolio-management framework exists that operationalizes this distinction in non-discretionary form. The AI/ML pipeline can recommend high-conviction positions that retain no operational edge. A framework without an edge-decay rule may indiscriminately retain such positions through the full trajectory of theta decay.

2.1.6. Mathematical Formalization of Filter-before-You-Solve

Filter-before-you-solve (FBYS) is a mathematical formalization that allows the contribution to be evaluated independently of the specific implementation. If U denotes the universe of candidate instruments at decision time t and F denotes a deterministic filter that maps U onto an admissible subset A, then:

$$F: U \to A, A \subseteq U \tag{2.1}$$

The filter F is composed of a finite set of rule-based predicates $\{p_1, p_2, \ldots, p_k\}$ that evaluate to True or False on each element of U. A is the set of instruments for which every predicate evaluates to True. The predicates $\{p_i\}$ are deterministic in the strict sense. Given the same input candidate x in U, every predicate returns the same Boolean value on every evaluation. The predicates are policy-rules of the framework (e.g. regulatory-eligibility rules, position-size limits, time-to-expiry windows, and operator-defined risk constraints).

Let L denote the learned-inference component that maps A onto the real-valued ranking space:

$$L: A \to \mathbb{R} \tag{2.2}$$

L is the framework's machine-learning ranker. The framework's instrument-selection decision at time t is the optimization:

$$x_t = \text{argmax}_{\{x \in A\}} L(x) \tag{2.3}$$

FBYS is the strict ordering of the two stages. F is applied before L, and L's domain is restricted to the admissible set A produced by F. Equation (2.3) is named the *hard-constraint architecture* because the rule-based predicates {pi} cannot be traded against learned ranker score. An instrument fails any predicate excluded from A regardless of how high $L(x)$ would have been on it.

The alternative architecture in which L's domain is the full universe U and the rule-based predicates $\{pi\}$ are applied as soft penalties or post-hoc filters on L's output produces a different optimization:

$$x_t = argmax_{\{x \in U\}} \left[L(x) - \lambda \sum_{i=1}^{k} 1\left(pi(x) = False \right) \right] \tag{2.4}$$

where the rule-violation penalty is a hyperparameter weighting and an indicator function summed over rule violations [28]-[30]. Equation (2.4) is the "soft-constraint" or "penalty-method" architecture common in reinforcement-learning frameworks [29] [30].

Equations (2.3) and (2.4) produce different decisions in general. Specifically, when there exists x in U with $L(x)$ high enough to outweigh the penalty term despite one or more of the rule-based predicates being violated. Equation (2.4), a framework could recommend an instrument that violates a hard rule if the learned ranker rates it sufficiently highly. In Equation (2.3), no rule violations can occur. The rule violators, *U-A*, are precisely the inputs on which the learned-ranker's training support is the sparsest (*i.e.* the most hallucination-prone ones). This is precisely why filtering them out prior to inference, Equation (2.3), removes the failure mode that the soft-constraint form, Equation (2.4), cannot.

A major contribution of this paper to the architectural-pattern level is the operational case for choosing Equation (2.3) over Equation (2.4) in regulated AI/ML deployment contexts where the rules $\{pi\}$ encode regulatory-eligibility, fiduciary, or fail-safe constraints that the framework operator does not wish to be tradeable against learned ranker score. The empirical case for choosing Equation (2.3) over Equation (2.4) is the deployment record reported in Sections 2.4 through 4.2. The mechanism by which Equation (2.3) removes hallucination from the framework's decision pipeline (in contrast to Equation (2.4), where hallucination-prone inputs remain in the learned ranker's domain) is treated formally in Section 2.1.7.

2.1.7. Why Filtering before Inference Removes Hallucination

This subsection derives the mechanism by which the deterministic-first/learned-second architecture of Equation (2.3) removes hallucination from the learned ranker's decision domain. The derivation rests on the cardinality of the search space over which L operates being no larger in Equation (2.3) than in Equation (2.4). By construction, the admissible set is a subset of the candidate universe. The cardinality of the admissible set is at most the cardinality of the universe. The empirical cardinality reduction can span multiple orders of magnitude. There is no claim that the specific reduction ratio is optimal or that it generalizes beyond the deployment context. However, the admissible set tends to be smaller than the candidate universe and this reduction is generally large enough to be operationally consequential.

Hallucination is the phenomenon in which a learned-inference component returns a high-confidence ranking score on an input for which it has insufficient training-data support. Namely, an input that lies outside the well-supported region of the training distribution. This definition is consistent with the broader ML usage of the term and is adopted throughout this study. Under this definition, the rule-violators in *U-A* are precisely the inputs most prone to hallucination behaviour by the learned ranker, because they are the inputs on which the learned ranker's training data is sparsest. The deterministic-first architecture of Equation (2.3) eliminates these inputs from the learned ranker's decision domain entirely,

by construction: x_t is selected from A, so the learned ranker's evaluation never queries any x in U-A.

In the soft-constraint architecture of Equation (2.4), the learned ranker is evaluated on every x in U (including rule-violators) and the optimization can return a rule-violator if the learned rank is sufficiently high enough to outweigh the rule-violation penalty. The rule-violators are the most hallucination-prone inputs as established above. The optimization of Equation (2.4) can therefore return a hallucinated recommendation if the rule-violation penalty is set too low relative to the magnitude of the learned ranker's hallucinated output on rule-violators. No value of the rule-violation penalty uniformly prevents hallucinated recommendations in Equation (2.4) without introducing other failure modes. For example, prohibitively high penalty values could make Equation (2.4) functionally equivalent to Equation (2.3). The deterministic-first architecture of Equation (2.3) eliminates this failure mode by construction.

The implication for ML-accuracy under the framework's two-stage calibration cycle (Section 3.2) is that the recursive training-and-calibration data generated is concentrated entirely within the admissible region. A learned ranker, therefore, receives training updates only on inputs that have already passed the deterministic predicates. These are inputs for which the framework has both regulatory-fit and operator-defined risk-fit. Consequently, the training signal is both higher-quality (no rule-violator noise contaminates the supervised signal) and higher-density (the same number of training observations is concentrated on a strictly smaller set). Both effects compound across recursive calibration cycles. As the framework deploys for longer, the learned ranker's estimation error within the admissible region converges faster than it would in an architecture that trains the learned ranker on all of U.

2.2. Architecture: Deterministic-First/Learned-Second

2.2.1. The Layered Architectural Commitment

The framework's design rests on a layered separation of decision-making responsibilities.

The Deterministic Filter and Hard-Constraint Solver. The first substantive layer of the framework is fully deterministic and free of any AI/ML component. It applies a sequence of admissibility filters to the candidate universe, then submits the post-filter survivors to a hard-constraint MIP that produces a candidate allocation. Both stages are mathematically deterministic: identical inputs produce identical outputs, the operational complexity is well-defined, and no element of the decision relies on a learned component. Inputs are: the price/volume snapshot at the close of the most recent trading session, the operator-specified risk policy, and any active operator-judgement-layer overrides. Outputs are: a candidate allocation under the operator-specified risk policy.

The Learned-Inference Component. Subsequent layers incorporate AI/ML methods including deep-learning and transformer-based numerical-feature pro-

cessors. These components produce ranking signals that influence the joint optimization objective and inform the operator-judgement-layer decisions. Critically, the learned-inference stage operates within the constrained feasible region the deterministic-filter stage has already established. The learned-inference stage (i) cannot promote rejected candidates back into the feasible region, (ii) cannot exceed any position-size or correlation-group cap, and (iii) cannot disable any hard constraint. ML components affect the ranking and weighting of admissible positions, but they do not affect what is admissible.

The architectural commitment is therefore not "no AI/ML anywhere", which would be unproductive in a contemporary AI-in-business context, but rather "no AI/ML in the layer where regulatory eligibility, fiduciary obligations, or capital-at-risk failures must be precluded by construction." The deterministic-filter stage disposes of the substantial majority of the universe through purely mathematical operations. AI/ML is permitted in the layers where its adaptive modeling capability adds value (pattern recognition over conditional distributions, ranking, attention-based feature aggregation) but is structurally prevented from contradicting the rule set encoded in the deterministic-filter stage. This protocol is referred throughout as the *deterministic-first/learned-second* commitment.

2.2.2. The Filter Stage

The first operation on a candidate universe is a sequence of admissibility conditions applied as vectorized mathematical operations. Conceptually, the filtering stage is a series of Boolean predicates evaluated in parallel across the input universe where each predicate excludes non-compliant candidates. The output is the set of admissible candidates on which the downstream MIP operates.

The framework employs six filtering stages for options-trading, organized by the type of admissibility they enforce:

Filter 1: Long calls only; puts excluded. The framework operates a long-call-only options strategy. Put options are excluded by hard architectural commitment.

Filter 2: Inflowing-sector restriction. Only sectors classified as flow-positive at the evaluation time are admissible. Sectors classified as flow-negative or flow-neutral are excluded.

Filter 3: In-the-money compliance with cost-efficient days-to-expiry selection. A call option is admissible only if it satisfies three concurrent admissibility conditions on (i) in-the-money depth, (ii) delta, and (iii) days-to-expiry. The days-to-expiry condition admits either a primary band or a stricter cost-efficient sub-band defined by additional tighter floors on in-the-money depth and delta.

Filter 4: Single-strike-per-name. No more than one strike per underlying may be open simultaneously.

Filter 5: Position-size cap. No position may exceed a hard per-position percentage of total portfolio value at entry. The architectural commitment is to a hard cap enforced at the solver level.

Filter 6: Correlation-group cap. Aggregate exposure to any one sector or cor-

relation peer group may not exceed a hard per-group percentage of total portfolio value.

Empirically, the six filters admit a small fraction of the input universe to the candidate set on a typical decision day. This depends on which sectors are flow-positive and which underlyings have ITM call options that satisfy the three concurrent compliance criteria. The compression ratio is operationally consequential. It focuses the joint MIP optimization on a tractable working set and concentrates the learned-inference stage's training updates on inputs that have already passed the deterministic predicates. The deterministic stage is therefore both a hallucination-prevention mechanism (Section 2.1.7) and a computational-tractability mechanism.

2.2.3. The Joint Mixed-Integer Optimization

The post-filter candidate set is presented to a deterministic MIP that produces an allocation as a single joint optimization rather than as a sequence of independent position-by-position decisions. The MIP formulation respects the architectural commitments of Section 2.2.4 (hard constraints at the solver level) and produces a candidate allocation in which every position-level decision is consistent with portfolio-level policy. Joint optimization across the candidate set is computationally feasible because the deterministic-filter stage has already reduced the search space to a tractable working set. The joint optimization across the candidate set, hard constraints at the solver level, multi-objective goal-programming structure are independent of the specific MIP formulation.

The computational implication of joint optimization requires care. Joint optimization with mixed binary admissibility variables and continuous weighting variables is an NP-hard problem. Tractability rests on two architectural commitments. First, the deterministic-filter stage compresses the input universe substantially before the MIP runs, holding the working-set size at a level where commercial MIP solvers return optimal solutions in a couple of seconds on commodity hardware. Second, the MIP formulation uses standard linear-and-quadratic constraint forms. The framework does not depend on specialized solver capabilities or research-grade decomposition techniques. The architectural pattern is therefore reproducible by any researcher with access to a commercial MIP solver (Gurobi, CPLEX, or equivalent) and the deterministic-filter stage's compression performance.

2.2.4. Hard Constraints at the Solver Level

A central architectural commitment is that portfolio-level rules are enforced as *constraints* in the mathematical-programming sense, not via *penalties* in an objective function. Constraints define the feasible region of the optimization such that the solver is mathematically incapable of returning a solution outside the feasible region. Penalties, by contrast, allow the optimizer to violate the rule if the resulting objective improvement exceeds the penalty cost.

The distinction is consequential. Penalty-based optimizers, including most mod-

ern reinforcement-learning portfolio agents, typically resolve constraints as soft penalties on the learned objective. The optimizer can violate a stated constraint if the violation produces an objective improvement larger than the penalty. In high-stakes regulated contexts, this is exactly the wrong direction for failure since the rules being violated typically exist precisely because their violation must be precluded by construction. The hard-constraint formulation eliminates this failure mode by mathematical construction. The MIP solver cannot return a constraint-violating allocation. There is no operator setting that allows the violation to be tolerated for sufficient objective improvement, because the framework does not expose constraint enforcement as a hyperparameter.

Hard constraint enforcement at the solver level is one of the structural mechanisms that produces the hallucination-removal property. The combined property is consolidated in Section 2.3.4. The hard-constraint enforcement described here works in conjunction with the hard-filter exclusion described in Section 3.2.

2.2.5. Profit-Confirmation and Edge-Decay Exit Rules

Two model-derived, non-discretionary exit rules that execute without operator intervention complete the architectural components. Both rules are applied at the close of the trading session and are encoded in the framework's policy layer.

The *multi-axis profit-confirmation exit rule* applies to profitable positions and requires convergent positive-return signals at the close before triggering an exit. The convergence requirement prevents premature exit on intraday spikes and prevents the framework from exiting profitable positions on noise. It is a non-discretionary commitment. The operator cannot override the multi-axis convergence requirement at run time as it is built into the policy layer.

The *edge-decay enforcement rule* imposes position closure when the modelled advantage of the trade has expired, regardless of the mark-to-market state. It operationalizes a structural property of long-call positions that is theoretically well-established. The convexity edge of an in-the-money call decays with time even when the underlying continues to trend favourably. Long-option positions carry negative theta [13] [14] and variance-risk-premium research has indicated that this time-decay cost compensates short-volatility positions [15] [16]. Long-option holders pay a persistent premium for convexity that erodes through time independent of the underlying's path. The framework's signal layer evaluates each open position against an edge-expiry condition. When it fires, the position is closed at the next session's open. The rule is unconditional on intermediate mark-to-market state and unconditional on calendar position. The operator does not have the option to retain the position past the edge-expiry signal.

The edge-decay enforcement rule is completely *model-derived* as opposed to calendar-derived. The rule fires when the edge-attribution model determines that the convexity edge has decayed below a closure threshold. This may occur earlier or later than when a fixed-date time-stop would fire on the same position. Furthermore, the rule is *non-discretionary*. When the edge-attribution model fires the closure signal, the position is closed irrespective of any intermediate mark-to-

market trajectory or operator preference. There is no published AI/ML portfolio-management framework that operationalizes this distinction in non-discretionary form for long-call positions. This distinguishing aspect merits emphasis because, without this rule, an AI/ML pipeline would produce high-conviction positions that continue to be held simply because price-based exit thresholds have not fired, yet retain no operational edge. The edge-decay rule prevents this failure mode by closing positions at the model-derived edge-expiry signal rather than waiting for price-based confirmation that the position has already gone wrong. Consequently, exit decisions are driven by changes in the model's estimated edge function rather than solely by realized price movements.

2.3. AI/ML Components

2.3.1. What ML Does in This Framework

AI/ML components contribute to three distinct task categories within the constrained feasible region established by the deterministic-filter stage.

The drawdown and time-decay components of the MIP's optimization objective rely on *conditional-distribution estimation*. For example, the distribution of expected retracement is conditional on prior unrealized gain, sector regime, days-to-expiry remaining, and other state variables. These conditional distributions are not reducible to closed-form rules, since they reflect non-linear, context-dependent patterns in the data. AI/ML components implemented as deep-learning numerical-feature processors produce these estimates from the CSV-snapshot training corpus. The architectural value is that the ML estimator can capture conditional-distribution shape (skewness, kurtosis, regime-conditional variance) without imposing parametric assumptions that the data does not support.

Pattern-recognition tasks over multivariate market state, sector-flow classification, and regime classification are handled by transformer-based numerical-feature processors. These components operate over the feature corpus (CSV-snapshot datasets combined with public market data) and produce categorical outputs (sector classification labels, regime labels) that flow into the MIP. The architectural value of ML is that the classification tasks involve interactions between multiple market-state variables that are well-suited to transformer-attention layers and poorly-suited to hand-engineered classification rules.

Once the deterministic-filter stage has admitted a candidate set and the joint MIP has produced a feasible allocation, the framework's learned-inference layer produces *rankings within the feasible region*. This ranking informs operator-judgement-layer decisions and recursive-calibration updates, but does not override, constraint enforcement.

All AI/ML components are public-domain deep-learning architectures from the time-series and sequence-modelling literature. Six categories of component are integrated through an ensemble layer. (1) *Sequence models* are Bidirectional Long Short-Term Memory (BiLSTM) networks produce sequence-aware embeddings of intra-session and across-session position trajectories that capture mo-

mentum, mean-reversion, and persistence patterns that cross-sectional models cannot resolve. (2) The *transformer-based numerical-feature processors* are a family of attention-based architectures spanning the standard Transformer, the Temporal Fusion Transformer (TFT), Informer, PatchTST, iTransformer, and TimesNet operates over multivariate market-state inputs. The architectures contribute complementary inductive biases including: TFT for mixed static + time-varying inputs with interpretable variable-selection; Informer for long sequences via sparse attention; PatchTST for local-temporal-pattern extraction via patching; iTransformer for cross-variable rather than cross-time attention; TimesNet for multi-periodicity modelling. (3) A categorical *multi-state regime classifier* produces BULL/BEAR /SIDEWAYS labels per daily snapshot. The discrete-label output feeds the regime-conditional decision logic (*i.e.* the $\alpha(s, r)$ parameterization [3]). (4) A categorical *composite quality scorer* that produces HIGH/MEDIUM/LOW tier labels that are constructed from a composite of profitability, growth, safety, and payout characteristics according to the Quality-minus-Junk lineage of [31]. (5) A distribution-detecting *data drift monitor* and its associated *regime-shift balancing* algorithms rebalance the inference layer's outputs during regime transitions. The architectural role is to detect input-feature distributional shifts sufficient to invalidate the current inference-layer calibration and to trigger Phase 2 inference-layer recalibration cycles (Sections 2.4.1, 4.2.1). (6) An *ensemble layer* is used to aggregate outputs from the sequence and transformer architectures into the downstream inference inputs to mitigate single-model bias.

All components are publicly-documented architectures, locally executed on the L1 public-data cache (Sections 2.3.3 and 2.4.5), and elaborated at an architectural-pattern level in [3].

2.3.2. What AI/ML Does Not Do

The AI/ML components do *not*: (i) Select positions directly. Position selection is the joint product of the deterministic-filter stage's filter pass and the deterministic MIP's output; (ii) Override hard constraints. Constraint enforcement occurs at the deterministic-filter stage and the ML components operate downstream within the constrained feasible region; (iii) Perform large-language-model deliberation as part of position selection. There is no LLM in the decision path. ML components in the subsequent layers are deep-learning and transformer-based numerical-feature processors that produce categorical or continuous-score outputs. They do not introduce natural-language deliberation as part of position selection; and, (iv) Make external API calls or transmit data beyond the local machine. All ML inference runs locally on commodity hardware. There is no cloud inference dependency, no third-party API dependency, and no internet dependency in the decision path beyond the data-ingestion stage that pulls market quotes from public sources.

2.3.3. Locally-Executed Inference and Operational Properties

The complete inference stack, which runs locally on a standard laptop, consists of

(i) the L1 public-data cache, (ii) SimDec binning, (iii) the multi-state regime classifier, (iv) the composite quality scorer, (v) the BiLSTM and Transformer-family deep-learning components integrated through an ensemble layer, (vi) the Data Drift monitor, and (viii) the joint MIP solver. The L1 cache refreshes continuously throughout each trading day via free public APIs with each refresh cycle completing within minutes. The auditability, data-sovereignty, and reproducibility properties all depend on the complete stack remaining under operator control. This is a deliberate architectural commitment and not undertaken to optimize costs or simplification for practical terms.

The commitment has consequences for operational properties that are particularly relevant in business contexts. Because the position state, allocation decisions, and performance attribution remain on within the local execution environment, there is no data egress because no third party receives the operator's portfolio data as part of normal operations. A network outage, API rate limit, or third-party service degradation does not affect the framework's ability to run a complete evaluation cycle. Market-data ingestion is the single network-dependent step and, if market data is available, the rest of the pipeline runs locally. Consequently, no external dependency exists in the decision path. End-to-end cycle latency on commodity hardware occurs in low single-digit seconds. The computational footprint is dominated by data ingestion rather than the filter pass or the MIP solve. ML inference in the subsequent layers contributes only a small fraction to the total cycle latency.

2.3.4. Hallucination Removal as a Structural Property

The central architectural property that most directly motivates the framework's applicability to AI-in-business deployments is the removal of hallucinations. ML Hallucination is the production of high confidence, but factually incorrect, outputs which occurs, particularly, in deep-learning and transformer-based architectures. In capital-allocation contexts, hallucination can produce positions that violate stated risk policy. Algorithmic mitigation strategies (calibration, retrieval-augmented generation, chain-of-thought verification) reduce hallucination probability, but cannot eliminate it. The framework's architectural strategy is different because the ML components are confined to a layer downstream from the deterministic filter and the hard-constraint solver. The deterministic filter excludes precisely those inputs upon which the learned ranker has insufficient training-data support (see Section 2.1.7) from the learned ranker's decision domain by construction. Therefore, hallucination-prone inputs cannot reach the inference layer. The learned ranker operates exclusively on inputs that have already passed the deterministic predicates and lie within the well-supported region of the training distribution.

The hallucination-removal property arises from three structural mechanisms operating in series. (i) The *Hard-filter exclusion at the deterministic-filter stage* (Section 2.2.2) removes the substantial majority of the input universe before any ML component runs. The candidate set on which downstream layers operate is

strictly smaller than the input universe (Section 2.1.7). ML components cannot promote rejected candidates back into the feasible region because they never reach them. (ii) The *post-filter candidate ranking* is what the learned-inference layers see as input. ML can affect fine-grained ordering within the admissible region but cannot affect membership in the admissible region. (iii) *Hard constraint enforcement* at the MIP solver level (Section 2.2.4) imposes position-size caps, correlation-group caps, and policy compliance as mathematical constraints. Because the solver returns infeasibility rather than a constraint-violating allocation, the downstream ML components operate solely on solver-feasible allocations.

The combined property mechanisms operate in series. Under the operational definition (Section 2.1.7), hallucination is the phenomenon in which a learned-inference component returns a high-confidence ranking score on an input for which it has insufficient training-data support. The rule-violators are precisely those inputs most prone to hallucination behaviour because they are the inputs on which the learned ranker's training data is the sparsest. The deterministic-filter stage excludes these inputs entirely from the decision domain. By construction, any input evaluated has already passed the deterministic predicates and lies within the well-supported, feasible region of the training distribution. Hallucination-prone inputs cannot reach the inference layer and the solver-level hard-constraint enforcement (Section 2.2.4) provides a second structural guarantee that no constraint-violating allocation can be returned. Consequently, the framework cannot produce hallucinated outputs. By design, hallucination cannot occur inside the decision pipeline and not merely bounded above through post-hoc enforcement. This process is referred to as the *hallucination removal* property.

Relevance. In contexts where regulatory compliance, audit-trail requirements, or capital-at-risk considerations make pure end-to-end ML deployment problematic, the hallucination-removal property is the architectural feature that allows ML adoption at all. The framework demonstrates a deployment pattern in which ML's expressive power is harnessed for those tasks in which ML adds value (conditional-distribution estimation, pattern recognition, ranking). The hard rules of the deployment context are enforced at a layer ML cannot violate. The pattern can be generalized to many other AI-in-business contexts such as AI-driven credit decisions, AI-driven medical decision support, AI-driven supply-chain reordering, and other regulated AI/ML deployment contexts.

2.3.5. SimDec as the Framework's Explainability and Sensitivity-Analysis Layer

The explainability commitment is operationalized through the Simple Binning [7] [9] of SimDec [8]. The architectural argument for SimDec over the established Saltelli-style Sobol estimators [10] [11] is not just relative efficiency. It is that SimDec's properties match the binding operational constraints. As noted above, the complete inference stack can be performed entirely on a laptop hardware.

Eight architectural properties validate SimDec as the *de facto* choice. (i) The *sampling-design flexibility* for binning requires only a single-dataset of observa-

tional settings [7] [9]. (ii) The regime × quality × sector conditional slices that the edge-decay rule depends on possess *smaller-sample stability* than Saltelli-Sobol estimators [7] [12] [32]. (iii) There is *zero infrastructure cost for auditability* as Yahoo Finance, SimDec's open-source code, and the laptop deployment combine to produce a reproducibility surface without licensing, cloud, or vendor dependencies. (iv) For *joint-state attribution as an operational unit*, Sobol scalar indices answer *how much*, but not *where*. Conversely, the edge-decay rule fires on joint-state conditions which correspond to SimDec's native output. (v) SimDec enables *continuous recalibration* within sub-ten-minute cycles on commodity hardware [7] [12]. (vi) The SimDec bins directly create categorical the classifier, while the continuous deep-learning outputs feed standard SimDec binning without intermediate transformation. Hence, there is *direct compatibility with the ML stack*. (vii) The ML stack performs prediction, SimDec performs attribution, and the Data Drift monitor triggers joint recalibration when input distributions shift. Consequently, there is *architectural complementarity rather than competition*. (viii) Sobol analysis would force either lower-cadence sensitivity analysis, cloud-scale compute, or library and sampling-design dependencies that would each break a foundational architectural constraint elsewhere within the framework. By employing SimDec, there is *coherence throughout all of the framework's broader commitments*.

Therefore, the selection is not that "we computed sensitivity indices that happen to be SimDec". The choice is based on the fact that the framework's deployment context, compute envelope, and reproducibility commitment mean that SimDec can serve as the sole method that is completely compatible with the architecture as documented.

2.4. Data Engineering, Two-Phase Calibration, and Reproducibility

This section documents the data engineering, training-investment calibration cycle, and reproducibility design. **Figure 1** shows the upstream market-state characterization layer that feeds the architectural composition described in Section 2.2 and Section 2.3, captured on the 24 April 2026 trading session on which the AMD trade closed.

The figure shows the same trading session on which the AMD trade documented in Section 3.1.2 closed at +120% on the closed lot and on which the AAPL operator-override episode of Section 3.1.7 occurred. The dashboard visualizes the upstream input layer of the FBYS architectural composition documented in Section 2.2 and Section 2.3. The *Sector Dynamics by Return* panel shows the post-rotation regime that produced the AMD outcome: Technology ranked first at +2.30% with INFLOW status (73% breadth, 36/49 green), while Energy shows −0.86% with OUTFLOW status (17% breadth). The autonomous energy-to-semiconductors rotation described in Section 3.1.1 is directly visible in the live sector-status flags, with the semiconductor-bearing Technology sector now leading the

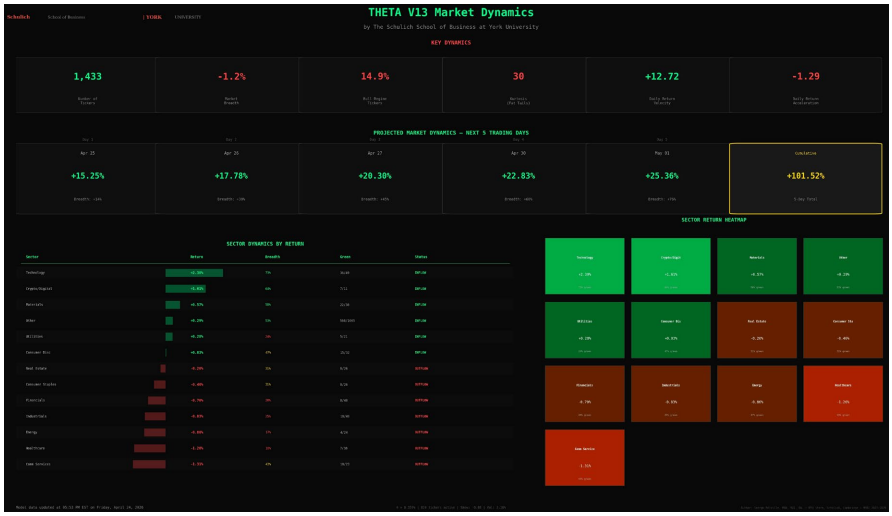

Figure 1. THETA V13 market dynamics dashboard, 24 April 2026 (05:53 PM EST), generated from 824 active tickers on operator-owned laptop hardware.

inflow ranking and the previously-favoured Energy sector in outflow. This sector-conditional state is the empirical realization of the $\alpha(s,r)$ scaling factor of Section 2.2.5. The *Key Dynamics* row shows the regime-classifier and higher-moment market-state inputs at the moment of the AMD close: 1,433 tickers in universe, Bull Regime Tickers 14.9%, Kurtosis 30 (fat tails), Daily Return Velocity +12.72, Daily Return Acceleration −1.29. The *Projected Market Dynamics* row shows the deep-learning ensemble's median five-day forward projection (BiLSTM and Transformer-family architectures of Section 2.3.1), with a 5-day compounded cumulative of +101.52%. The dashboard is generated using Yahoo Finance market data and Bloomberg sector classifications via free public APIs (Section 2.3.3) entirely on localized hardware with no cloud services. The C4 deployment-constraints satisfaction of the architecture is directly visible in the dashboard's provenance footer (model updated 05:53 PM EST Friday April 24 2026; 824 tickers active; market skew −0.88; ensemble θ 0.359%; mean daily volatility 3.16%).

2.4.1. Real-Money Operating Context

The framework's training dataset is generated by its own real-money operation, not by simulated back-testing or vendor-curated historical data. The framework operates against two regulated retail brokerage accounts. Position state is captured at intra-session resolution from broker CSV exports. Transaction history is recorded at execution-tick resolution from the broker's settlement records. The combination of (i) live-money training during a high-stakes deployment window, (ii) deliberate-stress-position-retention as a training-data acquisition methodology, (iii) continuous-snapshot intra-session CSV provenance, and (iv) full audit-trail traceability to a regulated Schedule I Canadian chartered bank brokerage is the methodologically distinctive feature that distinguishes this dataset from both academic back-testing corpora and undocumented hedge-fund tick-archives.

This operating context is methodologically consequential. The training dataset

is generated entirely from real-money positions whose existence is verifiable from the broker's settlement records. There is no possibility of look-ahead bias because the data was recorded in real time at the moment of decision. There is no survivorship-biased curation because every position the framework has held (winners, losers, expired-worthless) appears in the dataset. There is no simulation-reality gap because the data is the live trading record.

2.4.2. Continuous-Snapshot CSV Methodology

Position-level data is captured by continuous downloads of CSV-format position snapshots from the broker's web reporting interface throughout each trading session. Snapshots are downloaded at multiple intervals during each session that capture both peak-of-session market values and end-of-session close. Each snapshot file records the following information, per position:

- Security name (long-form description, e.g., "EXXON MOBIL CORP NEW MAY 1 26 CALL 160")
- Broker symbol
- Asset class (equity, ETF, long call, etc.)
- Denomination currency (USD or CAD)
- Quantity held
- Average cost per share
- Current market price
- Book value
- All-time value change (percentage and dollar)
- Current market value

The cadence of snapshot capture is sufficient to record each position's state through the full trajectory from peak unrealized gain to terminal decay (or alternatively from entry to peak gain to profitable close). The dataset is therefore a *time-resolved, position-state-resolved* record of every position the framework has held.

Temporal ordering of the training corpus. Each snapshot is written once at capture and is never revised. A snapshot bearing timestamp t enters the training corpus only after the decision taken at t has been executed and its outcome recorded. Labels are constructed from snapshots bearing timestamps later than t and are never joined back onto a decision taken at t. It follows that no score used in a decision at time t was produced by a model whose fitting corpus contained any snapshot bearing a timestamp at or after t. The ordering is a consequence of the capture mechanism rather than a convention applied to a historical dataset. The snapshots did not exist before the decisions they document, so they could not have informed them.

2.4.3. Transaction-History Attribution

Complementary to the position-state snapshots, the framework's training dataset includes transaction-history records downloaded as separate CSV exports from the broker's transaction-history report. For each transaction, these records con-

tain:

Description of the security

- Broker symbol
- Transaction date
- Settlement date
- Account currency
- Activity type (BUY, SELL, dividend, expiry)
- Quantity
- Currency of price
- Execution price
- Settlement amount

Transaction records establish the entry and exit prices of every position with primary-source verifiability. Combined with the position-state snapshots, they constitute a complete audit trail from entry decision through holding period through exit outcome.

2.4.4. Why This Dataset Is Unique

The continuous-snapshot dataset has properties that no other public retrospective dataset can match. *No look-ahead bias*: Each snapshot was recorded at the exact moment of the position's state at that timestamp, under the ordering invariant stated in Section 2.4.2. There is no possibility that the framework's training has been informed by data the framework would not have had at the corresponding decision point. *No survivorship-biased curation*: The dataset includes every position the framework has held irrespective of outcome. Expired-worthless options, retraced winners, and partially-closed positions all appear in the dataset alongside profitable closes. *No simulation-reality gap*: The data *is* the live record. There is no model of execution slippage, no model of liquidity, no model of fill probability. The fills the broker recorded are the fills the framework received. *Extreme-state capture*: Because snapshots are downloaded continuously, the dataset captures state during deepest-drawdown moments and peak-pressure decision points. These are exactly the states that historical retrospective datasets compress, smooth, or omit. These states are the highest-information-content observations for ML training. They are where learned-inference models most need supervised signal and where retrospective datasets least supply it. The framework's training corpus contains these states at the operational resolution of the original decision context. *Real decision-context preservation*: Each snapshot row records the operator's exact decision context (entry price, current state, accumulated P&L) at the moment the snapshot was taken. The operator's decisions (enter, hold, partial close, full close) are reflected in the subsequent snapshots. The framework's training dataset therefore records not only the position trajectory, but also the decision context in which each phase of the trajectory occurred.

1) The Theta Cliff Dataset: Intraday Primary-Source Empirical Documentation from a Real-Money Options Account

A specific feature of the live continuous-snapshot dataset is its capture of the

asymptotic theta-cliff trajectory at intraday resolution. The dataset captures eleven distinct intraday snapshots of a three-instrument long-call cohort (XOM, CVX, LNG) traversing terminal-decay on 1 May 2026 in which each snapshot timestamp, market price, and broker-recorded mark-to-market valuation is preserved. The asymptotic acceleration of theta in the final hours of an option's life is theoretically well-established but its empirical documentation in the academic literature relies on simulated or modelled trajectories rather than on primary-source records of position-level holdings. The dataset documented here is the first published academic dataset capturing the cliff at this resolution from a regulated retail brokerage account possessing both continuous-snapshot intra-session provenance and full audit-trail traceability. Subsequent academic work on options time-decay can use the dataset as primary-source empirical evidence independently of the framework architecture.

On the trading session of 1 May 2026, the framework held three concurrent long-call positions with that day's expiry that traversed the terminal-decay region across the session: (i) XOM 160-strike (8 contracts), (ii) CVX 200-strike (1 contract), and (iii) LNG 285-strike (1 contract). The cohort had been entered between 20 March and 21 April 2026 and held through the war/ceasefire/re-escalation pattern documented in Section 1. Eleven intraday snapshots were captured between approximately 09:35 and 16:00 EDT on 1 May 2026. Each snapshot records the broker's mark-to-market valuation per contract, position quantity, and total mark across the cohort. The cohort decayed from CAD 619.26 in aggregate mark at the first snapshot to CAD 13.58 at session close, a decay of 97.8% of the T1 mark within the session. The trajectory is the empirical asymptotic theta-cliff in primary-source form. Table 1 summarizes the cohort's observed decay trajectory across the eleven-snapshot session.

Table 1. Theta-cliff decay trajectory of the XOM/CVX/LNG cohort across eleven intraday snapshots, 1 May 2026. Prices in USD per share; market values in CAD as reported by the broker.

Position	Strike	Qty	Cost basis (CAD)	T1 px	T1 mark (CAD)	Terminal px	Terminal mark (CAD)	Realized (CAD)	T1→T2
XOM	160	8	9,372.54	0.45	489.96	0.01	10.86	−9,361.68	−84.4%
CVX	200	1	1,650.06	0.70	95.27	0.01	1.36	−1,648.70	−81.4%
LNG	285	1	3,048.01	0.25	34.03	0.01	1.36	−3,046.65	−80.0%
Cohort			14,070.61		619.26		13.58	−14,057.03	

For comparison, a typical 60-day-to-expiry option exhibits a theta of approximately 0.1% - 0.3% of premium per day. The three cohort positions exhibited cumulative session decay of between 96% and 99% of their T1 mark across only 6.5 trading hours (XOM 97.8%, CVX 98.6%, LNG 96.0%; cohort aggregate 97.8%). The instantaneous decay rate in the most aggressive interval (T1 to T2, approximately 75 minutes) corresponded to a per-minute decay rate of approximately 1.1% of remaining premium which is three orders of magnitude faster than the

calendar-time theta exposure of a far-from-expiry option. The cliff is not a smooth decay. It demonstrates a phase transition from finite to negligible value over a short interval at its asymptote.

The methodological contribution of this sub-section is the dataset itself, not the framework's use of it. The cohort traversed the cliff because the framework's edge-decay enforcement rule had been overridden by operator judgement during the 24 April 2026 decision window. An operator-override episode that constitutes part of the deployment record (Section 3.1.6). The override is documented at primary-source resolution and is independently informative about the operator-judgement layer regardless of whether the framework's architectural commitments are evaluated favourably. Subsequent academic work on options time-decay can use the eleven-snapshot trajectory as primary-source empirical evidence, while AI/ML deployment governance studies can use the override episode as primary-source empirical evidence on operator-discretion failure modes in regulated AI/ML systems.

The XOM cohort was a partial retention, not a full retention. A precision is owed on the XOM 1 May 2026 160-strike call cohort that contributed to the cliff dataset. The position was a 10-contract original cohort split across two entry tranches in March 2026. Two contracts were closed early on 30 March 2026 at USD 14.35 which generated a broker-recorded realized profit CAD 781.12 (the +66.67% return-on-capital episode of Section 3.1.2). The remaining eight contracts were retained against the framework's edge-decay enforcement signal during the operator-override decision window of 24 April 2026 and decayed to terminal expiry on 1 May 2026 with broker-recorded cost basis CAD 9,372.54 against T11 mark CAD 10.86. The cohort therefore split into an early-harvest sub-cohort (closed at substantial profit) and a retained sub-cohort (decayed to expiry under operator-override). Both outcomes are documented at primary-source resolution and both are part of the framework's deployment record. The early-harvest sub-cohort (2 contracts, CAD 781.12 realized profit) and the retained sub-cohort both contribute to the framework's deployment record. The retained sub-cohort contributes to the theta-cliff dataset (Section 2.4.4.1) and to the operator-override episode (Section 3.1.7).

A timestamped pre-cliff framework recommendation and the documented operator override. A primary-source framework dashboard generated on 24 April 2026 classified the eight retained XOM contracts as edge-expired under the framework enforcement rule (Section 2.2.5). The dashboard recommended position closure at the next session's open. The operator-judgement-layer decision was to retain the position against the framework's signal on the basis of an operator-judgement evaluation that the war/ceasefire/re-escalation pattern made the option's strike still reachable before expiry. The retention decision was implemented, the cohort decayed across the subsequent five trading days, and on 1 May 2026 the cohort traversed the asymptotic theta-cliff. The episode is the strongest evidence that the edge-decay enforcement architecture (Section 2.2.5) is doing

necessary work, because the architecture's signal was correct and the operator-override against it was loss-amplifying. The episode contradicts a counter-hypothesis that the architecture's edge-decay rule is a low-information layer that operator judgement could safely override. The opposite is the case as the override against the rule produced a loss-amplification of approximately CAD 9,361 that the rule, executed without override, would have averted.

2.4.5. Data Hierarchy and Lineage

The data architecture contains two distinct layers serving different functions that run entirely on a laptop (Section 2.3.3).

The *L1 public-data layer* draws from three sources accessed via free public APIs. The sources are: (i) the Yahoo Finance public market data which refreshes continuously throughout each trading day in which each refresh completes within minutes; (ii) the Bloomberg sector/subsector classifications which supply the industry-taxonomy substrate, the sector-conditional regime classifier, and the composite quality scorer consume; and (iii) the EIA petroleum-production statistics (Section 2.4.8) used to anchor the fundamentals of the regime classifier in energy-sector inferences. The L1 layer feeds the SimDec binning analysis (Section 2.3.5), the regime classifier, the composite quality scorer, and the downstream ML inference layers. The L1 cache schema was established on 1 October 2025, coincident with the live-deployment start (Section 3.2), and has been continuously refreshed via the public APIs since that date. The L2 layer is a broker-CSV audit record drawn from regulated retail brokerage accounts that supply the position-level provenance, the recursive-recalibration signal that closes the real-money training-investment loop (Section 3.2), and the audit-trail substrate for the operator-judgement-layer interactions (Sections 2.4.4.1 and 3.1.7). The two layers are independent for the reproducibility surface. The SimDec analysis on the L1 layer is reproducible from free public APIs and no access to the broker-CSV L2 record is required (Section 2.3.5).

The *L2 broker-CSV dataset* is organized in a hierarchy with three tiers. (i) *Position-state snapshots* (per-position, per-timestamp) are generated by the broker's reporting system and downloaded at multiple intervals through each trading session. Each snapshot file is named with a date stamp and account identifier (e.g., account_details_8947_24042026_5.csv). (ii) *Transaction-history records* (per-transaction) are generated by the broker's settlement system which are downloaded as separate CSV exports. Each transaction record is verifiable to the cent and includes the broker's settlement reference. (iii) *Broker-summary attribution* (aggregate, per-session) are generated by the broker's portfolio-summary widget and captured as primary-source screenshots when the operator wishes to verify session-level aggregate P&L attribution against the position-state-derived calculations. The 24 April 2026 same-session attribution documented in Section 3.1 is verified by such a screenshot.

The hierarchy enables cross-validation. Aggregate session P&L computed from the position-state snapshots can be compared to the broker's official session at-

tribution. Per-position trajectories from the snapshots can be compared to the per-transaction price record. Discrepancies that occur can be investigated and resolved before the data enters the training procedure.

2.4.6. The Historical Back-Testing Dataset and Walk-Validation Protocol

The *live-deployment* component of the two-phase calibration methodology is documented in Sections 2.4.2 through 2.4.5. The historical back-testing component is documented in this subsection.

Universe and timespan. The historical back-testing programme operates on a 10-year window (2016-2026) with a universe sampled from the U.S. equity-options chain. The universe selection is proprietary. The architectural commitment for reproducibility is that the universe is large relative to the admissible set produced by the deterministic filter at any given decision time. The universe spans multiple sectors and capitalization tiers such that the back-testing programme does not over-fit to a single-sector or single-cap regime. The 10-year window includes multiple regime cycles. These cycles include pre-pandemic, pandemic, post-pandemic, the inflation cycle of 2022-2023, the AI rally of 2023-2024, and the Iran-war/energy-shock period of 2026.

Walk-forward protocol. The 10-year window is processed using a walk-forward back-test protocol that divides the timespan into a sequence of training windows and out-of-sample validation windows. Each successive window incorporates data that the previous validation window has already reported on. The protocol prevents look-ahead bias in the threshold-derivation step. Gateway-rule thresholds are derived solely from the training window and are then validated against the next out-of-sample window without threshold adjustment. The walk-forward step size and window length are calibrated to the deployment cadence.

Validation Sharpe ratio. Across the walk-forward back-test runs on the 10-year multi-sector validation universe, the gateway-rule set produces an annualized Sharpe ratio on the validation windows. This is a *back-tested* validation Sharpe that is derived from the historical back-testing programme used to calibrate gateway-rule thresholds (Phase 1 of the two-phase calibration) and not a live deployment Sharpe. The validation Sharpe substantiates the gateway-rule derivation methodology. Live-deployment Sharpe characterization is described in Section 3.2.5.

Distinct purposes of the two phases. The historical back-testing dataset is used for *gateway-rule derivation* and determines what the filter thresholds should be in order to produce a feasible candidate set with favourable historical risk-adjusted return characteristics. The live-deployment dataset (Sections 2.4.2 through 2.4.5) is used for *inference-layer recalibration*. It provides the training data on which the ML components in the learned-inference stage (Section 2.3) recalibrate their conditional-distribution estimates, pattern-recognition outputs, and within-feasible-set ranking. The two phases address different parts of the architecture. The back-test calibrates the deterministic-filter stage (the deterministic filter and constraint solver) and the live deployment calibrates the learned-inference stage (the ML in-

ference layer). This key distinction is explored further in Section 4.2.1.

2.4.7. Reproducibility and Implementation

The implementation is documented to a level appropriate for replication of architectural commitments.

Codebase composition. The framework consists of a Python rule engine (deterministic filter and joint MIP solver), integration examples connecting the rule engine to broker CSV ingestion and to ML-component inference, a unit-test suite, and visualization routines for outcome reporting. The rule engine and integration examples are operable on a single laptop with no cloud dependencies. The only network access required is for market-data ingestion at session entry and intraday cycle points.

Test coverage. The codebase is supported by 22 unit tests covering the rule engine's filter conditions, the joint MIP's constraint enforcement, the recursive-calibration cycle, and the audit-trail persistence layer. All tests pass on the deployment-version codebase. The test suite is intended for architectural-pattern verification. Empirical reliability is established primarily through the live-deployment record and not through test coverage.

Visualization outputs. The diagnostic and reporting outputs are generated as high-resolution PNG files in a 4-panel format. These panel show (i) position-level state, (ii) sector-level allocation, (iii) cycle-level cost-benefit attribution, and (iv) regime-classification diagnostics. A 4-panel format is a reporting standard adopted across the outputs to enable cross-cutting visualization of state at different aggregation levels.

2.4.8. Fundamentals Data: Petroleum Production Statistics

In addition to the live-trading dataset (Section 2.4.2-2.4.5) and the historical back-testing dataset (Section 2.4.6), the signal layer ingests a fundamentals-data input from the U.S. Energy Information Administration's monthly petroleum-production statistics. This input is the third tier of the data hierarchy (Section 2.4.5) and used by the framework's regime-classification component during the energy-sector regime cycles.

The role of data within the framework is *anchoring* as opposed to *predictive*. The live-trading and back-testing datasets together provide rich market-data signals (price, volume, volatility, sentiment, derivative-instrument metadata). They do not provide fundamental signals that are auditable independent of the market. By construction, every price-based or sentiment-based input is a function of how market participants are pricing or talking about the underlying. In contrast, EIA STEO Table 3a reports physical production statistics on a monthly cadence from government statistical authorities with full historical revisions. A primary-source physical-fundamentals input is methodologically appropriate and operationally auditable for a regime classifier whose purpose is to identify supply-shock, supply-stress, and supply-stable regimes in the energy sector.

From this production series, a sector-coupling diagnostic is derived that is con-

ceptually distinct from a Pearson correlation against crude price. Pearson correlation against price measures whether sector returns co-move with the oil-price series. The sector-coupling diagnostic incorporates production-statistics data (production levels, inventory changes, refinery utilization) into a regime-conditional classification of sector-flow direction. The diagnostic is used to inform the framework's sector-flow classification component (Section 2.3.1) and is anchored to the verifiable EIA production series rather than to price alone.

The cadence of the fundamentals input is monthly, set by the EIA's STEO release schedule (second Tuesday of each month). This is the lowest cadence in the framework's data hierarchy. The framework's higher-cadence inputs (daily price/volume, intra-session position state) operate at conventional resolution. The monthly fundamentals input enters the regime-classification component on each STEO release and persists between releases.

The fundamentals input strengthens the framework's audit-trail commitment (Section 2.4.4) by adding a verifiable public-record anchor to the regime-classification component. The empirical record can verify the EIA series independently and can verify the sector-coupling diagnostic against the same series. This matters methodologically because the fundamentals input also strengthens the framework's reproducibility commitment (Section 2.4.7). Any researcher attempting to reproduce the architectural pattern can use the same EIA series as the public anchoring point, while substituting their own implementation of the diagnostic.

3. Results

This section will describe the empirical illustration of the architecture through a single training-investment cycle (Section 2.2.1) and the real-money training-investment economic model used to frame it (Section 2.2.2).

3.1. Empirical Illustration: The AMD Trade

3.1.1. Multi-Regime Sequence

The AMD trade resides at the end of a multi-month sequence during which the framework's signal layer has been operating against the energy sector through March 2026. This enables the identification of flow-positive conditions in the energy-related underlyings admitted into the candidate set by the deterministic-filter stage. The AMD entry on 21 April 2026 was the autonomous rotation out of the energy-sector regime into the semiconductor-sector regime that the framework's signal layer subsequently identified.

In late April, the loop's re-estimation produced a further autonomous rotation indicating a move out of energy and into semiconductors. The AMD trade (documented below) was the first material allocation made under the semiconductor regime and realized a 120% return on the closed lot. The operator did not direct any of the regime transitions. Each transition was a downstream consequence of prior re-estimations that have propagated through the joint optimizer.

Figure 2 shows the framework's semiconductor-sector option-screening dash-

board, which produces the bounded, ranked candidate output from which the AMD trade was selected.

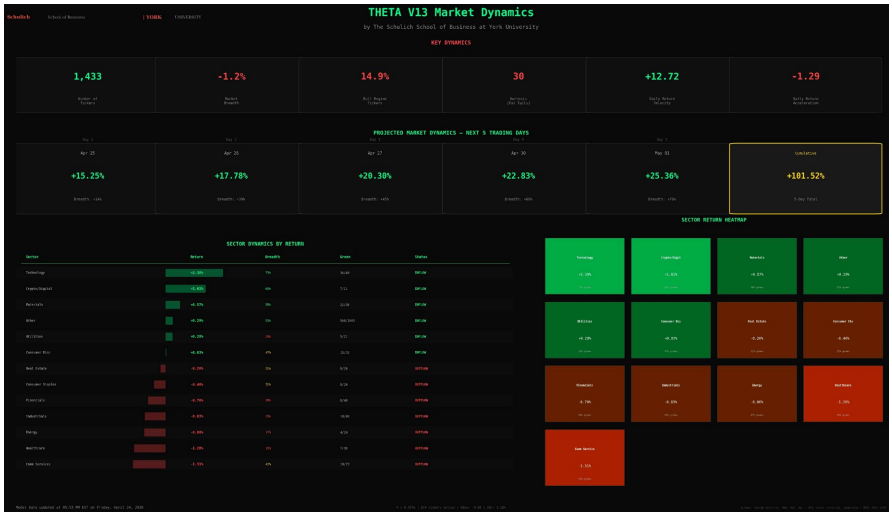

Figure 2. THETA V13 PS-010 ITM calls only dashboard, semiconductors sector, 21 April 2026 (17:58, market closed). [Source: dashboard run on laptop hardware (no cloud services); data source: Yahoo Finance market data and Bloomberg sector classification, both via free public APIs (Section 2.3.3)].

The figure shows the trading session on which the AMD trade (see Section 3.1.2 for description) was entered. The dashboard implements the FBYS architecture introduced in Section 2. The PS-010 ITM-only filter rule realizes the F operator (Section 2.2.5) and admits only the 87 in-the-money call contracts that satisfy the strategy's hard-constraint rules. The ATM line shows the OTM-blocked feasible-region boundary, with no OTM contracts admissible regardless of θ Score. The 2-8% ITM Distance Sweet Spot (shaded green) shows the SimDec joint-state cell window where the conditional edge-decay rate is most favourable (Section 2.3.5). Of the 87 admissible contracts, 33 occupy the Sweet Spot region. The θ Score colour gradient (red-to-green, 0-100 scale) is the learned-ranker output L (Section 2.3). The Top-8 PS-010 ITM Calls table is the ranked output feeding the non-discretionary exit-rule layer T (Section 2.3.4). In the table, AMD 270C is ranked first (θ Score 80, 5.1% ITM, 58 DTE) and AMD 260C is ranked fourth (θ Score 75, 8.6% ITM).

AMD was the framework's single highest-conviction semiconductor underlying on the trade-entry session. The dashboard shows contracts in the primary 58-DTE expiry band. The specific 22 May 31-DTE contract (see Section 3.1.2) was selected within the cost-efficient days-to-expiry sub-band. This represents a separable risk-management decision applied downstream of the framework's underlying-selection signal. The dashboard establishes that the framework ranks AMD first among semiconductor underlyings. The strike-and-expiry selection within that underlying will be described in Section 3.1.2.

The hallucination-removal property of the architecture is directly visible in this

dashboard layout. No OTM contract is reachable through any composition of L and T because the F operator excludes the entire OTM half-plane by construction (the dashed ATM line marks the boundary). A sector summary of the dashboard calculations shows that there were: 87 ITM calls, 0 puts (BLOCKED by the strategy's directional-bias rule), 33 Sweet Spot contracts, an average time-value of 51%, average leverage of 7×, average implied volatility of 61%, and 11 contracts with θ Score ≥ 70 (high-score threshold for trigger eligibility).

3.1.2. The AMD Trade Timeline

Entry (21 April 2026): AMD spot at the time of evaluation was approximately USD 303. The 22 May 2026 265-strike call was approximately 14% in the money, with delta approximately 0.85. The expiry was 31 days away, placing the contract inside the framework's cost-efficient days-to-expiry sub-band rather than the primary days-to-expiry band. The selection mechanism was the cost-efficiency sub-rule. Entry premium was approximately USD 38.20.

Hold (22 - 23 April 2026): AMD spot moved approximately +6% across the two-day window, taking the option premium to approximately USD 48.

Close on closed lot (24 April 2026): AMD spot advanced approximately +14.9% on the session, reaching approximately USD 348. The option premium reached approximately USD 84 per share—a +120% gain on the closed lot. The framework's profit-confirmation exit rule fired and the lot was closed.

Table 2 summarizes the trade's price points; intermediate intraday paths are approximate and reconstructed from broker statements.

Table 2. AMD spot and option premium evolution, 21-24 April 2026.

Date	Trading day	AMD spot (USD, approx.)	Option premium (USD, approx.)	Realized P&L on closed lot
21 April 2026	Entry	303	38.20	—
22 April 2026	Hold	315	46	—
23 April 2026	Hold	320	48	—
24 April 2026	Close	348	84	+120% (+CAD 4,100)

Per-tick execution prices are recorded in the broker transaction history. See the data availability statement.

3.1.3. Same-Session Portfolio Attribution

The 24 April 2026 close (the same session as the AMD partial close) produced a same-session aggregate portfolio gain of CAD 9,135 (6.16%) as displayed by the broker's official portfolio summary. This took the combined operating capital to CAD 157,454. The figure is verifiable from the broker's portfolio-summary display.

The benchmark context is verifiable from the same primary-source display. During the same session, the S&P 500 closed at 7,158.81 (+50.41, +0.71%), the

Dow Jones Industrial Average closed at 49,257 (−53.16, −0.11%), and the NASDAQ Composite closed in positive territory. Consequently, the framework's same-session return of +6.16% therefore represents approximately 8.68× the S&P 500's session return and approximately +5.45 percentage points of absolute outperformance versus the S&P benchmark on that day.

This single-session outperformance margin is a primary-source-verifiable data point on the framework's directional contribution beyond passive index exposure. The operating-capital base on the date in question (combined CAD 157,454 across the two regulated brokerage accounts) is documented in the same primary-source display. The +CAD 9,135 same-session aggregate gain is the broker-recorded realized plus mark-to-market figure across all positions in the two accounts on the trading session of 24 April 2026. This session aggregate is the broker's attribution across all positions in both accounts and is reported here as session context. It is not the benefit term of the train-ing-investment cycle of Section 3.2.2, which is computed on the AMD closed lot alone; the two figures are not additive.

3.1.4. Can This Outperformance Be Sustained on a Daily Basis?

A direct question follows from the +6.16%/+5.45-percentage-point outperformance reported in Section 3.1.3. Is this a daily phenomenon? Can the framework be expected to produce equivalent same-session outperformance on subsequent trading days, or in a compounded daily-return sense across a forward window? The answer is negative as the architecture is not designed to produce daily outperformance against a passive equity benchmark, and such performance should not be expected to be temporally stable and subject to stochastic variation in market conditions.

However, the architecture is designed to produce favourable conditional outcomes during regime cycles where the framework's signal layer has correctly identified flow-positive sectors and instruments. Conversely, it is designed to remove capital from the trading layer during regime cycles where the signal layer has not identified such favourable conditions. A daily-replicable +CAD 9,135 same-session gain would require continuous market-beating signals across all sessions. The framework does not claim this capability and the empirical record does not support it. The +6.16% same-session gain is a cycle-conditional outcome.

However, the architecture is designed to produce favourable conditional outcomes during regime cycles where the framework's signal layer has correctly identified flow-positive sectors and instruments. Conversely, it is designed to remove capital from the trading layer during regime cycles where the signal layer has not identified such favourable conditions. A daily-replicable +CAD 9,135 same-session gain would require continuous market-beating signals across all sessions. The framework does not claim this capability and the empirical record does not support it. The +6.16% same-session gain is a cycle-conditional outcome.

Selection bias of the reported session. The 24 April 2026 session is the session in which the framework's recalibrated exit logic fired correctly on a structurally

analogous position. It is the session that closes the XOM-to-AMD training-investment cycle of Section 3.2.2. We report the session because it is the cycle-closing observation, not because it is a representative session. A representative session within the deployment record would have an aggregate portfolio gain near zero, occasional moderate-positive sessions on cycle-firing events, and occasional moderate-negative sessions on the framework's training-cost positions. The +CAD 9,135 session is a cycle-firing observation.

Long-call convexity dispersion. The framework operates a long-call-only options strategy by hard architectural commitment (Filter 1). Long-call portfolios have an asymmetric return distribution of bounded losses and unbounded gains. However, most positions experience moderate-magnitude movement, while a minority of positions produce the large-magnitude outcomes that dominate the aggregate returns. Therefore, the dispersion of session returns is structurally large, with negative sessions occurring more commonly between the rare large-positive sessions. Consequently, a Sharpe-ratio-style daily-return projection would systematically misrepresent the strategy's actual risk profile [3] [14].

Regime-cycle structure. The training-investment economic model (in Section 3.2) predicts a compounding of *trained capability* across regime cycles, not a compounding of *session returns*. The cost-benefit cycle has a multi-week or multi-month structure, not a daily one. Within a single regime cycle, training-cost sessions and benefit-realization sessions are temporally separated. Reporting the benefit-realization session in isolation, then projecting its return forward as a daily-compounding rate, would conflate the cycle's economic structure with a false stationary assumption.

What the architecture predicts is that, once the training capability acquired during a given cycle has been correctly recalibrated into the inference layer, subsequent regime cycles will have higher cycle-firing rates conditional on the framework's signal-layer classification correctness. The empirical question of whether subsequent cycles produce outcomes consistent with this prediction is the principal research question of Section 4.3. The single cycle documented in Sections 3.1 and 3.2 represents a single observation. The architectural prediction can be evaluated against subsequent cycle outcomes as the deployment record accumulates. There is no claim that the prediction is supported by the single-cycle record. The actual claim is that the single-cycle record is consistent with the prediction and that the prediction is, itself, a falsifiable architectural commitment.

This distinction is explicitly made because the temptation to over-interpret a single high-outperformance session is one of the most common failure modes in finance ML reporting. The architectural argument of this paper does not require - and does not benefit from - daily-replicability claims. The architectural argument requires that: (i) the training-investment economic model produces measurable cost-benefit cycles; (ii) the AMD outcome is one such cycle's benefit realization; and, (iii) subsequent cycles are treated as independent evaluations, and will be evaluated on their own cost-benefit profiles.

3.1.5. Cross-Regime Transfer

From an AI/ML perspective, a property of the empirical sequence worth noting is that the XOM training data and the AMD pattern-firing share no security-level features. XOM is energy, while AMD is semiconductors. XOM is large-cap value, while AMD is large-cap growth. XOM is a quasi-utility, while AMD is a high-volatility growth name. The transferable feature was the structural pattern: deep-ITM long call, edge-decay condition firing on the framework's signal layer, exit-rule confirmation. The framework's recalibrated exit logic fired on the structural pattern, not on the underlying. Cross-regime transfer of structural patterns is what the architecture is designed to accomplish.

Cross-regime transfer of learning is expected theoretically for global ML systems but empirically uncommon in finance ML, where training and evaluation are typically conducted within the same asset class or factor regime [23] [33]. The XOM-to-AMD sequence represents one observation of the framework's cross-regime transfer. Whether it generalizes to other regime transitions is an open empirical question.

A second observation on cross-regime transfer comes from the AAPL 250-strike call session of 1 May 2026. The position, itself, (including its 20 April entry under the framework's pre-event signal, its 21 April reaction-and-override episode, and its 1 May 2026 close-at-peak outcome) is documented in Section 3.1.7. What Section 3.1.5 highlights is the cross-regime structural pattern. The AAPL position was entered before the framework had completed the recalibration cycle described in Section 3.2. The framework's signal layer, operating against a feature corpus that did not yet include the XOM-to-AMD training-investment cycle's outputs, classified the AAPL position as falling within the documented implied-volatility and days-to-expiry compliance bands. The position was entered. The 30 April Q2 earnings release with raised guidance, the 1 May rally, and the close-at-peak outcome together constitute a second cycle-firing event, this time on a technology underlying with a CEO-succession catalyst overlay. The structural pattern was again the transferable feature: deep-ITM long call, framework-signal-layer compliance, exit-rule confirmation. The cross-regime transfer in this case is from the energy regime (XOM training) to the technology regime (AAPL firing). The two cycle-firing events on AMD (24 April) and AAPL (1 May) within an 8-day window across different sectors is, itself, an observation worth flagging for subsequent-cycle empirical evaluation, though the sample size makes it inappropriate for any quantitative inference.

3.1.6. The AAPL Information-Catalyst Trade and Documented Operator Override

The AAPL position trajectory across the 20 April - 1 May 2026 window provides a primary-source-anchored illustration of the operator-judgement layer's interaction with the framework's signal layer. The position was entered on 20 April 2026 at framework-signal-layer compliance. An after-market disclosure that day produced a framework-signal reaction on 20 April at 16:51:16 EDT. An operator-

judgement-layer override was implemented in response. The position was retained through the 30 April fiscal Q2 earnings release with raised guidance and the 1 May intraday rally, closing at intraday peak on 1 May 2026 at +47.2% cost-to-peak return.

Phase 1: framework-aligned entry on 20 April 2026: A primary-source framework dashboard generated on 20 April 2026 at 15:37:27 EDT (approximately 23 minutes before regular-hours close and approximately 10 trading days before AAPL's fiscal Q2 2026 earnings release on 30 April 2026) classified the AAPL 250-strike 29 May 2026 call as compliant under the framework's documented filter set. The dashboard recorded that the candidate cleared the moneyness criterion (deep ITM at the time of evaluation), the days-to-expiry compliance band, the implied-volatility regime classification, and the position-size cap conditional on the operator's available capital. The framework's signal layer produced a positive ranking on the candidate. The operator-judgement-layer review of the dashboard was completed within the regular-hours window. The entry was implemented at an average cost of USD 24.60 per contract across the entry tranches recorded between 21 and 30 April 2026 in the broker transaction history. The entry was therefore framework-aligned in the Section 2.2.1 architectural sense. Specifically, (i) it satisfied the deterministic-filter stage's admissibility criteria, (ii) was selected by the joint MIP within the constrained feasible region, and (iii) was implemented at an operator-decision timestamp consistent with the dashboard recommendation.

The 20 April after-market disclosure: Apple Inc. announced after market close on 20 April 2026 that Tim Cook would transition from Chief Executive Officer to Executive Chair of the Board, effective 1 September 2026, with John Ternus, Senior Vice President of Hardware Engineering, succeeding as CEO on the same date. The board approval had occurred on Friday 17 April but had not been publicly disclosed prior to the 20 April after-market announcement (Form 8-K filed with the U.S. Securities and Exchange Commission on 20 April 2026 [34]). The operator's intraday position growth on 20 April, therefore, preceded the public disclosure by an interval that the broker timestamps establish as positive, but not precisely measurable from the records available in this study.

Phase 2: framework reaction and operator override: A framework dashboard generated on 20 April 2026 at 16:51:16 EDT (approximately 51 minutes after market close) ingested the after-market Tim Cook CEO-succession disclosure and produced a Phase 2 reaction on the AAPL position. The dashboard's revised classification flagged the position as elevated-uncertainty under the framework's signal-layer assessment of executive-succession events at major-cap technology underlyings approaching a fiscal-quarter earnings release. The framework's recommendation was a position trim of approximately 50% of the AAPL contracts at the next session's open, with the residual contracts retained pending the 30 April earnings release. The operator-judgement-layer decision was to override the trim recommendation and retain the full position. The override was implemented; the full position was retained through the 30 April earnings release; the 1 May intra-

day rally took the position to +47.2% cost-to-peak before close-at-peak. The override is documented in the framework's audit-trail at primary-source resolution (the dashboard timestamps, the recommendation text, and the operator-judgement-layer decision rationale have all been preserved).

The 30 April earnings release and the 1 May rally: Apple Inc. released its fiscal Q2 2026 earnings after market close on 30 April 2026 that indicated: earnings per share above consensus, services revenue at record level, and forward guidance raised. The disclosure was followed by a 1 May intraday rally that took the AAPL spot from USD 254 at 30 April close to a USD 268 intraday high on 1 May. The 250-strike call traded from a USD 24.60 cost basis to USD 36.21 intraday peak, which is a +47.2% cost-to-peak return on the retained position. The position was closed at intraday peak on 1 May. The realized return on the closed position is the +47.2% cost-to-peak figure alluded to in Section 1.

Leverage profile and position concentration: The Phase 2 override was not a marginal low-risk decision. Broker records show the AAPL 250-strike call concentration in the operating account at approximately 7.4% of combined operating capital at the override timestamp. This is approximately 1.5 times the framework's per-position cap of 5%. The over-cap concentration was implemented through a sequenced entry across the 21 April - 30 April window that pushed total cost basis to approximately CAD 11,650 against combined operating capital of approximately CAD 157,000. The framework's hard-constraint enforcement at the solver level (Section 2.2.4) would not have produced this concentration in the joint MIP allocation. The over-cap concentration is a primary-source-documented operator-judgement-layer departure from the framework's stated risk policy. The departure was within the operator's discretionary authority over the deployment account but was outside the framework's recommended allocation. The position closed at +47.2% cost-to-peak. The favourable outcome does not retrospectively justify the over-cap concentration (the framework's per-position cap is set on an ex-ante basis and is not contingent on outcome). The deployment record registers the departure as an operator-discretion event for the audit trail. The Section 4.2.2 limitations section returns to this episode as a documented case of operator discretion within the framework's deployment. It is available to the framework's recursive-calibration cycle as a training observation on the conditional-distribution properties of operator-judgement-layer departures from the stated risk policy.

Methodological framing: The AAPL trade documents two distinct interactions between the framework's signal layer and the operator's judgement layer across a single catalyst window. (1) The 20 April 15:37 entry was framework-aligned. (2) The 20 April 16:51 override was framework-contradicting. Both interactions are documented at primary-source resolution. The Phase 1 entry produced a position that subsequently captured a +47.2% cost-to-peak return. The Phase 2 override produced an over-cap concentration that would have produced a loss approximately 1.5 times the framework's per-position cap had the underlying not rallied. The two interactions illustrate the methodological boundary between framework-

recommended decisions and operator-discretion decisions within the framework's deployment context. The framework's audit-trail commitment (Section 2.4.4.1, Section 3.1.7, Section 4.2.2) is what makes both interactions visible to the framework's recursive-calibration cycle and to any subsequent evaluation of the deployment record. Without an audit-trail commitment, the +47.2% cost-to-peak outcome would be the visible record. With the audit-trail commitment, both the framework-aligned entry and the framework-contradicting override are visible.

3.2. Real-Money Training-Investment Economic Model

3.2.1. Training as Economic Activity

The framework's data-engineering methodology (Section 2.4) implies a particular economic framing of the training process. Conventional ML training is treated as an upfront fixed cost in which (i) the model is trained, (ii) the parameters are frozen, and (iii) inference proceeds at marginal-zero cost on the trained parameters. In contrast, the framework's training occurs continuously and at a non-zero per-cycle cost. Each new training data point requires the framework to *hold a real-money position* through the trajectory the data point represents. The position is not a simulation. The capital committed to it is real and the position's terminal state determines a real P&L outcome.

This implies an economic accounting structure for training that is unfamiliar in conventional ML but standard in research-and-development budgeting. *Training cost* is the realized loss accepted by deliberately retaining a position through a stress trajectory whose data the framework will use. The position could have been closed earlier at a profitable mark. Instead, the decision to retain it through subsequent decay is a methodological choice whose justification is the value of the training data the trajectory generates. *Training benefit* is the future realized gain on a structurally analogous position that the trained capability successfully closes at peak profitability. A close that would not have occurred without the training data the prior cost generated. *Net cycle return* is the difference between training benefit and training cost across a complete cost-benefit cycle. The cycles are not simultaneous. The cost is incurred at training time and the benefit is realized at a subsequent inference time when an analogous pattern fires.

3.2.2. The XOM-to-AMD Training Cycle

The framework's first complete training-investment cycle is documented as follows. The cost-incurrence event is the deliberate retention of an eight-contract long-call position on Exxon Mobil through 30 March - 24 April 2026, against a framework signal at 70.83% unrealized gain on 30 March 2026 that the position should be closed. The retention was implemented. The position decayed to terminal expiry on 1 May 2026. The realized loss on the eight-contract retained subcohort is broker-recorded at CAD 9,361.68. The retention was operationally deliberate and the framework's edge-decay enforcement signal was not silent (see Section 2.4.4.1 for the timestamped pre-cliff dashboard and the documented operator override).

The reversal materialized within the following two weeks. XOM fell from USD 176.41 (30 March) to USD 154.17 (after-hours trading on 24 April 2026) corresponding to a decline of 12.6% on the underlying that took the option from substantially in-the-money to deeply out-of-the-money relative to its 160-strike. The position's mark-to-market valuation declined commensurately. The 24 April 2026 framework dashboard recorded the position as edge-expired under the enforcement rule of Section 2.2.5. The operator-judgement-layer decision to retain the position into the 1 May 2026 terminal expiry generated a broker-recorded realized loss of CAD 9,361.68.

The XOM eight-contract retention was not isolated. Two structurally analogous long-call positions on Chevron and Cheniere Energy were retained on the same operator-judgement-layer decision basis (one contract each, same 1 May 2026 expiry, same March entry window, same edge-decay enforcement classification by 24 April 2026). The CVX 200-strike call decayed from a CAD 1,650.06 cost basis to CAD 1.36 terminal mark, which instigates a CAD 1,648.70 realized loss. The LNG 285-strike call decayed from CAD 3,048.01 cost basis to CAD 1.36 terminal mark, which triggers a CAD 3,046.65 realized loss. The full three-instrument cohort training cost is therefore CAD 9,361.68 + CAD 1,648.70 + CAD 3,046.65 = CAD 14,057.03 on the post-30-March retention decision. The cohort retention is documented at primary-source resolution in Section 2.4.4.1 (the theta-cliff dataset) and Section 3.1.6 (the operator-override documentation).

The cohort retention was operationally deliberate and the framework's exit signal was not silent. A primary-source framework dashboard generated on 20 April 2026 at 15:12:42 EDT (ten trading days before terminal expiry) explicitly recommended closure of the CVX 200-strike and LNG 285-strike positions with a model-stated conviction of 92% (Section 2.4.4.1). The operator's decision to retain the cohort through the subsequent ten-day window to terminal expiry on 1 May 2026 is, therefore, not a missed exit signal, but a documented, timestamped operator override. The override was a methodological choice taken with prior knowledge that the framework's exit signal had fired and with the explicit objective of obtaining the terminal-decay trajectory as training-and-calibration data for the framework's edge-decay enforcement architecture (Sections 2.2.5 and 2.4.4.1). The realized loss of CAD 14,057.03 is the cost of that deliberate dataset-acquisition decision. Under a counterfactual in which the framework's 20 April recommendation had been actioned at the timestamp it was generated, the cohort cost would have been substantially smaller. The magnitude of the cost differential is, itself, an empirical estimate of the value of the operator-judgement layer's discretion in this episode (negative-valued by the magnitude of the cliff loss in this instance).

This realized loss is the *training cost* of the cycle. It is not a description of a failed position reframed as training data. It is a deliberate methodological choice, with the operator's explicit acknowledgement at the retention moment that the position would not be closed at intermediate profitable marks, but would instead be retained to enrich the training dataset.

The benefit-realization event followed three weeks later. On 21 April 2026, the framework recommended a long-call option position on Advanced Micro Devices: AMD 22 May 2026 265-strike call, 5 contracts entered at USD 38.20. The framework's recalibrated exit logic—updated against the XOM retention's training-data record—fired correctly on 24 April 2026 at AMD spot approximately USD 348, generating a +120% return-on-capital on the closed lot (broker-recorded realized gain approximately CAD 4,100 in three trading days). The trained capability fired on a structurally analogous position in a different sector and at a different scale. The cross-sector firing is treated in Section 3.1.5.

The cycle's economics:

Training cost: CAD 14,057.03 realised loss across the cohort (XOM CAD 9,361.68 + CVX CAD 1,648.70 + LNG CAD 3,046.65), each computed as broker-recorded cost basis less terminal mark per Table 1.

Training benefit (AMD closed lot at +120%): CAD 4,100 realised gain.

Net cycle return: CAD 4,100 − CAD 14,057.03 = −CAD 9,957.03 in realised terms.

First-cycle break-even: not achieved; the cycle stands at approximately −CAD 9,957 in realised terms pending subsequent cycles.

3.2.3. The Live-Phase Status of This Cycle—A Methodological Clarification

The training-investment cycle described above is not a Phase 1 back-test but a *Phase* 2 *live-deployment cycle*. Phase 1 (the historical back-testing programme described in Sections 2.4.6 and 4.2.1) derives the three framework gateway rules (i) the Filter 3 thresholds, (ii) the position-size cap, and (iii) the correlation-group concentration cap. Phase 2 then operates that post-Phase-1 rule set live using real money in regulated retail brokerage accounts. The XOM-to-AMD cycle is a Phase 2 cycle. Both the training cost and the training benefit reported in Section 3.2.2 were incurred in real money in regulated brokerage accounts and are recorded in the broker's transaction history. There is no simulation in this cycle, no historical-data replay, and there is no model of fills, slippage, or liquidity. Phase 2 cycle data is what feeds the framework's inference-layer recalibration (Section 2.3) under the training-investment economic model.

We elaborate more extensively on the distinction between Phase 1 (historical back-test) and Phase 2 (live deployment) in Section 4.2.1.

3.2.4. Compounding of Trained Capability

The training-investment economic model implies a particular form of long-run compounding. For example, the model does not predict that the framework will produce a 120% return on every cycle. Instead, it predicts that the framework's accumulated training capability, once recalibrated correctly into the inference layer, will produce higher cycle-firing rates conditional on the framework's signal-layer classification correctness across subsequent regime cycles. The compounding is the cumulative effect of trained capability firing across cycles, not the per-

cycle return magnitude. Whether the predicted compounding pattern emerges in the deployment record is the principal research question of Section 4.3.

Single-cycle observations like the XOM-to-AMD sequence are first data points on this mechanism. Whether the predicted compounding holds across many cycles and many regimes is the empirical question Section 4.3's research programme is designed to answer.

3.2.5. Rolling Sharpe Trajectory across the Deployment Window

The framework's rolling 30-day Sharpe ratio across the deployment window from 27 October 2025 to 24th April 2026 (a window covering the immediate pre-deployment baseline and the active deployment phase) exhibits a non-monotonic but directionally-consistent trajectory. It runs from approximately 0.30 in the first observed window (28 October 2025) through approximately 0.85 by the end of January 2026. It contains a transient drawdown to approximately 0.45 in early February 2026, with a subsequent recovery to approximately 1.10 by 2 April 2026. The trajectory is consistent with the recursive-calibration mechanism predicted by the training-investment economic model: Sharpe rises as the framework's accumulated training capability is recalibrated correctly into the inference layer, drops during stress-period training-cost incurrence, and recovers as cycle-firing events realize the trained capability.

We treat the rolling Sharpe trajectory as suggestive empirical evidence consistent with the recursive-calibration mechanism described in Section 3.2. Three caveats apply. First, the 5-month window is a single observed trajectory; subsequent calibration cycles may not exhibit the same pattern. Second, the rolling Sharpe is computed on the framework's two-account aggregate operating capital and is therefore conditional on the operator's contemporaneous risk policy and the operator-judgement-layer decisions documented in Section 2.4.4.1, Section 3.1.7, and Section 4.2.2. Third, the trajectory is reported in the manuscript as descriptive evidence of the deployment record rather than as a performance claim: we do not claim the framework has demonstrated learning-over-time; we report that the Sharpe trajectory is in the direction the training-investment economic model predicts and is consistent with the architectural commitments documented in Section 3.2. The Sharpe trajectory's right edge—the 1.10 reading at 2 April 2026—is the deployment-record's most recent calibrated state at the time of submission. Whether the trajectory continues, reverses, or stabilizes is the principal empirical question of Section 4.3. The trajectory is treated as a single piece of evidence among several, not as the headline claim.

Statistical evidence of learning-over-time under the recursive-calibration mechanism would require: (i) a longer deployment window, ideally covering multiple cost-benefit cycles; (ii) a counterfactual benchmark Sharpe trajectory under identical operator capital and risk policy but without the framework's recursive-calibration cycle; and (iii) an effect-size estimate that distinguishes calibration-mechanism-driven Sharpe improvement from contemporaneous market-condition effects. The single-cycle observation reported here meets none of these con-

ditions. The Section 4.3 research programme reconciles the longitudinal evidence accumulation required to convert the Sharpe trajectory into statistical evidence of the calibration mechanism.

A specific high-resolution episode anchors the right edge of the Sharpe trajectory. The 24th of April generated a same-session aggregate gain of +CAD 9,135 (+6.16%) on the AMD partial close. The session is the cycle-closing observation for the XOM-to-AMD training-investment cycle (Section 3.2.2) and is the highest-information-content single session in the manuscript's deployment record. The 24th April session is not representative of the deployment record's typical session. It is the cycle-firing observation. The Section 3.1.4 discussion of daily replicability explicitly addresses this distinction.

4. Discussion

This section develops the cross-industry generalization of the architectural pattern (Section 2.3.1) and establishes the substantive limitations of the empirical record (Section 2.3.2).

4.1. Why Bounding Matters: Cross-Industry Generalization of the Pattern

The architectural pattern documented in Sections 2.2 through 2.4 is not specific to the portfolio management of options. What it does provide is: (i) a deterministic-filter stage admitting only rule-compliant candidates into a constrained feasible region, (ii) a SimDec sensitivity-analysis layer that decomposes input-space variance into joint-state combinations, (iii) a bounded ML inference layer whose outputs cannot escape the rule set (Section 2.3.4), and (iv) a non-discretionary rule trigger that fires on joint-state conditions rather than on free-form ML inference. This architecture generalizes to any decision domain in which the following four conditions co-occur: (i) the cost of an ML hallucination is high (regulatory, financial, safety, or human); (ii) the underlying dynamics are non-linear and conditional on the system's own shifting state; (iii) the audit-trail traceability or regulatory explainability is required; and (iv) the deployment constraints (locally-executed inference, commodity hardware, public or auditable data) preclude pure end-to-end ML.

The central novelty of the pattern lies not in the use of any individual tool, but in the way the tools constrain one another. SimDec acts as a deterministic decomposition layer that governs the stochastic deep-learning engine. The deep-learning components produce predictions, classifications, and rankings. At the same time, the SimDec layer forces these outputs to map back to explainable joint-state cells before they can cause a non-discretionary rule to fire. The closed loop feature is what contributes the hallucination-removal property of Section 2.3.4. All AI inferences are necessarily validated by the decomposition layer **before** they reach the decision interface, not afterwards.

The following six industry contexts illustrate the nature of the transfer. They

are presented in approximately descending order of global market size (2024 figures) and their relative ordering should not be construed as any explicit form of market-prioritization claim.

Predictive maintenance in aerospace and manufacturing (global manufacturing value-added, USD 16.8 trillion, 2024; UN/UNIDO). A *conditional degradation curve* computed under SimDec from the joint state of operating regime and component class produces a non-discretionary maintenance trigger that fires on empirically-derived joint-state evidence rather than calendar position.

Healthcare and pharmacokinetics (global healthcare market, USD 11 trillion, 2024; WHO/CMS aggregated). An *effective half-life* parameterization, mapped through joint-state partitioning of patient biomarkers, allows deep-learning efficacy and dosing predictions while ensuring that any AI-generated recommendation violating the joint-state-derived rule set is mathematically excluded from the feasible region, not merely flagged.

Adverse clinical decision-making in acute care systems (global healthcare expenditure ~USD 10 trillion annually, World Bank/WHO estimates). A harm-generation function defined over patient-state, diagnostic uncertainty, and treatment-path dependency yields a conditionally activated intervention trigger under uncertainty decomposition (SimDec joint-state layer) that is driven by joint-state deterioration signals—such as physiological instability, diagnostic inconsistency, and escalation probability—rather than protocol-driven timing or calendar-based review intervals.

Supply-chain logistics, particularly cold-chain perishables (global logistics market, USD 10 trillion, 2024; Precedence Research aggregated). An *effective value-decay* parameterization feeding non-discretionary rerouting and dynamic-pricing triggers - bounded by the SimDec joint-state layer - produces auditable autonomous decisions in which the explainability of the rerouting matters as much as the rerouting itself.

Energy-grid management and battery storage (global energy-sector consumer spending, USD 8 trillion; grid and generation investment, USD 1.4 trillion annually, 2024; IEA *World Energy Investment 2025*). An *effective storage decay* parameterization under SimDec joint-state partitioning produces non-discretionary discharge triggers when the joint state crosses a cliff cell - directly analogous to the options edge-decay rule.

Telecommunications and network infrastructure (global telecommunications market, USD 2 trillion, 2024; ITU/GSMA aggregated). A bounded AI/ML pipeline using SimDec joint-state partitioning over network-state inputs produces auditable, non-discretionary traffic-shaping triggers compatible with SLA-driven regulatory accountability for routing decisions.

Nuclear-reactor operations and isotope decay management (global nuclear power-plant market, USD 35 billion, 2024; IAEA operational data, 417 reactors/377 GW). A SimDec joint-state partitioning over reactor-state inputs forces any AI-generated control recommendation - including non-discretionary triggers - to map

back to an empirically-validated joint-state cell, making AI adoption defensible in reactor operations.

In each domain the structural argument is identical to the documented options case. Pure end-to-end ML carries hallucination risk unacceptable to regulatory or operational stakeholders. Conversely, a hybrid architecture with deterministic filtering, SimDec joint-state partitioning, and bounded ML inference operating in series produces decisions whose reasoning is auditable from first principles. The framework documented in this study is an existence proof: a bounded AI/ML deployment pattern of this form can be built, deployed, and operated against real-money outcomes in a regulated environment, on commodity hardware, with full audit-trail traceability. The detailed per-industry structural mappings and the authoritative sources for the 2024 global-market-size figures are fully developed in [35]. These include the decay analogue, the conditional scaling factors, the SimDec joint-state inputs, the hallucination-removal property, and the regulatory and operational stakes for each destination domain.

4.2. Limitations

Several limitations of this study are discussed in the subsequent sections.

4.2.1. Two-Phase Calibration: Back-Testing and Live Deployment Distinguished

The framework's methodological characterization requires careful clarification as the term "back-testing" has been applied liberally throughout the finance-ML literature and is sometimes asserted to be present or absent in ways that mischaracterize a hybrid methodology. Consequently, we wish to be explicit about its application to the framework's *two-phase* methodology and the distinct role of back-testing within it.

Phase 1 provides the historical back-testing for gateway-rule derivation. The framework's gateway rules (the Filter 3 thresholds for in-the-money compliance, the position-size cap, the correlation-group cap, the regime-classification thresholds) were derived from a 10-year historical back-testing programme over a multi-sector validation universe. The back-testing programme was a multi-cycle aggregate on the historical record. The validation Sharpe ratio (Section 2.4.6) is classified as back-test evidence, not live-deployment evidence. The Phase 1 back-testing methodology is appropriate for gateway-rule threshold derivation because gateway-rule thresholds are slow-moving architectural parameters that should be calibrated against multi-cycle historical evidence rather than against single-cycle live evidence. Phase 1 is not the framework's empirical record; it is the framework's threshold-derivation methodology.

Phase 2 is the live-deployment phase used to produce the empirical, inference-layer, recalibration record reported in Sections 3.1 and 3.2. It was run over the period from October 2025 through April 2026. The Phase 2 record is single-realization evidence on the framework's training-investment economic model with one cycle (XOM-to-AMD), one operator, one capital base, one regulatory juris-

diction, and one regime cycle (energy-sector volatility under geopolitical stress). The Phase 2 evidence weight is therefore not comparable to the Phase 1 evidence weight, despite both phases informing the framework's architecture. Phase 1 calibrates the gateway-rule thresholds. Phase 2 calibrates the inference-layer ranking within the constrained feasible region. Conflating their evidence weights produces incorrect inferences about the framework's expected return profile.

The two phases serve distinct purposes that are both individually appropriate to their respective methodological roles. Back-testing on a 10-year window is appropriate for gateway-rule threshold derivation. Live deployment on a single regime cycle is appropriate for inference-layer recalibration evidence. Neither phase substitutes for the other; both are required for the architectural pattern to be implementable. The methodological commitment in this paper is to evaluate each phase against its own evidence standard. Phase 1 is assessed against the multi-cycle back-test validity criteria. Phase 2 is evaluated against the single-cycle live-deployment provenance criteria.

For empirical purposes, the Sharpe validation supplies back-test evidence on the gateway-rule threshold derivation. The 120% AMD outcome provides single-realization evidence on the inference-layer recalibration cycle. These two pieces of evidence are not interchangeable and, taken separately, neither supports a generalized expected-return claim. The framework's expected-return profile remains an open empirical question as discussed in Section 4.3.

4.2.2. Operator-Judgment Calibration

The framework's filter thresholds (in-the-money depth, delta, days-to-expiry band, position-size cap, correlation-group concentration cap) reflect operator judgement on the framework's risk policy as appropriate for the deployment context. Namely, context in which there is a single-operator deployment using two regulated retail brokerage accounts in which the operator bears direct fiduciary responsibility for the deployed capital. Different operators with different risk policies, different fiduciary contexts, or different capital bases would specify different thresholds. Since the threshold values are operator-specific, the framework's architectural commitments are operator-agnostic. The single-operator evidence base means that the framework's empirical record cannot, by itself, distinguish framework-attributable outcomes from operator-judgement-attributable outcomes. The audit-trail commitment (Sections 2.4.4.1 and 3.1.7) makes operator-judgement-layer interactions discernable at primary-source resolution. This visibility addresses the attribution problem at the documentation level, but does not resolve it at the inference level. Multi-operator validation is identified in Section 4.3 as a research-programme requirement. A specific case of operator-judgement-layer interaction is documented at primary-source resolution in Section 3.1.7. It could be observed that the AAPL Phase 2 override, in which the operator-judgement layer overrode the framework's Phase 2 trim recommendation, generated an over-cap concentration of approximately 7.4% of operating capital against the framework's per-position cap of 5%. The favourable 47.2% cost-to-peak outcome

on the position does not retrospectively justify the over-cap concentration - the framework's per-position cap is set on an ex-ante basis. The episode is documented as a primary-source-anchored case of operator-discretion departure from the framework's stated risk policy and is available to the framework's recursive-calibration cycle as a training observation.

4.2.3. ML-Component Status

The specific ML-component formulations referenced in Section 2.3, the proprietary calibration values referenced in Section 2.2, and the agent topology that connects the framework's sub-systems are the subject of a patent application filed with Innovation York at York University. The architectural pattern documented in this paper (deterministic-first/learned-second; real-money training-investment calibration, edge-decay enforcement, operator-judgement-layer audit trails) is not patent-protected and is offered for replication and adaptation by other researchers under the IP-scoping convention.

4.2.4. Single-Operator, Single-Capital-Base Evidence

The empirical record presented in this study is attributable to a single operator, with a single capital base (approximately CAD 160,000 in two operating accounts), and employing a single regulated brokerage. Consequently, the framework's architectural commitments could produce dissimilar outcomes when operating under different operators, larger capital bases, or different brokerage execution environments. In principle, the architectural contributions are operator-agnostic, capital-agnostic, and brokerage-agnostic. Demonstrating this prevailing agnosticism in practice would require multi-operator and multi-environment evaluations.

4.2.5. Sample Size and Statistical Power

A single training-investment cycle (XOM-to-AMD) represents one observation. The cumulative cost-benefit accounting (training cost CAD 14,057 across the XOM/CVX/LNG cohort, cumulative benefit CAD 13,235, net approximately CAD −822) provides one realization of one cycle. Whether the cost-benefit profile is robust across many cycles cannot be inferred from this single observation. The research programme outlined in Section 4.3 specifies the multi-cycle evaluation protocol required. Observation count is constrained by cycle duration rather than by computational cost: the framework executes locally with no per-cycle inference expense, so the accumulation of further cycles is limited only by the multi-week structure of the cost-benefit cycle itself.

4.3. Research Programme

To convert the architectural contributions into a more generalizable, empirically-validated framework would require a four-stage evaluation protocol. The research programme for this conversion is outlined using longitudinal evidence accumulation.

Stage 1 involves a multi-cycle training-investment evaluation tracked over at least $N = 10$ complete training-investment cycles (where N is the number of cycles required to distinguish framework-attributable outcomes from operator-judgement-layer outcomes at conventional statistical significance). Each cycle should record: (i) The cost-incurrence event which is a deliberate retention through a stress trajectory whose data the framework needs; (ii) The cost-amount represented by the broker-recorded realized loss on the retained position; (iii) The benefit-realization event covering the subsequent firing of the trained capability on a structurally analogous position; (iv) the benefit-amount represented by the broker-recorded realized gain on the cycle-firing position; (v) The cycle-net-return calculated as the benefit minus cost; and, (vi) The cycle-duration providing the cost-incurrence to benefit-realization interval. The empirical question is whether the cycle-net-return distribution is positive in expectation, and how the distribution evolves as the framework's training capability accumulates. The training-cost stamping point would need to be defined precisely. The XOM/CVX/LNG cohort documented in Section 2.4.4.1 had its training-cost stamped on 24 April 2026 - the date the operator-judgement-layer override was implemented against the framework's edge-decay enforcement signal. The stamping point is the operationally meaningful timestamp because it is the moment the cost was committed. The realization timestamp (1 May 2026 terminal expiry) is the moment the cost was recorded. Subsequent cycles should record both timestamps and report the cost-incurrence-to-benefit-realization interval based on the stamping point. The stamping-point definition is a research-programme commitment, not a methodological mechanism within the framework.

In Stage 2, the architectural commitments are asset-class-agnostic in principle. Deploying the framework on an alternative asset class (e.g. single-stock equities, ETF rotation, or futures) would test whether the architectural design produces equivalent operational properties in different feature distributions. The deterministic-first/learned-second commitment, the hard-constraint solver-level enforcement, and the hallucination-removal property are not specific to options. Their behaviour in non-options environments remains an open empirical question.

In Stage 3, the multi-operator validation framework's filter threshold reflects operator judgement. Deploying the framework with different operator calibration choices would test whether the architectural commitments survive variation in calibration. For the architectural approach considered in this paper, the operational properties (reproducibility, auditability, hallucination removal, training-investment cost-benefit cycle structure) will hold for any reasonable calibration choice within the stated architectural commitments.

Stage 4 would involve a comparative, out-of-sample evaluation. Direct comparisons with alternative AI/ML portfolio-management methodologies (*i.e.* deep reinforcement-learning agents, classical Markowitz allocations, end-to-end neural optimizers) on a forward-looking, out-of-sample window would judge whether the architectural commitments produce favourable outcomes against various

well-established alternatives. These comparisons should hold capital base, asset class, and operating-window constant across the methodologies, with each methodology's terminal outcome attributable to its specific architectural choices rather than to operator differences.

5. Conclusions

This paper has provided a deterministic-first/learned-second architectural design for AI-driven options portfolio management. The architecture confines its ML components to a layer downstream from a deterministic filter and a hard-constraint solver. This placement ensures that no ML hallucination can circumvent the established rule set. The architectural pattern is generalizable to AI/ML deployment contexts that require auditable deterministic gating prior to learned-inference application. Section 4.1 developed six business-aligned cross-industry deployment contexts: (i) predictive maintenance in manufacturing (USD 16.8 trillion global TAM); (ii) healthcare and pharmacokinetics (USD 11 trillion); (iii) supply-chain logistics (USD 10 trillion); (iv) energy-grid management (USD 8 trillion); (v) telecommunications (USD 2 trillion), and (vi) nuclear-reactor operations (USD 35 billion). The framework's calibration combines two phases that serve distinct purposes. Phase 1 derives the gateway-rule thresholds for the deterministic filter under a walk-forward back-test protocol using historical back-testing over a 10-year multi-sector universe. Phase 2 recalibrates the inference layer using continuous CSV-format position snapshots using a live deployment over a 6-month real-money window. The two-phase calibration is structurally appropriate to the deterministic-first/learned-second architecture, while being methodologically uncommon in published AI/finance literature.

The architecture is illustrated over an empirical training-investment cycle. The deliberate retention of an eight-contract long-call position on XOM generated training data at a real-money cost of CAD 14,057. The recalibrated exit logic of the framework subsequently fired correctly on the structurally analogous semiconductor position of AMD at a 120% realized return on the closed lot. The same trading session produced an aggregate portfolio gain of 6.16% versus 0.71% for the S&P 500. The XOM-to-AMD sequence is a within-framework cost-benefit cycle on the deployment record of the same operator.

A major contribution of this study is the architectural design. This architecture provides (i) a hard-rule enforcement at the solver level, (ii) a training-investment calibration as an economic model, (iii) an edge-decay enforcement as a non-discretionary architectural primitive, and (iv) the operator-judgement-layer audit trails. The architectural commitments can be assessed on their stated terms: specifiability, implementability, and evaluability against subsequent-cycle outcomes. The empirical illustration provided a single-realization of evidence over one cycle. However, the prescribed research programme would necessitate longitudinal evidence evaluated over many trading-session cycles, multiple operators, and multiple asset classes. The empirical record presented in this paper establishes the base

point for that programme and will be expanded upon more fully in future research.

Acknowledgements

This research was supported in part by grant OGP0155871 from the Natural Sciences and Engineering Research Council. This study received no external corporate funding or third-party capital. The data-capture deployment that generated the empirical record used the author G.M.'s personal capital held in regulated retail brokerage accounts. During the preparation of this manuscript, the author(s) used Anthropic Claude (Sonnet 4.5 and Opus 4.7) for the purposes of editorial review, structural revision, citation management, and tracked-change manuscript management. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

Author Contributions

G.M.: conceptualization, methodology, software, formal analysis, investigation, data curation, writing - original draft, writing - review and editing, visualization, project administration. D.G.: methodology, validation, writing - review and editing. S.I.: conceptualization, methodology, validation, writing - review and editing. J.S.Y.: methodology, validation, writing - review and editing, supervision, funding acquisition.

Data Availability Statement

Author G.M. can supply replication code in Python or R upon request. This code is independent of the data-capture pipeline and uses only the closed-form Black-Scholes pricer, standard implied-volatility solvers, and the broker-CSV record reported in Table 1. All SIMDEC code is freely available, open-source, in Python, R, Matlab, and Julia from: https://github.com/Simulation-Decomposition, A no-code-required, web-based dashboard can be accessed at: https://simdec.io/. An open-access SIMDEC reference book can be downloaded from: https://doi.org/10.4324/9781003453789. Position-level CSV snapshots, transaction-history records, and broker-summary screenshots referenced in this paper are held by G.M. and are available to researchers under reviewer-confidentiality protocols. Public-record references (EIA petroleum-production statistics, broker public-record entries for the equity underlyings discussed) are independently verifiable and are cited at the relevant locations in the body.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

[1] Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., *et al*. (2023) Survey of Hallucination in Natural Language Generation. *ACM Computing Surveys*, **55**, Article No. 248. https://doi.org/10.1145/3571730

[2] Rudin, C. (2019) Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead. *Nature Machine Intelligence*, **1**, 206-215. https://doi.org/10.1038/s42256-019-0048-x

[3] Melville, G. and Yeomans, J. (2026) Decomposing the Theta Cliff: A SIMDEC Filtering of Asymptotic Time-Decay in Long-Call Options with a Real-Money Intraday Illustration. *AI*, **7**, Article No. 257. https://doi.org/10.3390/ai7070257

[4] Doshi-Velez, F. and Kim, B. (2017) Towards a Rigorous Science of Interpretable Machine Learning. https://arxiv.org/abs/1702.08608

[5] Jiang, M., Huang, T., Guo, B., Lu, Y. and Zhang, F. (2025) Enhancing Robustness in Large Language Models: Prompting for Mitigating the Impact of Irrelevant Information. In: Mahmud, M., Doborjeh, M., Wong, K., Leung, A.C.S., Doborjeh, Z. and Tanveer, M., Eds., *Neural Information Processing. ICONIP* 2024. *Communications in Computer and Information Science*, Vol. 2295, Springer, 207-222. https://doi.org/10.1007/978-981-96-7030-7_15

[6] Zhu, D.H., Xiong, Y.J., Zhang, J.C., Xijiong, X. and Xia, C.M. (2025) Understanding before Reasoning: Enhancing Chain-of-Thought with Iterative Summarization Pre-Prompting. https://doi.org/10.48550/arXiv.2501.04341

[7] Kozlova, M., Ahola, A., Roy, P.T. and Yeomans, J.S. (2025) Simple Binning Algorithm and SimDec Visualization for Comprehensive Sensitivity Analysis of Complex Computational Models. *Journal of Environmental Informatics Letters*, **13**, 38-56.

[8] Kozlova, M., Moss, R.J., Yeomans, J.S. and Caers, J. (2024) Uncovering Heterogeneous Effects in Computational Models for Sustainable Decision-Making. *Environmental Modelling & Software*, **171**, Article ID: 105898. https://doi.org/10.1016/j.envsoft.2023.105898

[9] Marzban, S. and Lahmer, T. (2016) Conceptual Implementation of the Variance-Based Sensitivity Analysis for the Calculation of the First-Order Effects. *Journal of Statistical Theory and Practice*, **10**, 589-611. https://doi.org/10.1080/15598608.2016.1207578

[10] Sobol, I.M. (2001) Global Sensitivity Indices for Nonlinear Mathematical Models and Their Monte Carlo Estimates. *Mathematics and Computers in Simulation*, **55**, 271-280. https://doi.org/10.1016/s0378-4754(00)00270-6

[11] Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., *et al*. (2008) Global Sensitivity Analysis: The Primer. Wiley. https://doi.org/10.1002/9780470725184

[12] Helo, J., Kozlova, M., Roy, P. and Yeomans, J.S. (2026) Regional Variance-Based Sensitivity Analysis for Complex Models. Risk Analysis.

[13] Black, F. and Scholes, M. (1973) The Pricing of Options and Corporate Liabilities. *Journal of Political Economy*, **81**, 637-654. https://doi.org/10.1086/260062

[14] Hull, J.C. (2018) Options, Futures, and Other Derivatives. 10th Edition, Pearson.

[15] Carr, P. and Wu, L. (2009) Variance Risk Premiums. *Review of Financial Studies*, **22**, 1311-1341. https://doi.org/10.1093/rfs/hhn038

[16] Bakshi, G. and Kapadia, N. (2003) Delta-Hedged Gains and the Negative Market Volatility Risk Premium. *Review of Financial Studies*, **16**, 527-566.

https://doi.org/10.1093/rfs/hhg002

[17] Merton, R.C. (1973) Theory of Rational Option Pricing. *The Bell Journal of Economics and Management Science*, **4**, 141-183. https://doi.org/10.2307/3003143

[18] Schwab Center for Financial Research (2026) Theta Decay in Options Trading: Three Strategies. Charles Schwab. https://www.schwab.com/learn/story/theta-decay-options-trading

[19] Cao, J. and Han, B. (2013) Cross Section of Option Returns and Idiosyncratic Stock Volatility. *Journal of Financial Economics*, **108**, 231-249. https://doi.org/10.1016/j.jfineco.2012.11.010

[20] Natenberg, S. (2015) Option Volatility and Pricing: Advanced Trading Strategies and Techniques. 2nd Edition, McGraw-Hill.

[21] Gu, S., Kelly, B. and Xiu, D. (2020) Empirical Asset Pricing via Machine Learning. *The Review of Financial Studies*, **33**, 2223-2273. https://doi.org/10.1093/rfs/hhaa009

[22] Kolm, P.N., Tütüncü, R. and Fabozzi, F.J. (2014) 60 Years of Portfolio Optimization: Practical Challenges and Current Trends. *European Journal of Operational Research*, **234**, 356-371. https://doi.org/10.1016/j.ejor.2013.10.060

[23] López de Prado, M. (2018) Advances in Financial Machine Learning. John Wiley & Sons.

[24] Taleb, N.N. (2007) The Black Swan. The Impact of the Highly Improbable. Random House.

[25] Markowitz, H. (1952) Portfolio Selection. *The Journal of Finance*, **7**, 77-91. https://doi.org/10.1111/j.1540-6261.1952.tb01525.x

[26] Ignizio, J.P. (1976) Goal Programming and Extensions. Lexington Books.

[27] Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., *et al*. (2025) A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. *ACM Transactions on Information Systems*, **43**, Article No. 42. https://doi.org/10.1145/3703155

[28] Courant, R. (1943) Variational Methods for the Solution of Problems of Equilibrium and Vibrations. *Bulletin of the American Mathematical Society*, **49**, 1-23. https://doi.org/10.1090/s0002-9904-1943-07818-4

[29] Fiacco, A.V. and McCormick, G.P. (1968) Nonlinear Programming: Sequential Unconstrained Minimization Techniques. Wiley. (Reprinted as SIAM Classics in Applied Mathematics 4, 1990)

[30] Coello, C.A. (2002) Theoretical and Numerical Constraint-Handling Techniques Used with Evolutionary Algorithms: A Survey of the State of the Art. *Computer Methods in Applied Mechanics and Engineering*, **191**, 1245-1287. https://doi.org/10.1016/s0045-7825(01)00323-1

[31] Asness, C.S., Frazzini, A. and Pedersen, L.H. (2019) Quality Minus Junk. *Review of Accounting Studies*, **24**, 34-112. https://doi.org/10.1007/s11142-018-9470-2

[32] Owen, A.B. (2013) Variance Components and Generalized Sobol' Indices. *SIAM/ASA Journal on Uncertainty Quantification*, **1**, 19-41. https://doi.org/10.1137/120876782

[33] Krauss, C. (2016) Statistical Arbitrage Pairs Trading Strategies: Review and Outlook. *Journal of Economic Surveys*, **31**, 513-545. https://doi.org/10.1111/joes.12153

[34] Apple Inc. (2026) Form 8-K, Current Report Pursuant to Section 13 or 15(d) of the Securities Exchange Act of 1934, Filed with the U.S. Securities and Exchange Commission on 20 April 2026, Available via SEC EDGAR.

[35] Melville, G. and Yeomans, J.S. (2026) Architectural Transferability in Bounded AI: Five Conditions for Transfer from Options Trading to Regulated Decision Domains.

Abbreviations

The following abbreviations are used in this manuscript.

AAPL	Apple Inc, NASDAQ
AI	Artificial Intelligence
AMD	Advanced Micro Devices Inc, NASDAQ
ATM	At-The-Money
BiLSTM	Bidirectional Long Short-Term Memory
BS	Black-Scholes
CAD	Canadian Dollar
CVX	Chevron Corp, NYSE
DTE	Days-to-Expiration
CSV	Comma Separated Value Files
ITM	In-The-Money
LNG	Chevron Corp, NYSE
IV	Implied Volatility
MIP	Mixed Integer Programming
ML	Machine Learning
OTM	Out-of-The-Money
P&L	Profit and Loss
SLA	Service Level Agreements
USD	United States Dollar
XOM	Exxon Mobil Corp, NYSE
XAI	Explainable Artificial Intelligence