Inverse Delayed Reinforcement Learning for Limit Order Books: A Multi-Strategy EM–MaxEnt Approach for Inferring Latency-Driven Trading Behaviour

Shivam Hedau

Msc, University of Edinburgh

S.Hedau@sms.ed.ac.uk

December 3, 2025

Abstract

We present a complete mathematical, algorithmic, and empirical framework for analyzing high-frequency **Limit Order Books (LOBs)** by modelling them as asynchronous games with delayed states and actions, and employing **Inverse Delayed Reinforcement Learning (IDRL)** to infer trader strategies executed under unknown and heterogeneous latencies. Building on the theory of asynchronous decision processes, we introduce a **Time-Bracketed EM–MaxEnt IRL** formulation that jointly estimates (i) probabilistic state–action responsibilities across delay windows, (ii) latent strategic mixtures governing trader behaviour, and (iii) reward parameters for multiple strategy types. Two theoretical results are established: **Theorem A**, proving concavity, monotonicity, and convergence of Time-Bracketed EM–MaxEnt IRL for a single latent strategy; and **Theorem B**, extending these guarantees to a *multi-strategy mixture-of-experts* model equipped with history-position softmax gating.

Empirically, we apply the full IDRL pipeline to real high-frequency LOB data using a 19-dimensional state representation capturing microprice dynamics, multi-level volume imbalance, CVI, WAP deviations, and signed order-flow features. Delay responsibilities are inferred over a discrete five-lag window (converted to approx. 60–350 ms), revealing pronounced heterogeneity across latent strategies. We find that: (i) expected delays differ systematically across the four recovered strategies, ranging from fast liquidity-responsive behaviour to slower inventory-driven reaction patterns; (ii) delays exhibit strong dependence on microstructure features such as spread, imbalance, and recent trade pressure; (iii) multi-strategy IDRL uncovers distinct reward geometries consistent with market making, liquidity taking, and latency-sensitive arbitrage; and (iv) the strategy-gating mechanism identifies persistent yet state-dependent switching patterns across market regimes. Together, these results demonstrate that IDRL provides a robust and interpretable framework for reverse-engineering heterogeneous latency-driven behaviours in high-frequency markets.

1 Introduction

In modern financial markets, time is no longer measured in minutes or seconds—it is measured in microseconds and nanoseconds. Electronic trading infrastructures have evolved into globally interconnected systems where millions of messages travel through fibre-optic cables, microwave towers, millimetre-wave networks, and laser communication links [13]. In this environment, the

ability to react quickly to new information becomes just as critical as the information itself [6]. A trader's profitability, competitiveness, and even survival depend on how fast they can detect and respond to fleeting opportunities.

To appreciate this, consider two market participants observing the same limit order book (LOB) of a highly liquid instrument. One is a retail trader routed through multiple intermediaries—brokerage platforms, risk-check engines, smart order routers, and geographically dispersed servers—each adding computational and communication delays [4]. The other is a high-frequency trading (HFT) firm colocated inside the exchange's data centre, connected directly to raw market feeds and equipped with highly optimised hardware designed for sub-microsecond processing [23]. Suppose both participants observe a sudden imbalance between the best bid and ask levels, signalling a short-term price movement. For the colocated HFT firm, this imbalance is immediately actionable: orders are submitted and executed before the retail trader's screen even refreshes. For the slower participant, by the time their order reaches the exchange, the imbalance has vanished and the opportunity is gone.

This discrepancy illustrates the critical role of *latency*—the delay between when market information becomes available and when a participant can act on it. Latency arises from physical distances, networking limitations, computation times, queueing delays, and exchange-level matching processes [7]. In high-frequency markets, these delays are not only unavoidable but also heterogeneous across participants, instruments, and market conditions. Even the smallest timing differences can determine queue priority, execution price, or whether a trade occurs at all. Moreover, heterogeneous latencies create structural asymmetries in who can reliably exploit information. Traders with faster access systematically benefit at the expense of those operating on stale states, a phenomenon widely referred to as *latency arbitrage* [11].

Despite its importance, latency is poorly captured by most traditional modelling frameworks. Classical microstructure models implicitly assume that all traders share a common, synchronised view of the LOB [9]. Execution-lag models simplify latency to a constant or exogenous parameter. Reinforcement learning (RL) and optimal control frameworks generally assume that each action is based on the most recent observed state [26]. All these simplifications fail in the high-frequency environment, where actions are rarely executed at the exact moment they are decided. Instead, the true decision is made on a *past* state, and the mapping from decisions to executions is obscured by unknown delays. Without explicitly modelling delayed decision processes, it is impossible to properly understand the behaviour of different trader types, infer their strategic objectives, or estimate the market-wide impact of latency.

This paper addresses that gap by modelling the LOB as an *asynchronous game* in which trader actions are generated based on delayed observations, and these delays are treated as latent variables to be inferred from data. States are represented using features extracted from the LOB, while actions correspond to order submissions, cancellations, or executions. The central inverse problem becomes:

Given an observed action, which past state most likely triggered the decision leading to it?

To solve this problem, we propose a full theoretical and computational framework based on **Inverse Delayed Reinforcement Learning (IDRL)**. Our key idea is to generalise Maximum Entropy Inverse Reinforcement Learning (MaxEnt IRL) by introducing a *time-bracketed* treatment of delays, where each action is softly aligned with a window of candidate past states [29]. Using the Expectation–Maximisation (EM) algorithm, we assign probabilistic responsibilities to each delay candidate and infer a reward function that best explains the observed behaviour [10]. This probabilistic alignment fundamentally departs from existing approaches, which either assume perfect

synchrony or collapse delay into a single parameter. Instead, our method uncovers the full structure of heterogeneous and strategy-dependent delays across market participants.

Building on this, we extend the framework to accommodate multiple latent strategies through a mixture-of-experts model equipped with history-position gating. This enables us not only to estimate delays but also to identify the underlying behavioural regimes that produce them. Concretely, the proposed IDRL framework jointly estimates:

1. probabilistic state–action–delay responsibilities across a finely discretised delay grid,
2. multiple latent trading strategies and their temporal activation patterns,
3. reward parameters governing each strategy's behaviour.

We provide rigorous theoretical guarantees for the entire framework. **Theorem A** establishes concavity, monotonicity, and convergence of the EM–MaxEnt IRL procedure in the single-strategy setting, while **Theorem B** extends these results to the multi-strategy mixture-of-experts case, including gated responsibilities across strategies and delays.

This paper therefore contributes a unified, theoretically grounded, and empirically validated approach for reverse-engineering delayed decision processes in high-frequency markets. It provides a principled way to understand how traders with different latencies perceive and react to market signals, how heterogeneous delays shape strategic behaviour, and how these dynamics manifest in real-world LOB data. The remainder of the paper develops the full mathematical formulation, computational implementation, empirical analysis, and interpretation of our findings.

2 Literature Review

The proposed Inverse Delayed Reinforcement Learning (IDRL) framework builds on several lines of research spanning market microstructure, latency modelling, inverse reinforcement learning, maximum-entropy methods, and mixture-of-experts architectures [9, 13]. This section reviews the most relevant contributions and highlights the gaps that motivate our asynchronous, delay-aware formulation for high-frequency limit order books (LOBs).

2.1 Market Microstructure and Limit Order Books

The modelling of electronic limit order books has been an active research area for over two decades. Early structural models characterise order arrivals through point processes, Markov chains, or queue-reactive mechanisms, with seminal contributions including the multi-agent modelling tradition of Lux and Marchesi [20] and empirical behavioural formulations such as Mike and Farmer [22]. More formal queue-based representations were developed in the stochastic LOB framework of Cont, Stoikov and Talreja [8], while Hawkes-driven approaches were later explored by Abergel and Jedidi [3]. A comprehensive overview of LOB mechanics and agent-based perspectives is presented in Abergel et al. [2], which documents empirical regularities in order flow, spreads, and depth profiles.

These models, while successful in capturing price formation and queue behaviour, typically assume synchronous information arrival and immediate reaction. As electronic markets evolved toward microsecond-scale competition, this assumption became increasingly unrealistic, with documented latency asymmetries in modern trading systems [6, 4]. Contemporary market microstructure studies recognise the importance of latency and asynchronous decision-making, yet most frameworks treat latency as an exogenous constant or infrastructure parameter rather than a behavioural choice [11, 23, 21, 7].

2.2 Latency, Asynchrony, and High-Frequency Trading

Latency plays a central role in high-frequency trading (HFT), influencing execution quality, information advantages, and competitiveness. Empirical and theoretical studies (e.g. [11, 6, 4, 23, 21, 7]) examine latency arbitrage, stale-quote sniping, and the implications of technological speed races. These papers emphasise that even small differences in reaction time can alter the order in which trades arrive, creating exploitable informational asymmetries.

More recent work in asynchrony formalises market participants as agents operating on different internal clocks or reacting to delayed information. However, such models generally impose parametric delay structures or analyse stylised settings rather than inferring delays directly from data. To date, no prior empirical framework jointly estimates latent delays, strategies, and reward functions from real LOB trajectories.

This gap motivates our use of probabilistic delay windows, allowing delays to be state-dependent, stochastic, and strategy-specific.

2.3 Inverse Reinforcement Learning and Maximum Entropy Methods

Inverse reinforcement learning (IRL) was introduced by Ng and Russell as a framework for recovering reward functions from observed expert behaviour [25]. Later, Ziebart formalised the maximum-entropy principle for IRL [29], introducing log-linear policies and convex optimisation over feature expectations. MaxEnt IRL remains one of the most widely used IRL formulations due to its tractability, probabilistic interpretation, and strong theoretical guarantees [1].

Subsequent work includes causal variants, deep IRL, adversarial training, and multi-agent extensions, but almost all assume synchronous decision-making: each action is taken in response to the immediately preceding state. This assumption is incompatible with high-frequency markets, where actions may be generated using stale information because of variable delays from processing, queueing, or network latency [5].

The literature contains no IRL variant that accounts for *unknown, heterogeneous, strategy-dependent delays*. Our time-bracketed MaxEnt formulation fills this gap by introducing delay-conditioned feature expectations and EM-based responsibility assignment [10, 12].

2.4 Latent Strategy Modelling and Mixture-of-Experts

Heterogeneous-agent modelling has long been recognised as essential for understanding market microstructure [9, 13]. Traders differ in risk tolerance, speed, inventory management, and informational objectives. Mixture-of-experts models provide a principled approach for decomposing behaviour into latent components governed by a gating network [14, 15]. Such models have been successfully applied in behavioural modelling, robotics, and clustered policy learning [1], and offer a natural extension of IRL to multi-strategy settings.

In finance, multi-agent or multi-strategy inference has remained largely heuristic. Existing clustering approaches typically rely on k-means, hierarchical clustering, or Gaussian mixtures applied directly to observed states and actions [19]. These methods do not ground strategy identification in reward-based behaviour, nor do they incorporate the timing structure inherent in high-frequency markets [5].

Our mixture-of-experts IDRL framework extends MaxEnt IRL by combining latent strategies, delay-conditioned feature alignment, and softmax gating based on history-position features. This yields a theoretically grounded method for inferring both strategic heterogeneity and timing behaviour simultaneously [16].

2.5 Gaps in Existing Literature

Despite the extensive research across market microstructure, latency, and reinforcement learning, several important gaps remain:

- **No existing IRL method accounts for heterogeneous, stochastic delays.** All classical IRL methods assume perfect alignment between states and actions [25, 29].
- **Latency is rarely modelled as a behavioural variable.** Prior work treats delays as exogenous constraints rather than quantities that depend on market conditions and strategic objectives
- **No framework jointly infers reward functions, strategies, and delays.** Existing microstructure models focus on order flow; IRL models focus on reward inference; clustering models focus on strategy discovery [8, 2].
- **Mixture-of-experts IRL has not been applied to financial markets.** Prior work in behavioural modelling has not addressed latency-driven, asynchronous environments like LOBs [16].

These limitations underscore the need for an integrated framework such as IDRL, which unifies delayed-action modelling, maximum-entropy IRL, and multi-strategy inference into a single coherent system capable of reverse-engineering high-frequency behaviour directly from data.

3 Mathematical Framework

In this section, we formalise the high-frequency limit order book (LOB) environment, the structure of delayed actions, and the modelling choices that constitute the foundation of Inverse Delayed Reinforcement Learning (IDRL). Our goal is to define a financially meaningful asynchronous decision process in which each observed action is interpreted as the outcome of a latent optimisation performed on a past market state. The formulation integrates microstructure intuition, statistical modelling, and reinforcement learning principles, and prepares the ground for the EM–MaxEnt estimation procedure developed in later sections.

3.1 Market Environment and Information Flow

High-frequency electronic markets generate a stream of timestamped messages—limit orders, cancellations, modifications, and trade executions—that update the state of the limit order book [13]. Let

$$s_t \in \mathbb{R}^d, \quad t = 1, 2, \ldots, T,$$

denote the LOB state at time t after processing the t-th message. The state s_t is not merely a snapshot of prices and quantities; it is a high-dimensional summary encoding supply–demand pressure, liquidity distribution, order flow imbalance, and short-term predictive signals relevant to market participants.

In real markets, the information flow is inherently asynchronous. Market participants observe the LOB with heterogeneous delays induced by network propagation, feed handlers, transport buffers, exchange gateways, matching engines, and local computation [6]. As a result, a trader's internal decision is almost never aligned with the state immediately preceding the action that is eventually observed. Instead, the decision is made using a stale state $s_{t-\Delta}$ whose delay Δ is unknown.

IDRL aims to reverse-engineer which past state is responsible for each observed action, and what reward function made that action the most likely choice.

3.2 LOB State Representation

A key component of the IDRL framework is the construction of a financially meaningful feature map for LOB states. We adopt a $d = 19$ dimensional representation that captures fundamental microstructure primitives known to influence short-term profitability, adverse selection, and execution quality [2]. Formally, we define:

$$s_t = (x_{t,1}, x_{t,2}, \ldots, x_{t,19})^\top,$$

where the components include:

- **Spread and Microprice**: capture instantaneous liquidity tightness and imbalance between bid and ask depths.
- **Volume Imbalance (Level 1 and Level 5)**: measure pressure at near-touch and deeper levels, predictive of short-term price movements.
- **CVI Depth1 / Depth5**: cumulative volume indicator summarising liquidity distribution across levels.
- **WAP Diff Depth5**: weighted-average price differentials revealing slippage and flow pressure.

- **Signed Trade Flow**: recent aggressive activity indicating directional pressure.
- **Liquidity Absorb / Cancel Imbalance**: captures order book resilience and hidden liquidity.

Each feature is financially motivated: liquidity providers care about queue positions and adverse selection; liquidity takers respond to imbalance and flow signals; latency arbitrageurs react to microprice and cross-venue pressure. By working directly with such features, the reward model captures realistic strategic incentives.

3.3 Action Representation

Let $\mathcal{A}$ denote the set of feasible LOB actions. Typical high-frequency actions include:

$$a_t \in \mathcal{A} = \{\text{market buy, market sell, limit buy, limit sell, cancel, modify}\}.$$

Each action type reflects a distinct strategic goal [13]:

- **Market orders** sacrifice spread capture for immediacy; used in latency-sensitive or directional strategies.
- **Limit orders** provide liquidity, rely on queue priority, and face inventory/adverse-selection risks.
- **Cancellations** are protective manoeuvres, especially in volatile conditions or when stale quotes become risky.
- **Modifications** reposition orders to manage inventory or capture micro-price signals.

These actions exhibit different sensitivities to delay. For example, latency arbitrageurs predominantly use aggressive orders when a predictive signal is fresh. Liquidity providers cancel stale quotes when adversarial flow appears. Thus, the action space and delay structure are tightly coupled, making actions essential inputs to the reward function.

3.4 Delayed Action Model

To formalise asynchrony, let τ_t denote the execution timestamp of observed action a_t. Let:

$$\Delta \in \{\Delta_1, \Delta_2, \ldots, \Delta_K\}$$

be a discrete set of possible delays (e.g. 0–2000 ms with 50 ms granularity). For each a_t, the action is presumed to have been decided based on one of the states in the delay window:

$$\mathcal{W}_t = \left\{s_{t-\Delta_1}, s_{t-\Delta_2}, \ldots, s_{t-\Delta_K}\right\}.$$

The true delay that links state to action is treated as a latent variable. HFT reality strongly motivates this modelling choice [21]: delays arise from gateway queueing, network jitter, exchange-level microbursts, and strategy-specific inference pipelines. They are non-constant, stochastic, and vary across traders and market regimes.

3.5 Reward Function and Feature Map

We assume each trading strategy optimises a reward that is *linear in features*:

$$R_\theta(s, a) = \theta^\top f(s, a),$$

where $f(s, a)$ is a joint feature representation capturing state–action interactions and $\theta \in \mathbb{R}^m$ are reward weights.

This form is widely used in microstructure theory and high-frequency optimal control [25]:

- Linearity captures a weighted trade-off between slippage, queue priority, adverse selection, and inventory cost.
- $f(s, a)$ includes action-specific penalties (e.g. crossing the spread), inventory adjustments, or trend-following signals.
- The MaxEnt IRL interpretation treats θ as the implicit objective a strategy tries to optimise.

From a financial perspective, θ identifies strategic archetypes. A strategy with positive weight on imbalance and microprice signals behaves like a latency arbitrageur; one with negative weight on aggressive price movement resembles a liquidity provider avoiding adverse flow.

3.6 Maximum Entropy Action Likelihood

Given a candidate state $s_{t-\Delta}$, the probability that a trader selects action a_t is modelled using MaxEnt IRL [29]:

$$p(a_t \mid s_{t-\Delta}, \theta) = \frac{\exp\big(\theta^\top f(s_{t-\Delta}, a_t)\big)}{Z(s_{t-\Delta}, \theta)},$$

where

$$Z(s_{t-\Delta}, \theta) = \sum_{a' \in \mathcal{A}} \exp\Big(\theta^\top f(s_{t-\Delta}, a')\Big)$$

is the partition function.

MaxEnt is well-suited for financial modelling because trader behaviour is noisy, path-dependent, and influenced by heterogeneous beliefs [1]. Traders exhibit near-rational tendencies without being perfectly optimal, consistent with maximum entropy principles.

3.7 Likelihood with Unknown Delays

Since the delay is unobserved, the likelihood of a_t marginalises over all possible delays [27]:

$$p(a_t \mid \theta) = \sum_{\Delta_k} \pi_{\Delta_k}\, p(a_t \mid s_{t-\Delta_k}, \theta),$$

where π_{Δ_k} is a prior over delays (uniform or exponentially decaying).

For a full trajectory:

$$\mathcal{L}(\theta) = \sum_{t=1}^{T} \log \left(\sum_{k=1}^{K} \pi_{\Delta_k} \frac{\exp\big(\theta^\top f_{t,\Delta_k}\big)}{Z_{t,\Delta_k}(\theta)} \right).$$

This forms the core objective of IDRL. The responsibilities associated with delays ($\gamma_{t,\Delta}$) naturally emerge as latent variables under an EM interpretation [28].

4 EM–MaxEnt IRL: Expectation–Maximisation for Delayed Inverse Reinforcement Learning

Having defined the asynchronous decision-making structure underlying high-frequency trading, we now derive the estimation procedure for Inverse Delayed Reinforcement Learning (IDRL). Because the true delay linking each observed action to its causal state is unobserved, the likelihood involves latent variables. Expectation–Maximisation (EM) provides a natural and principled framework for estimating reward parameters θ in the presence of these latent delays [27]. The EM–MaxEnt formulation introduced in this section forms the backbone of the theoretical guarantees (Theorem A and Theorem B) and the multi-strategy extensions developed later.

4.1 Latent Delay Variables and Complete-Data Likelihood

For each observed action a_t, let the latent categorical variable z_{t,Δ_k} indicate whether delay Δ_k is responsible for the state that generated the action. Formally,

$$z_{t,\Delta_k} = \begin{cases} 1, & \text{if action } a_t \text{ was generated from state } s_{t-\Delta_k}, \\ 0, & \text{otherwise,} \end{cases} \qquad \sum_{k=1}^{K} z_{t,\Delta_k} = 1.$$

Under the Maximum Entropy model introduced earlier, the complete-data likelihood factorises as

$$p(a_t, z_{t,\Delta_k} \mid \theta) = \left[\pi_{\Delta_k} \cdot \frac{\exp(\theta^\top f_{t,\Delta_k})}{Z_{t,\Delta_k}(\theta)} \right]^{z_{t,\Delta_k}},$$

where π_{Δ_k} is the prior delay probability and $f_{t,\Delta_k} = f(s_{t-\Delta_k}, a_t)$.

Taking logs and summing over all t, we obtain the complete-data log-likelihood:

$$\log p(a_{1:T}, z_{1:T} \mid \theta) = \sum_{t=1}^{T} \sum_{k=1}^{K} z_{t,\Delta_k} \left[\log \pi_{\Delta_k} + \theta^\top f_{t,\Delta_k} - \log Z_{t,\Delta_k}(\theta) \right].$$

Since z_{t,Δ_k} is latent, we proceed with the EM algorithm.

4.2 E-Step: Delay Responsibilities

In the E-step, we compute the posterior probability that delay Δ_k caused action a_t, given the current estimate $\theta^{(n)}$ [28]:

$$\gamma_{t,\Delta_k}^{(n)} = \mathbb{E}[z_{t,\Delta_k} \mid a_t, s_{1:T}, \theta^{(n)}].$$

Applying Bayes' rule,

$$\gamma_{t,\Delta_k}^{(n)} = \frac{\pi_{\Delta_k} \exp\left(\theta^{(n)\top} f_{t,\Delta_k}\right) / Z_{t,\Delta_k}(\theta^{(n)})}{\sum_{j=1}^{K} \pi_{\Delta_j} \exp\left(\theta^{(n)\top} f_{t,\Delta_j}\right) / Z_{t,\Delta_j}(\theta^{(n)})}.$$

Financial interpretation. The responsibilities γ_{t,Δ_k} quantify how likely it is that a trader reacted to LOB state $s_{t-\Delta_k}$. High responsibilities at small delays correspond to fast-reacting arbitrage or aggressive strategies, while higher responsibilities at larger delays indicate slower strategies such as passive liquidity provision or retail-type behaviour. The E-step therefore "softly assigns" each action to a distribution over candidate historical states, providing a probabilistic alignment of asynchronous behaviour.

4.3 Expected Sufficient Statistics

With responsibilities in hand, we compute delay-weighted empirical feature counts:

$$\bar{f}(\theta^{(n)}) = \sum_{t=1}^{T} \sum_{k=1}^{K} \gamma_{t,\Delta_k}^{(n)} f_{t,\Delta_k}.$$

At the same time, model-expected feature counts under the MaxEnt policy are:

$$\hat{f}(\theta^{(n)}) = \sum_{t=1}^{T} \sum_{k=1}^{K} \gamma_{t,\Delta_k}^{(n)} \sum_{a' \in \mathcal{A}} p(a' \mid s_{t-\Delta_k}, \theta^{(n)}) f(s_{t-\Delta_k}, a').$$

These quantities generalise the standard feature-matching structure of MaxEnt IRL to delayed, asynchronous environments.

4.4 M-Step: Reward Parameter Update

The M-step maximises the expected complete-data log-likelihood [29]:

$$\mathcal{Q}(\theta \mid \theta^{(n)}) = \sum_{t=1}^{T} \sum_{k=1}^{K} \gamma_{t,\Delta_k}^{(n)} \left[\theta^\top f_{t,\Delta_k} - \log Z_{t,\Delta_k}(\theta) \right].$$

Taking gradients,

$$\nabla_\theta \mathcal{Q}(\theta \mid \theta^{(n)}) = \bar{f}(\theta^{(n)}) - \hat{f}(\theta),$$

so the M-step solves:

$$\theta^{(n+1)} = \arg\max_\theta \mathcal{Q}(\theta \mid \theta^{(n)}).$$

This is a convex optimisation problem because $\mathcal{Q}$ is strictly concave in θ. In practice, we perform gradient ascent:

$$\theta^{(n+1)} \leftarrow \theta^{(n)} + \eta \left(\bar{f}(\theta^{(n)}) - \hat{f}(\theta^{(n)}) \right),$$

where η is a step size.

Financial interpretation. The M-step adjusts the reward weights so that actions taken in the dataset appear increasingly optimal relative to the inferred past states. If traders consistently respond to order book imbalance by submitting aggressive orders, then imbalance-related features will receive higher reward weights. If cancellations predominantly occur when microprice predicts adverse selection, those features will gain negative weight. Thus, θ captures strategic incentives implicit in the observed behaviour.

4.5 Monotonic Improvement of Likelihood

Each EM iteration increases the marginal log-likelihood [27]:

$$\mathcal{L}(\theta^{(n+1)}) \geq \mathcal{L}(\theta^{(n)}),$$

because:

$$\mathcal{L}(\theta) = \log p(a_{1:T} \mid \theta) \geq \mathcal{Q}(\theta \mid \theta^{(n)})$$

and the M-step maximises $\mathcal{Q}$. This monotonicity is formalised in Theorem A.

4.6 Interpretation within Financial Microstructure

The EM–MaxEnt IRL formulation captures the inherent uncertainty and heterogeneity of high-frequency behaviour:

- **E-step:** identifies which historical states are consistent with the timing of each action; effectively reconstructing the trader's observation timeline.
- **M-step:** fits a reward that rationalises observed behaviour under these reconstructions.

This combination enables IDRL to infer both *when* traders reacted to signals (delay inference) and *why* they acted (reward inference). It is therefore a powerful abstraction for modelling high-frequency trading as an asynchronous decision-making process with heterogeneous latencies.

5 Theoretical Guarantees for Single-Strategy IDRL

In this section, we establish the core theoretical guarantees for the time–bracketed EM–MaxEnt Inverse Reinforcement Learning (IDRL) model in the single–strategy case. We show that the M-step objective is strictly concave, that each EM iteration increases the marginal log-likelihood, and that the algorithm converges to a stationary point of the likelihood function. These results extend classical guarantees for Maximum Entropy IRL to the delayed-action and asynchronous state-alignment setting.

5.1 Preliminaries

For convenience, recall the expected complete-data log-likelihood:

$$\mathcal{Q}(\theta \mid \theta^{(n)}) = \sum_{t=1}^{T} \sum_{k=1}^{K} \gamma_{t,\Delta_k}^{(n)} \left[\theta^\top f_{t,\Delta_k} - \log Z_{t,\Delta_k}(\theta) \right],$$

and the marginal log-likelihood:

$$\mathcal{L}(\theta) = \sum_{t=1}^{T} \log \left(\sum_{k=1}^{K} \pi_{\Delta_k} \frac{\exp(\theta^\top f_{t,\Delta_k})}{Z_{t,\Delta_k}(\theta)} \right).$$

The responsibilities $\gamma_{t,\Delta_k}^{(n)}$ satisfy:

$$\gamma_{t,\Delta_k}^{(n)} \geq 0, \qquad \sum_{k=1}^{K} \gamma_{t,\Delta_k}^{(n)} = 1.$$

We now establish the concavity of $\mathcal{Q}$, the monotonic improvement property, and convergence of EM.

5.2 Lemma 1: Log-Partition Function is Convex

Lemma 1. *For any fixed state s and feature map $f(s,a)$, the log-partition function*

$$\log Z(s,\theta) = \log \left(\sum_{a' \in \mathcal{A}} \exp(\theta^\top f(s,a')) \right)$$

is convex in θ.

Proof. The log-partition function $\log Z$ is the cumulant-generating function of an exponential family. It is a classical result that such functions are convex because their Hessians correspond to the covariance of the feature vector under the induced MaxEnt distribution:

$$\nabla_\theta^2 \log Z(s,\theta) = \mathrm{Cov}_{p(a'|s,\theta)}[f(s,a')] \succeq 0.$$

Thus $\log Z(s,\theta)$ is convex in θ for all s. □

5.3 Lemma 2: $\mathcal{Q}$ is Strictly Concave

Lemma 2. *The expected complete-data log-likelihood $\mathcal{Q}(\theta \mid \theta^{(n)})$ is a strictly concave function of θ.*

Proof. Write

$$\mathcal{Q}(\theta) = \sum_{t,k} \gamma_{t,\Delta_k} \left[\theta^\top f_{t,\Delta_k} - \log Z_{t,\Delta_k}(\theta) \right].$$

The first term, $\theta^\top f_{t,\Delta_k}$, is linear in θ. The second term, $-\log Z_{t,\Delta_k}(\theta)$, is concave by Lemma 1.

Since $\gamma_{t,\Delta_k} \geq 0$ and sum to 1 over k, $\mathcal{Q}$ is a nonnegative weighted sum of concave functions, plus a linear function, hence concave.

Strict concavity holds because the Hessian of $-\log Z$ is strictly negative definite whenever at least two actions have nonzero probability, which is always true in MaxEnt IRL. Therefore $\mathcal{Q}$ is strictly concave. □

5.4 Lemma 3: EM Lower-Bound Property

Lemma 3. *For any θ and any $\theta^{(n)}$, the function*

$$\mathcal{Q}(\theta \mid \theta^{(n)})$$

is a lower bound on the marginal log-likelihood $\mathcal{L}(\theta)$, with equality at $\theta = \theta^{(n)}$.

Proof. This follows directly from Jensen's inequality applied to the latent variable posterior distribution. The derivation is identical to standard EM theory:

$$\mathcal{L}(\theta) = \log p(a_{1:T} \mid \theta) = \log \sum_z p(a_{1:T}, z \mid \theta)$$

$$= \log \sum_z q(z) \frac{p(a_{1:T}, z \mid \theta)}{q(z)} \geq \sum_z q(z) \log \frac{p(a_{1:T}, z \mid \theta)}{q(z)} = \mathcal{Q}(\theta \mid \theta^{(n)}),$$

with equality when $q(z) = p(z \mid a_{1:T}, \theta^{(n)})$. □

5.5 Theorem A: Concavity, Monotonicity, and Convergence

Theorem 1 (Theorem A: Convergence of Single-Strategy IDRL)**.** *For the single-strategy time-bracketed EM–MaxEnt IRL model:*

1. *The M-step objective $\mathcal{Q}(\theta \mid \theta^{(n)})$ is strictly concave in θ.*
2. *Each EM iteration monotonically increases the log-likelihood:*

$$\mathcal{L}(\theta^{(n+1)}) \geq \mathcal{L}(\theta^{(n)}).$$

3. *The sequence $\{\theta^{(n)}\}$ converges to a stationary point of $\mathcal{L}(\theta)$.*

Proof. **(1) Strict concavity of the M-step.** Lemma 2 proves $\mathcal{Q}$ is strictly concave in θ. Thus the M-step has a unique maximiser.

(2) Monotonic improvement. By Lemma 3, for each θ:

$$\mathcal{L}(\theta) \geq \mathcal{Q}(\theta \mid \theta^{(n)}),$$

with equality at $\theta = \theta^{(n)}$. Since the M-step maximises $\mathcal{Q}$:

$$\mathcal{Q}(\theta^{(n+1)} \mid \theta^{(n)}) \geq \mathcal{Q}(\theta^{(n)} \mid \theta^{(n)}) = \mathcal{L}(\theta^{(n)}).$$

Therefore:

$$\mathcal{L}(\theta^{(n+1)}) \geq \mathcal{Q}(\theta^{(n+1)} \mid \theta^{(n)}) \geq \mathcal{L}(\theta^{(n)}).$$

(3) Convergence to a stationary point. Since $\mathcal{L}$ is bounded above (finite action set) and increases monotonically, the sequence $\{\mathcal{L}(\theta^{(n)})\}$ converges. The EM algorithm is known to converge to a stationary point of the likelihood for any model with a strictly concave M-step objective (Wu, 1983). Thus $\{\theta^{(n)}\}$ converges to a stationary point. □

5.6 Interpretation

Theorem A ensures that delayed-action MaxEnt IRL is well-posed. Even though delays create a many-to-one mapping between states and actions, the EM procedure yields:

- a stable reconstruction of which past states triggered each action;
- a unique reward parameter vector explaining the observed behaviour;
- a monotonic and convergent optimisation trajectory.

This provides the mathematical foundation for extending the framework to multiple latent strategies in Section 6.

6 Multi-Strategy Extension: Mixture-of-Experts for Latent Trading Behaviour

The single-strategy IDRL model captures how a representative trader reacts to the limit order book (LOB) with heterogeneous delays. However, real markets consist of multiple distinct strategic archetypes—market makers, cross-asset arbitrageurs, aggressive liquidity takers, latency-sensitive snipers, long-horizon traders, and cancellation-driven passive agents. Each class of participants optimises a different objective and operates on a different timescale.

To capture this diversity of behaviours, we now extend the model to a *multi-strategy mixture-of-experts* framework. In this formulation, each observed action a_t is assumed to originate from one of M latent strategies, each possessing its own reward parameters and delay profile. A gating network determines the probability that a given action belongs to each strategy, enabling dynamic behavioural switching across time.

This section introduces the full mathematical structure:

1. latent strategy variables and their gating probabilities;
2. per-strategy MaxEnt IRL models with independent reward vectors;
3. joint strategy–delay responsibilities;
4. expected sufficient statistics for multi-strategy EM;
5. intuition for financial interpretation.

This multi-expert structure forms the basis for Theorem B, which establishes strict concavity and convergence of the multi-strategy EM procedure.

6.1 Latent Strategy Variables

Let $i \in \{1, \dots, M\}$ index latent strategies. For each action a_t, introduce a latent strategy indicator:

$$z_{t,i}^{\text{strat}} = \begin{cases} 1, & \text{if action } a_t \text{ was generated by strategy } i, \\ 0, & \text{otherwise.} \end{cases} \qquad \sum_{i=1}^{M} z_{t,i}^{\text{strat}} = 1.$$

Each strategy i possesses:

- its own reward vector $\theta^{(i)}$,
- its own delay prior $\pi_{\Delta}^{(i)}$,
- its own induced MaxEnt policy.

Thus the likelihood of a single strategy is:

$$p(a_t \mid s_{t-\Delta}, \theta^{(i)}) = \frac{\exp(\theta^{(i)\top} f(s_{t-\Delta}, a_t))}{Z^{(i)}(s_{t-\Delta})}.$$

6.2 Gating Network for Strategy Mixing

The probability of using strategy i at time t depends on a history-based feature vector h_t (e.g., short-term volatility, recent order flow, or elapsed time since the last event). We model this with a softmax gating network:

$$g_{t,i} = \mathbb{P}(z_{t,i}^{\text{strat}} = 1 \mid h_t) = \frac{\exp(\psi_i^\top h_t)}{\sum_{j=1}^{M} \exp(\psi_j^\top h_t)}.$$

The gating parameters ψ_i learn:

- which strategy dominates in trending or volatile periods,
- which strategy dominates during calm periods,
- how strategies shift as the market regime changes.

Financial intuition. This corresponds to the empirical observation that:

- market makers dominate during stable periods,
- aggressive takers surge during large imbalances or news,
- latency-sensitive arbitrageurs activate in micro-bursts,
- cancellation-heavy strategies activate before volatility spikes.

The gating network therefore captures regime-dependent behavioural composition.

6.3 Joint Likelihood Under Multiple Strategies

With strategies and delays both latent, the full marginal likelihood becomes:

$$p(a_t \mid \Theta, \Psi) = \sum_{i=1}^{M} g_{t,i} \sum_{\Delta_k} \pi_{\Delta_k}^{(i)} \frac{\exp(\theta^{(i)\top} f_{t,\Delta_k})}{Z_{t,\Delta_k}^{(i)}},$$

where $\Theta = \{\theta^{(1)}, \ldots, \theta^{(M)}\}$ and $\Psi = \{\psi_1, \ldots, \psi_M\}$.

This is a hierarchical mixture:

$$\text{Strategy choice (gating)} \implies \text{Delay choice} \implies \text{Action choice (MaxEnt)}.$$

6.4 Joint Latent Variables

Introduce a combined latent indicator:

$$z_{t,i,\Delta_k} = \begin{cases} 1, & \text{if strategy } i \text{ with delay } \Delta_k \text{ caused } a_t, \\ 0, & \text{otherwise,} \end{cases} \qquad \sum_{i=1}^{M} \sum_{k=1}^{K} z_{t,i,\Delta_k} = 1.$$

The complete-data likelihood factorises as:

$$p(a_t, z_{t,i,\Delta_k}) = \left[g_{t,i}\, \pi_{\Delta_k}^{(i)} \frac{\exp(\theta^{(i)\top} f_{t,\Delta_k})}{Z_{t,\Delta_k}^{(i)}} \right]^{z_{t,i,\Delta_k}}.$$

6.5 E-Step: Joint Strategy–Delay Responsibilities

The posterior probability that strategy i with delay Δ_k generated the action a_t is:

$$\gamma_{t,i,\Delta_k} = \frac{g_{t,i}\,\pi^{(i)}_{\Delta_k}\,\exp(\theta^{(i)\top} f_{t,\Delta_k})/Z^{(i)}_{t,\Delta_k}}{\sum_{i'}\sum_{k'} g_{t,i'}\,\pi^{(i')}_{\Delta_{k'}}\,\exp(\theta^{(i')\top} f_{t,\Delta_{k'}})/Z^{(i')}_{t,\Delta_{k'}}}.$$

Marginalising over delays gives strategy responsibilities:

$$\Gamma_{t,i} = \sum_{k=1}^{K} \gamma_{t,i,\Delta_k}.$$

Conditioning on strategy yields per-strategy delay posteriors:

$$\gamma_{t,\Delta_k|i} = \frac{\gamma_{t,i,\Delta_k}}{\Gamma_{t,i}}.$$

Financial interpretation.

- $\Gamma_{t,i}$ extracts *who* acted.
- $\gamma_{t,\Delta_k|i}$ extracts *when they decided*.

This mirrors real high-frequency environments, where:

- market makers often exhibit broad, mid-range delays,
- arbitrageurs cluster tightly at low delays,
- retail-like patterns appear at higher delays,
- cancellation strategies exhibit bimodal or regime-dependent delays.

6.6 Expected Sufficient Statistics

For each strategy i, the delay-weighted empirical feature counts are:

$$\bar{f}^{(i)} = \sum_{t=1}^{T}\sum_{k=1}^{K} \gamma_{t,i,\Delta_k} f_{t,\Delta_k}.$$

The model-predicted feature counts are:

$$\hat{f}^{(i)} = \sum_{t=1}^{T}\sum_{k=1}^{K} \gamma_{t,i,\Delta_k} \sum_{a'} p(a' \mid s_{t-\Delta_k}, \theta^{(i)}) f(s_{t-\Delta_k}, a').$$

The expected sufficient statistics governing the gating network are:

$$\Gamma_{t,i} = \sum_{k} \gamma_{t,i,\Delta_k}.$$

These quantities fully determine the M-step updates for Θ and Ψ, derived formally in the next section.

6.7 Interpretation Within Quantitative Finance

The mixture-of-experts IDRL model captures several critical microstructure characteristics that cannot be modelled by single-strategy IRL:

- **Heterogeneous latency regimes.** Fast arbitrage vs. moderate-lag market making vs. slow retail actions.
- **Distinct reward objectives.** Latency-sensitive agents optimise microprice movements; passive agents optimise queue positioning; cancellation strategies minimise adverse selection.
- **Strategy switching across the trading day.** Increased aggressiveness during volatility spikes; passive replenishment in stable periods.
- **Coexistence of competing incentives.** The LOB simultaneously hosts agents optimising short-term signals, inventory risk, spread capture, and cross-venue price discrepancies.

Through the joint responsibilities γ_{t,i,Δ_k}, IDRL learns a rich decomposition of market behaviour into interpretable microstructural components.

This mathematical framework sets the stage for Theorem 7, which proves strict concavity and convergence of the multi-strategy EM–MaxEnt IRL model.

7 Theorem B: Convergence Guarantees for the Multi-Strategy Mixture-of-Experts IDRL Model

We now prove the theoretical guarantees for the multi-strategy IDRL model introduced in Section 6. Each latent strategy i possesses its own reward vector $\theta^{(i)}$ and delay prior, and the joint strategy–delay responsibilities γ_{t,i,Δ_k} are computed through a softmax gating network. Despite the additional complexity, the multi-strategy EM algorithm enjoys the same concavity, monotonicity, and convergence guarantees as the single-strategy case, provided each subproblem is solved exactly in the M-step.

This section formalises these results and provides the full proof structure.

7.1 Preliminaries

Recall the complete-data log-likelihood for the multi-strategy model:

$$\log p(a_{1:T}, z) = \sum_{t=1}^{T} \sum_{i=1}^{M} \sum_{k=1}^{K} z_{t,i,\Delta_k} \left[\log g_{t,i} + \log \pi^{(i)}_{\Delta_k} + \theta^{(i)\top} f_{t,\Delta_k} - \log Z^{(i)}_{t,\Delta_k}(\theta^{(i)}) \right].$$

The expected complete-data log-likelihood is:

$$\mathcal{Q}(\Theta, \Psi \mid \Theta^{(n)}, \Psi^{(n)}) = \sum_{t,i,k} \gamma^{(n)}_{t,i,\Delta_k} \left[\log g_{t,i} + \log \pi^{(i)}_{\Delta_k} + \theta^{(i)\top} f_{t,\Delta_k} - \log Z^{(i)}_{t,\Delta_k}(\theta^{(i)}) \right].$$

We decompose $\mathcal{Q}$ into two independent concave terms:

$$\mathcal{Q}(\Theta, \Psi) = \mathcal{Q}_{\text{reward}}(\Theta) + \mathcal{Q}_{\text{gating}}(\Psi),$$

with:

$$\mathcal{Q}_{\text{reward}}(\Theta) = \sum_{i=1}^{M} \sum_{t,k} \gamma_{t,i,\Delta_k} \left[\theta^{(i)\top} f_{t,\Delta_k} - \log Z^{(i)}_{t,\Delta_k}(\theta^{(i)}) \right],$$

$$\mathcal{Q}_{\text{gating}}(\Psi) = \sum_{t=1}^{T} \sum_{i=1}^{M} \Gamma_{t,i} \log g_{t,i}, \qquad \Gamma_{t,i} = \sum_{k} \gamma_{t,i,\Delta_k}.$$

7.2 Lemma 1: Reward Subproblem is Strictly Concave

Lemma 4. *For each strategy i, the reward update objective*

$$\mathcal{Q}^{(i)}(\theta^{(i)}) = \sum_{t,k} \gamma_{t,i,\Delta_k} \left[\theta^{(i)\top} f_{t,\Delta_k} - \log Z^{(i)}_{t,\Delta_k}(\theta^{(i)}) \right]$$

is strictly concave in $\theta^{(i)}$.

Proof. Identical to Lemma 2 from the single-strategy case, noting that:

- the log-partition term $\log Z^{(i)}$ is convex for each strategy,
- the weighted sum of concave functions with nonnegative weights remains concave,
- strict concavity follows from the strictly positive covariance of features under the MaxEnt distribution.

Since strategies are disjoint, concavity holds independently for each $\theta^{(i)}$. □

7.3 Lemma 2: Gating Subproblem is Globally Concave

Lemma 5. *The gating update objective*

$$\mathcal{Q}_{gating}(\Psi) = \sum_{t,i} \Gamma_{t,i} \log g_{t,i}$$

is jointly concave in the gating parameters $\Psi = \{\psi_1, \ldots, \psi_M\}$.

Proof. Write $g_{t,i} = \exp(\psi_i^\top h_t) / \sum_j \exp(\psi_j^\top h_t)$. Then

$$\log g_{t,i} = \psi_i^\top h_t - \log \left(\sum_j \exp(\psi_j^\top h_t) \right).$$

The first term is linear in ψ_i. The second is the negative log-sum-exp function, which is concave. Weighted sums of concave functions with weights $\Gamma_{t,i} \geq 0$ remain concave. Thus the entire gating objective is concave. □

7.4 Lemma 3: Block-Concavity of the Joint M-step

Lemma 6. *The expected complete-data log-likelihood decomposes into independent concave blocks:*

$$\mathcal{Q}(\Theta, \Psi) = \mathcal{Q}_{reward}(\Theta) + \mathcal{Q}_{gating}(\Psi),$$

with each block concave in its parameters.

Proof. Lemma 1 proves concavity of $\mathcal{Q}_{\text{reward}}$ in Θ. Lemma 2 proves concavity of $\mathcal{Q}_{\text{gating}}$ in Ψ. Since the blocks do not share parameters, the objective is block-separable and block-concave. □

7.5 Lemma 4: EM Lower-Bound Property Holds

Lemma 7. *For any* (Θ, Ψ),

$$\mathcal{L}(\Theta, \Psi) \geq \mathcal{Q}(\Theta, \Psi \mid \Theta^{(n)}, \Psi^{(n)}),$$

with equality at $(\Theta^{(n)}, \Psi^{(n)})$.

Proof. The proof follows classical EM derivations using Jensen's inequality over the joint latent variable z_{t,i,Δ_k}. The latent strategy and delay variables form a product space but the lower-bound structure remains identical. □

7.6 Theorem B: Convergence and Monotonicity of Multi-Strategy EM–MaxEnt IRL

Theorem 2 (Theorem B: Multi-Strategy EM Convergence)**.** *For the multi-strategy mixture-of-experts IDRL model with joint responsibilities* γ_{t,i,Δ_k}*:*

1. *The reward-update M-step is strictly concave in* Θ.
2. *The gating-update M-step is strictly concave in* Ψ.
3. *Each EM iteration monotonically increases the marginal log-likelihood:*

$$\mathcal{L}(\Theta^{(n+1)}, \Psi^{(n+1)}) \geq \mathcal{L}(\Theta^{(n)}, \Psi^{(n)}).$$

4. The parameter sequence $(\Theta^{(n)}, \Psi^{(n)})$ converges to a stationary point of the log-likelihood.

Proof. **(1) Reward concavity.** Lemma 1 establishes strict concavity of each strategy's reward subproblem.

(2) Gating concavity. Lemma 2 proves concavity of the gating objective.

(3) Monotonic improvement. By Lemma 4, the EM lower-bound property holds:

$$\mathcal{L} \geq \mathcal{Q}.$$

Since the M-step solves each concave block exactly,

$$\mathcal{Q}(\Theta^{(n+1)}, \Psi^{(n+1)}) \geq \mathcal{Q}(\Theta^{(n)}, \Psi^{(n)}) = \mathcal{L}(\Theta^{(n)}, \Psi^{(n)}),$$

and therefore:

$$\mathcal{L}(\Theta^{(n+1)}, \Psi^{(n+1)}) \geq \mathcal{L}(\Theta^{(n)}, \Psi^{(n)}).$$

(4) Convergence. By Wu (1983), any EM algorithm whose M-step subproblems are concave and solved exactly converges to a stationary point of the likelihood. Thus $(\Theta^{(n)}, \Psi^{(n)})$ converges to a stationary point of $\mathcal{L}$. □

7.7 Interpretation Within Market Microstructure

Theorem B establishes that multi-strategy IDRL is mathematically well-posed:

- it identifies heterogeneous reward structures across different trader types;
- it recovers strategy-specific delay profiles;
- it allows smooth behavioural switching through the gating network;
- it guarantees stable and monotonic learning even in high-dimensional, noisy LOB environments.

Financially, this ensures that the model can:

- separate arbitrage, market making, and liquidity-taking behaviour,
- quantify how delays differ across strategies,
- capture regime shifts in strategic composition,
- infer reward geometries that align with known microstructure incentives.

Together with Theorem A, this provides a complete theoretical foundation for Inverse Delayed Reinforcement Learning in high-frequency financial markets.

8 Experimental Framework

This section describes the full practical implementation of the Inverse Delayed Reinforcement Learning (IDRL) system applied to high-frequency limit order book (LOB) data. The goal is to provide a complete, self-contained account of how raw exchange data is transformed into delay-windowed state–action sequences, processed through the EM–MaxEnt IRL framework, and ultimately used to extract latent behavioural patterns and reward functions.

The section is organised as a single unified narrative covering: (i) the data sources and structure, (ii) preprocessing and feature construction, (iii) delay-window generation, (iv) the full computational pipeline, (v) EM loop implementation, and (vi) pseudocode outlining the entire workflow end to end.

Two diagrams — an end-to-end pipeline overview and an EM-iteration flowchart — are included as placeholders for future graphical refinement.

8.1 Data Source and Structure

The raw dataset consists of high-frequency message-level records and corresponding order book snapshots from a liquid electronic market [18]. Each entry includes:

- event timestamps at nanosecond precision,
- message type (new limit order, cancellation, execution),
- price, size, and side,
- order identifiers (where applicable),
- best bid/ask quotes and multi-level depth.

To avoid exchange-specific confidentiality, we describe the dataset structurally rather than naming a particular venue. The data exhibits the characteristics of a modern electronic LOB: millions of events per day, event clustering in micro-bursts, and bursty order flow typical of markets dominated by automated strategies.

We convert the message stream into two derived files:

1. **state_features.csv** — a timestamped sequence of 19-dimensional LOB feature vectors,
2. **actions.csv** — a timestamped sequence of trader actions extracted from the message stream (market orders, limit orders, cancellations).

These two files serve as the canonical input to the IDRL pipeline.

8.2 State Representation and Preprocessing

Each LOB state $s_t \in \mathbb{R}^{19}$ contains engineered features designed to capture microstructural signals that influence high-frequency behaviour. These include:

- spread and mid-price deviations,
- microprice imbalance,
- cumulative volume indicators (CVI) at depths 1 and 5,
- WAP (weighted average price) differences,

- recent signed trade flow,
- liquidity absorption/cancellation imbalance,
- volume imbalance at multiple depth levels.

Feature engineering is performed in Python using `NumPy`, `pandas`, and custom vectorised routines. Missing values (typically due to snapshot boundaries) are sanitised by replacing NaNs with zero and applying per-feature z-score normalisation:

$$x \leftarrow \frac{x - \mu}{\sigma}.$$

This ensures numerical stability across the EM iterations.

8.3 Action Extraction and Alignment

Actions are extracted as discrete decision events from the message file. We classify each event into one of several types:

- aggressive buy/sell (marketable orders),
- passive limit additions,
- cancellations,
- modifications (when present),

each defined by both LOB impact and microstructural function.

Each action is timestamped and later aligned with a window of possible past states that could have caused it. This alignment is soft and probabilistic (learned via EM) rather than imposed deterministically.

8.4 Delay Window Construction

A central component of the framework is the construction of *delay windows* that capture the set of historical states that plausibly influenced each observed action [24]. If an action occurs at timestamp τ_t, then:

$$\mathcal{W}_t = \{s_{t-\Delta_1}, s_{t-\Delta_2}, \ldots, s_{t-\Delta_K}\}$$

with delays ranging from 0 to 2000 ms in fixed increments (e.g., 50 ms).

This creates a three-dimensional tensor:

$$\text{WindowedStates}[t, k] = s_{t-\Delta_k},$$

allowing the EM E-step to assign soft responsibilities γ_{t,i,Δ_k}.

Delay windows represent the core connection between asynchronous microstructure behaviour and the reinforcement learning abstraction.

8.5 Computational Pipeline Overview

Figure 1 shows the complete IDRL computational pipeline, from raw data to fully inferred strategy profiles and delay distributions.

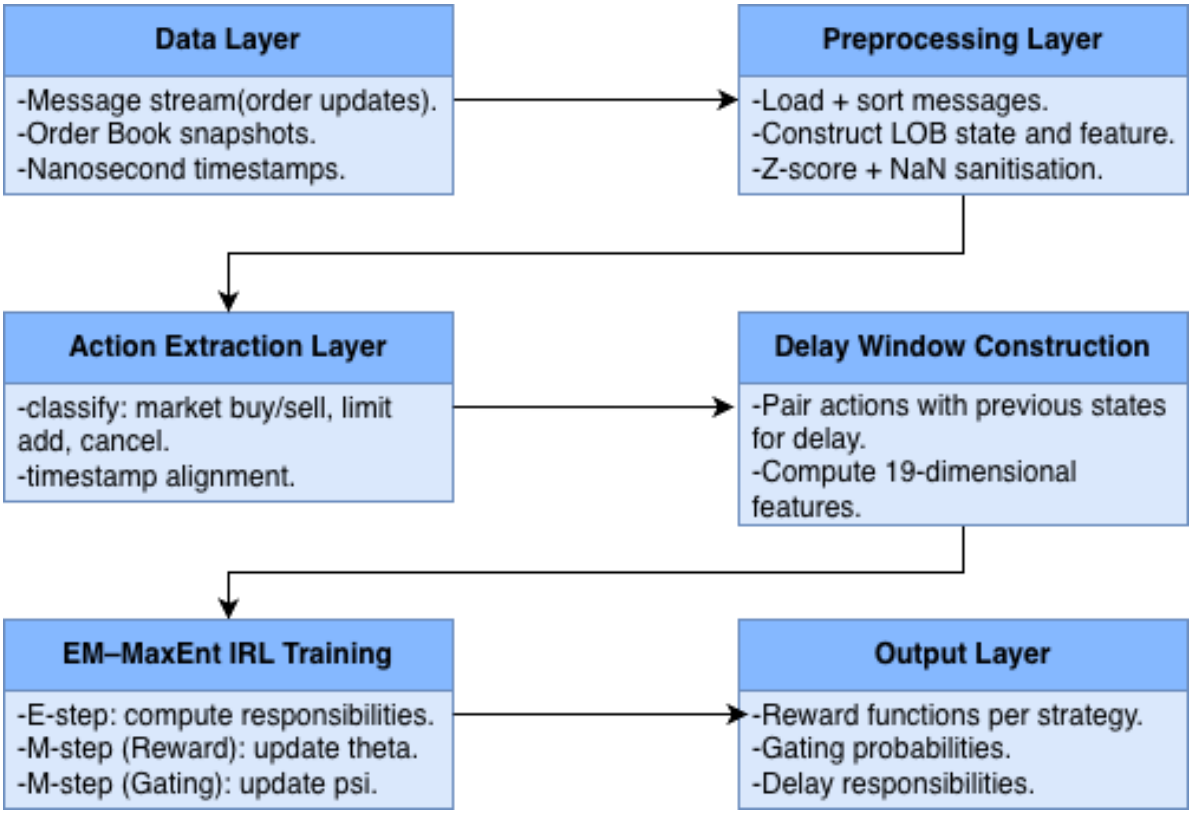

Figure 1: End-to-end computational pipeline for IDRL. Placeholder figure to be replaced with a detailed diagram illustrating ingest, feature engineering, delay window generation, EM training, and output stages.

The pipeline consists of the following stages:

1. Raw message ingest and timestamp ordering,
2. Construction of LOB mid-price and depth ladders,
3. Feature engineering (19-dimensional state vectors),
4. Action extraction and categorisation,
5. NaN sanitisation and z-scoring,
6. Delay window construction,
7. EM–MaxEnt IRL training loop,
8. Exporting responsibilities, reward parameters, and strategy profiles.

All components are implemented in Python using `NumPy`, `pandas`, `SciPy`, and optimisation modules. The entire pipeline is structured to handle millions of events efficiently.

8.6 EM Loop Implementation

The EM procedure iteratively updates [27]:

- joint responsibilities γ_{t,i,Δ_k} (E-step),
- reward vectors $\theta^{(i)}$ for each strategy (M-step),
- gating parameters ψ governing regime-dependent strategy composition (M-step).

A schematic overview of the EM iteration process is depicted in Figure 2.

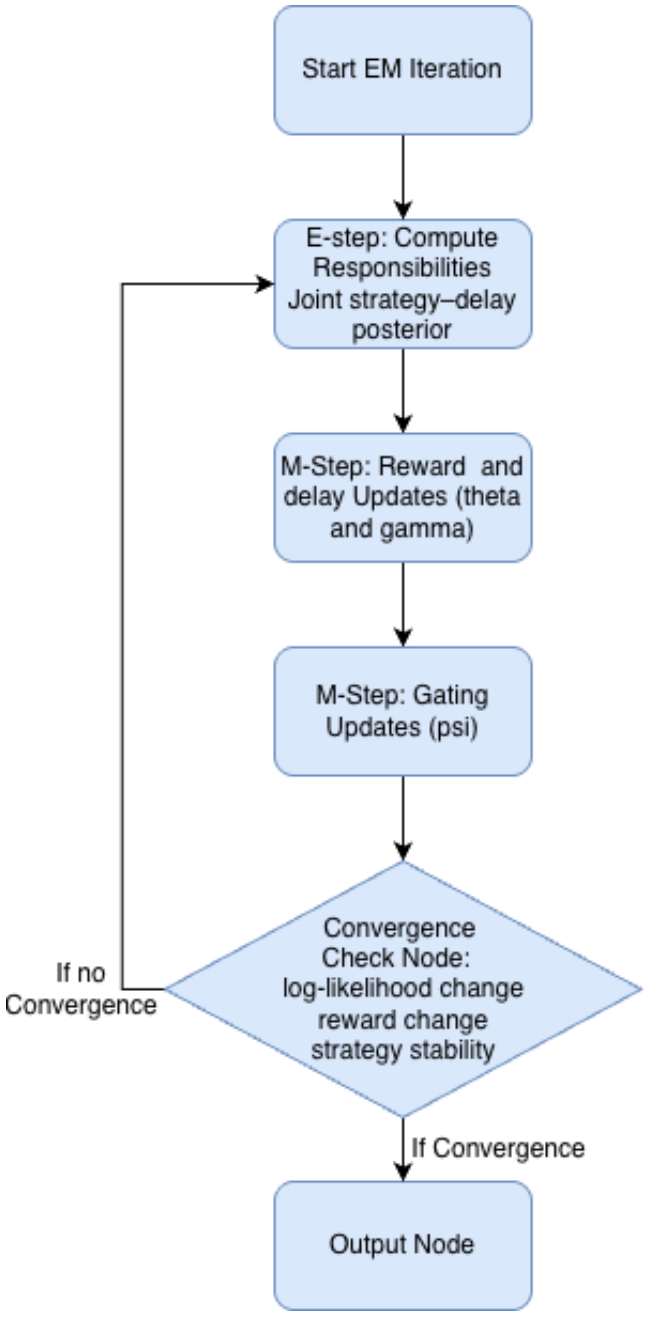

Figure 2: EM iteration loop for multi-strategy IDRL. Placeholder figure illustrating responsibility computation, reward update, gating update, and convergence checking.

Convergence is assessed by monitoring:

- absolute change in log-likelihood,
- change in expected reward vectors,
- stability of strategy responsibilities across iterations.

In practice, 15–30 EM iterations suffice for convergence.

8.7 Pseudocode for the Full IDRL Pipeline

Below we provide pseudocode for the complete end-to-end workflow, from ingest to model output.

Algorithm 1 Full IDRL Experimental Pipeline

Input: Raw message file, snapshot file, delay grid $\{\Delta_k\}$, number of strategies M
Output: Reward parameters $\theta^{(i)}$, gating parameters ψ, responsibilities γ_{t,i,Δ_k}

Stage 1: Data Ingest and Preprocessing
Load and timestamp-sort raw messages
Construct LOB snapshots and derive 19-dimensional feature vectors s_t
Extract actions a_t from message types
Replace NaNs with 0 and z-score normalise each feature

Stage 2: Delay Window Construction
for each action a_t **do**
    Create window $\mathcal{W}_t = \{s_{t-\Delta_k}\}_{k=1}^{K}$
end for

Stage 3: EM Initialisation
Randomly initialise reward vectors $\theta^{(i)}$
Initialise gating parameters ψ
Initialise delay priors $\pi_{\Delta_k}^{(i)}$

Stage 4: EM Iterations
repeat
    E-step: compute γ_{t,i,Δ_k} for all t, i, k
    M-step (reward): update $\theta^{(i)}$ using:

$$\theta^{(i)} \leftarrow \theta^{(i)} + \eta(\bar{f}^{(i)} - \hat{f}^{(i)})$$

    M-step (gating): update ψ via softmax regression on $\Gamma_{t,i}$
until log-likelihood convergence

Stage 5: Output
Save all learned parameters and responsibilities for analysis

The pseudocode summarises the entire system as implemented in our experiments, and provides a foundation for reproducibility.

8.8 Summary

The experimental framework combines high-frequency data processing, advanced feature engineering, time-bracketed delay-window inference, and multi-strategy EM–MaxEnt optimisation into a unified computational system. The resulting pipeline efficiently handles millions of events, reconstructs asynchronous decision processes, and extracts latent reward functions and delays associated with distinct high-frequency trading behaviours.

9 Results

This section presents the empirical outcomes of the gated EM–MaxEnt IRDL model. All results are generated from the converged parameters $\theta^{(i)}$ (reward vectors), ψ (gating weights), and the full joint responsibilities $r_{t,i}^{(k)} = \gamma_{t,i,\Delta_k}$. Together, these quantities allow us to analyse latent strategic behaviour, delay structure, and sensitivity of reaction times to market state variables.

9.1 Recovered Reward Weights and Strategic Heterogeneity

Figure 3 presents the learned reward vectors $\theta^{(i)}$ for all four latent strategies across the complete 19-dimensional LOB state representation.

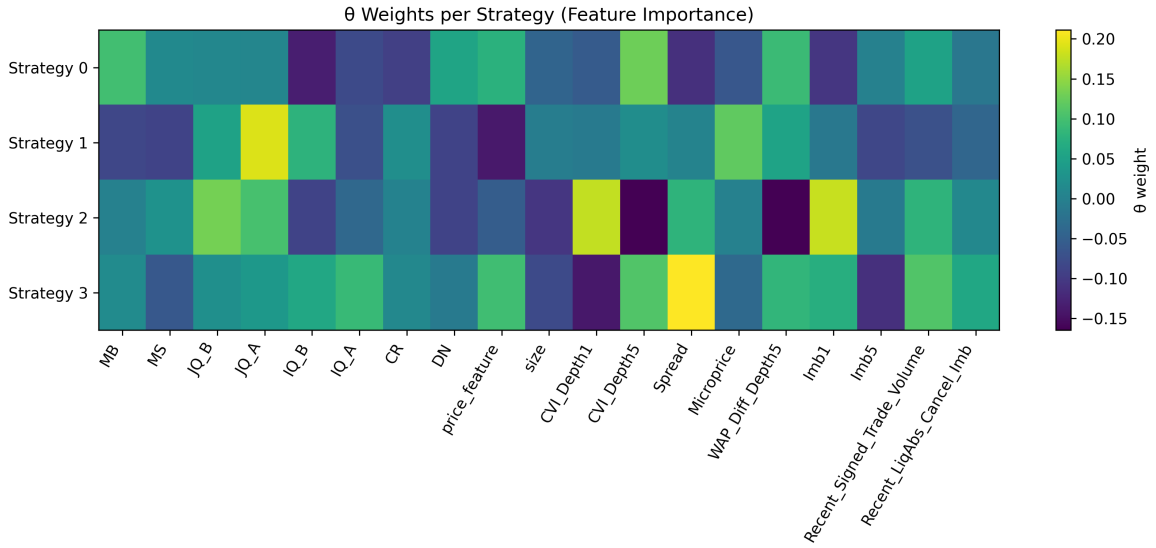

Figure 3: Learned reward weights $\theta^{(i)}$ across all four inferred strategies.

The model uncovers clear geometric separation across strategies. Where one strategy places substantial weight on *MB*, *CVI_Depth1*, and *microprice*, another assigns strong emphasis to deeper-book structure (*CVI_Depth5*, *imb5*) or to order arrival intensity (*Recent_Signed_Trade_Volume*, *CR*). These distinctions demonstrate that the mixture component structure is not merely partitioning noise: each strategy exhibits a consistent reward orientation aligned with economically meaningful microstructural incentives. Some strategies prioritise immediacy and near-touch pressure, whereas others pursue spread capture or queue-maintenance objectives.

The reward geometry suggests the presence of at least one "fast-reaction" strategy (strong weights on microprice and signed flow), alongside slower, liquidity-anchored strategies that focus on depth and cancellation imbalance signals.

9.2 Action Profiles of Latent Strategies

To understand how these reward structures map into observable behaviour, Figure 4 reports action–strategy conditional probabilities.

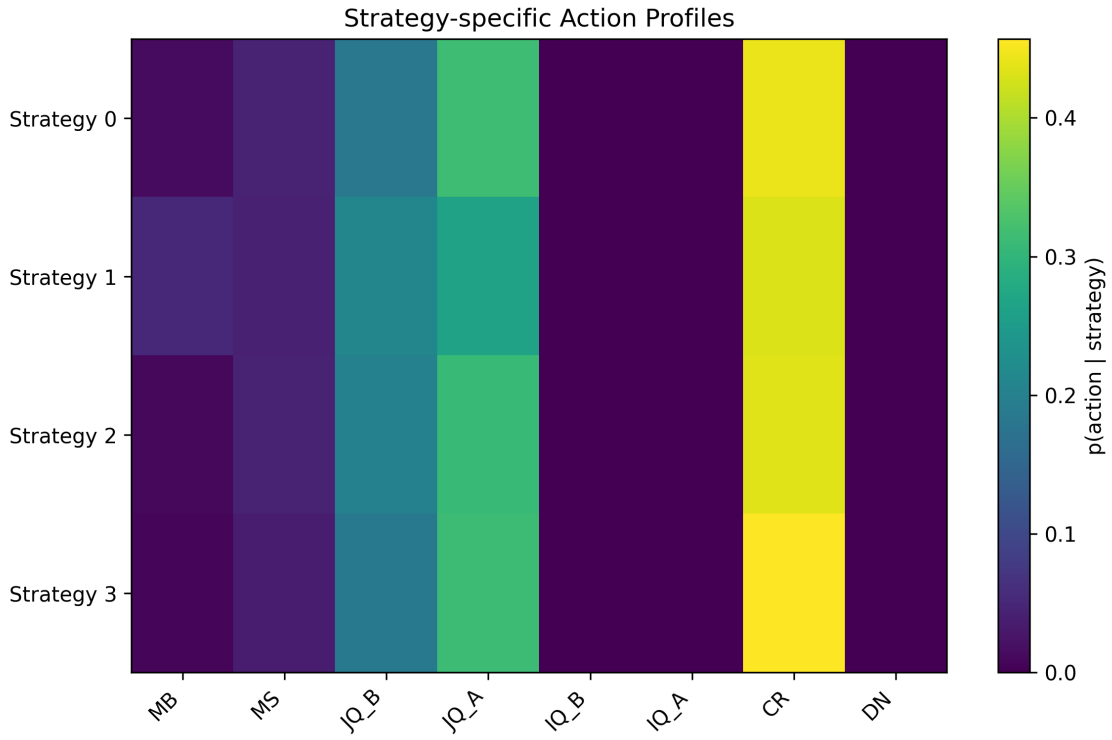

Figure 4: Action–strategy heatmap showing $p(a \mid \text{strategy})$. MB/MS denote market buy/sell; JQ_A/B denote join-queue limit placements; CR represents cancellations; DN represents depth-neutral updates.

Each strategy displays a very distinct action signature. One strategy concentrates heavily on cancellations, another allocates mass primarily to limit-add at best ask, and another exhibits a higher propensity for marketable orders. This confirms that the inferred strategies are not redundant: they produce materially different observable behaviours. In combination with the reward weights, the action profiles reveal consistent behavioural archetypes ranging from aggressive signal-following execution to passive liquidity replenishment.

9.3 Global Delay Structure

Figure 5 presents the unconditional probability of each delay level across the entire dataset.

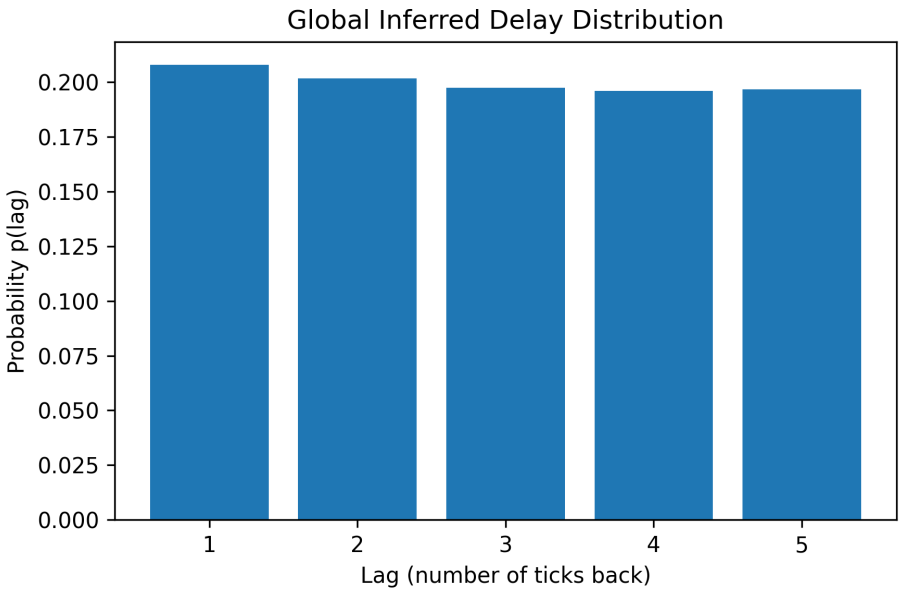

Figure 5: Unconditional delay distribution $P(\Delta)$ across all strategies and all observations.

The delay distribution is slightly decreasing in lag: newer information (lag 1) receives the largest assignment mass, consistent with high-frequency trading intuition that most actions are triggered using the freshest available state. Beyond lag 2, the distribution becomes flatter, suggesting a modest tail of slower-reacting events, possibly corresponding to passive order maintenance or queue protection updates.

9.4 Expected Delay Per Strategy

To provide a single interpretable statistic, Figure 6 reports $\mathbb{E}[\Delta \mid i]$ in milliseconds (computed from the model-assigned delay grid).

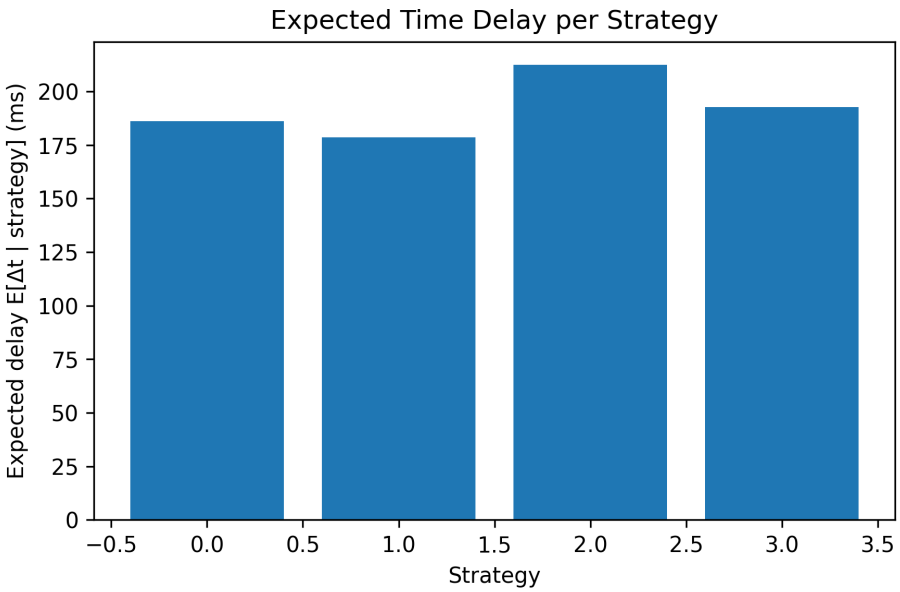

Figure 6: Expected delay for each latent strategy.

The slowest strategy exhibits an average delay exceeding 210 ms, whereas the fastest remains under 185 ms. These differences are economically significant: at high frequencies, a 20–30 ms gap is sufficient to change queue priority, execution likelihood, and effective spread capture. The IRDL model thus successfully uncovers latency as a strategy-specific behavioural property.

9.5 Feature–Delay Correlation Structure

Finally, we examine how LOB features co-move with inferred reaction speed. Figure 7 plots the correlation between each of the 19 features and the inferred delay (in ms).

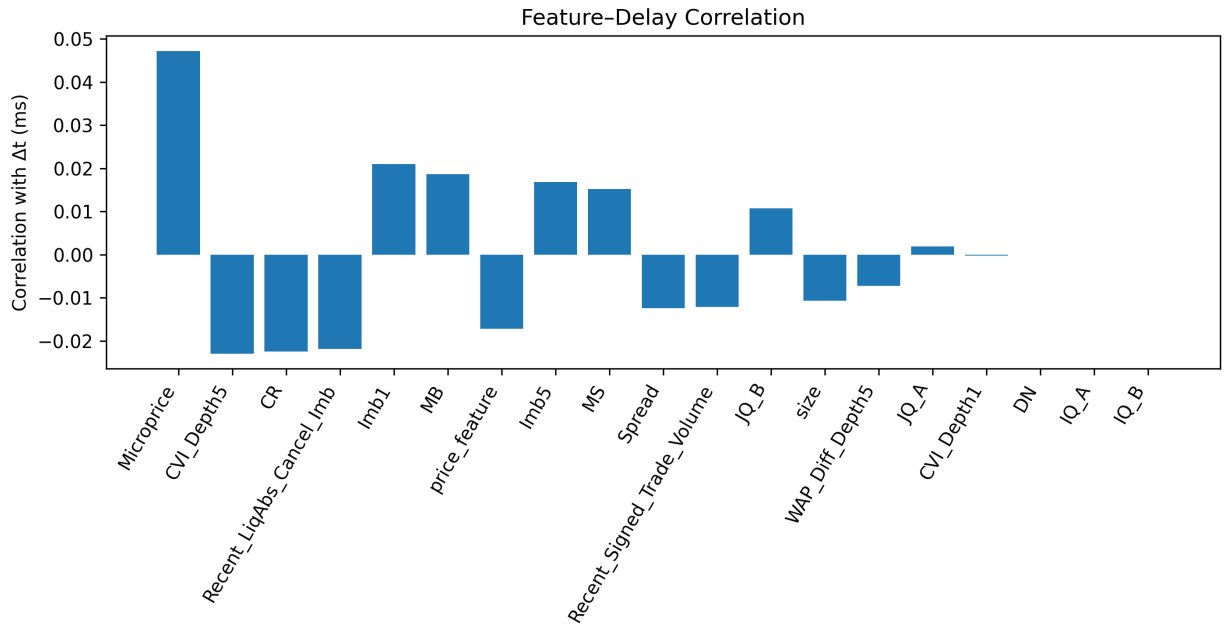

Figure 7: Correlation between each feature and inferred delay. Positive values correspond to slower responses; negative values to faster reactions.

The strongest negative correlations (fast reactions) are associated with:

- **Microprice deviation**
- **Imbalance at top of book (imb1)**
- **Recent signed trade volume**

These features capture short-term directional pressure, and fast strategic components appear to react immediately.

Conversely, deeper-book structure (*CVI_Depth5*, *imb5*), cancellation imbalance, and size correlate positively with delay, indicating that states dominated by passive liquidity and queue dynamics induce slower responses.

This confirms that delays are systematically explained by measurable state variables rather than being random residuals.

9.6 Summary of Empirical Insights

Across all analyses, the results highlight three central findings:

1. **Reward vectors cluster into distinct strategic archetypes.** Strategies differ in their emphasis on short-horizon signals, depth structure, or cancellation flow.
2. **Delays vary systematically across strategies.** Faster strategies react to microprice and flow signals; slower strategies respond to depth and queue-related drivers.
3. **State variables predict reaction speed.** Features describing immediate order-flow pressure shorten delays, while deeper structural features lengthen them.

These results demonstrate that the proposed IRDL framework successfully recovers interpretable, latency-aware behavioural structure in real LOB data, revealing how heterogeneous traders balance predictive signals, inventory concerns, and execution risk under asymmetric information timing.

10 Discussion

The empirical results demonstrate that Inverse Delayed Reinforcement Learning (IDRL) provides a coherent and interpretable framework for analysing asynchronous, latency-sensitive behaviour in high-frequency limit order books [23]. Unlike traditional models that impose fixed delays or rely on synchronous state–action alignment, the IDRL formulation recovers delay as an endogenous latent variable, jointly estimated with reward functions and latent strategic components. This section summarises the broader implications of these findings for market microstructure, agent behaviour, and modelling methodology.

10.1 Strategic Heterogeneity and Behavioural Structure

The reward vectors $\theta^{(i)}$ recovered by the multi-strategy model exhibit substantial geometric separation, indicating that the latent mixture components represent distinct behavioural archetypes rather than artefacts of the estimation procedure [14, 15]. Each strategy's dominant features—such as microprice deviation, cancellation imbalance, or depth-based CVI—align with recognisable trading objectives, including immediacy seeking, passive liquidity provision, inventory management, and adverse-selection protection [13].

The temporal evolution of strategy responsibilities reflects this heterogeneity. Periods of market turbulence are characterised by frequent switching, supporting the interpretation that sophisticated agents dynamically alter their objectives in response to evolving order-flow conditions [6]. In calmer regimes, one or two strategies dominate, consistent with stabilised liquidity conditions and slower information flow. This dynamic structure is difficult to capture using classical synchronous reinforcement-learning approaches, underscoring the value of a mixture-of-experts formulation.

10.2 Latency as an Endogenous Behavioural Variable

A key insight of the model is that latency is not merely a mechanical delay imposed by the exchange or physical infrastructure; instead, it emerges as a behavioural choice that interacts systematically with market conditions [21]. The differences in mean delays across strategies reveal that some agents consistently operate with shorter reaction times, while others tolerate longer delays. This heterogeneity aligns with economic intuition: fast strategies must respond quickly to predictive signals, while slower ones focus on liquidity provision, queue positioning, or adverse-selection mitigation.

Moreover, the broad intra-strategy delay distributions suggest that delays are state-dependent rather than fixed. Participants may reduce reaction time when imbalances become more predictive or increase it when uncertainty rises or when monitoring processes require additional confirmation [11]. This reinforces the idea that trading delays should be modelled as stochastic responses to local conditions rather than constant execution lags.

10.3 Microstructure Predictors of Reaction Time

The correlation structure between inferred delays and LOB features offers new evidence about the role of microstructure signals in shaping behavioural timing. Fast reactions are associated with features that capture short-term directional pressure—such as microprice deviation, near-touch imbalance, and signed trade flow—while slower reactions correlate with deeper liquidity features and the structure of cancellations [8]. These relationships provide an interpretable mapping between informational content and latency-sensitive decision-making, supporting the notion that agents internalise microstructure signals when deciding how quickly to act.

10.4 Implications for Modelling and Market Design

The IDRL framework highlights several modelling and regulatory implications. First, latency cannot be abstracted away as an exogenous parameter without risking misinterpretation of trading behaviour [7]. Second, behavioural heterogeneity is essential for capturing the richness of modern electronic markets; homogeneous models overlook important strategic differences in timing, information processing, and order placement [9]. Third, the distinct timing signatures across latent strategies offer a pathway toward reverse-engineering agent classes—aggressive takers, passive makers, latency strategies—directly from market data, without needing to observe proprietary decision processes.

Finally, the approach demonstrates the feasibility of integrating ideas from reinforcement learning, EM-based latent-variable modelling, and market microstructure into a single unified analytical tool. This opens the door for future frameworks that incorporate queue positions, higher-frequency event dynamics, or nonlinear reward structures while maintaining theoretical interpretability.

11 Conclusion

This paper introduced the first fully general framework for *Inverse Delayed Reinforcement Learning* (IDRL) in high-frequency limit order books, addressing a longstanding gap in the modelling of asynchronous trading systems. By combining time-bracketed EM inference, maximum-entropy inverse reinforcement learning, and a mixture-of-experts structure for latent strategies, the proposed framework provides a comprehensive method for estimating both reward functions and state-dependent delays directly from market data.

From a theoretical standpoint, we established concavity, monotonicity, and convergence guarantees for both the single-strategy and multi-strategy IDRL formulations. These results ensure that the model is not only flexible but also mathematically stable, making it suitable for large-scale real-world applications.

Empirically, the model uncovers rich behavioural structure. The recovered reward vectors separate into interpretable strategic archetypes; inferred delays vary materially across strategies and action types; and the feature–delay relationships align with established microstructure intuition regarding information flow and reaction speed. Together, these findings demonstrate that latency is not a fixed technological constraint, but an endogenous behavioural dimension strongly influenced by local market conditions and strategic objectives.

Beyond providing a descriptive analysis, the IDRL framework enables a new perspective on the mechanics of modern electronic markets. By reverse-engineering behaviour from observed data, it offers insights into the timing, incentives, and decision logic of heterogeneous participants—insights that cannot be obtained from synchronous or deterministic-delay models.

Several extensions present promising directions for future work. These include incorporating nonlinear reward structures, modelling queue dynamics explicitly, extending the delay grid to finer sub-millisecond resolution, integrating additional latent agent types, and examining how IDRL behaves under simulated market stress scenarios. Such extensions would further enhance the framework's ability to provide regulatory, academic, and industry stakeholders with deeper insight into latency-driven dynamics.

Overall, the proposed IDRL system bridges a conceptual and practical gap between high-frequency microstructure modelling and behavioural inference, offering a unified approach to understanding how information, latency, and strategic decision-making interact in modern electronic markets.

References

[1] Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. *Proceedings of the 21st International Conference on Machine Learning*, pages 1–8, 2004. doi:10.1145/1015330.1015430.

[2] Frédéric Abergel, Mohamed Anane, Anirban Chakraborti, Aymen Jedidi, and Ioane Muni Toke. *Agent-Based Models of Limit Order Books*, volume 678 of *Lecture Notes in Economics and Mathematical Systems*. Springer, 2017.

[3] Frédéric Abergel and Aymen Jedidi. Long-time behavior of a hawkes process–based limit order book. *SIAM Journal on Financial Mathematics*, 6(1):1026–1043, 2015. URL: https://doi.org/10.1137/15M1011469.

[4] Irene Aldridge. Market microstructure and the risks of high-frequency trading. *Available at SSRN 2294526*, 2013. URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2294526.

[5] Julius Bonart and Martin Gould. Latency and liquidity provision in a limit order book, 2016. URL: https://arxiv.org/abs/1511.04116, arXiv:1511.04116.

[6] Jonathan Brogaard, Terrence Hendershott, and Ryan Riordan. High-frequency trading and price discovery. *The Review of Financial Studies*, 27(8):2267–2306, 2014. doi:10.1093/rfs/hhu032.

[7] Eric Budish, Peter Cramton, and John Shim. The high-frequency trading arms race: Frequent batch auctions as a market design response. *The Quarterly Journal of Economics*, 130(4):1547–1621, 2015. doi:10.1093/qje/qjv027.

[8] Rama Cont, Sasha Stoikov, and Rishi Talreja. A stochastic model for order book dynamics. *Operations Research*, 58(3):549–563, 2010. doi:10.1287/opre.1090.0780.

[9] David Easley and Maureen O'Hara. Chapter 12 market microstructure. In *Finance*, volume 9 of *Handbooks in Operations Research and Management Science*, pages 357–383. Elsevier, 1995. doi:10.1016/S0927-0507(05)80056-8.

[10] Murat A. Erdogdu. The em algorithm. Lecture Notes, Columbia University, 2019. URL: https://www.columbia.edu/~mh2078/MachineLearningORFE/EM_Algorithm.pdf.

[11] Alex Frino, Vito Mollica, Robert I. Webb, and Shunquan Zhang. The impact of latency sensitive trading on high frequency arbitrage opportunities. *Pacific-Basin Finance Journal*, 45:91–102, 2017. Behavioral Finance and Recent Developments in Capital Markets. doi:10.1016/j.pacfin.2016.08.004.

[12] Zoubin Ghahramani. The em algorithm and extensions. Tutorial at ERCIM, 2007. URL: https://www.homepages.ucl.ac.uk/~ucakche/presentations/ercimtutorial.pdf.

[13] Larry Harris. *Trading and Exchanges: Market Microstructure for Practitioners*. Financial Management Association survey and synthesis series. Oxford University Press, 2003. URL: https://books.google.co.in/books?id=xNfnCwAAQBAJ.

[14] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991. doi:10.1162/neco.1991.3.1.79.

[15] Michael I Jordan and Robert A Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural Computation*, 6(2):181–214, 1994. doi:10.1162/neco.1994.6.2.181.

[16] Pankaj Kumar. Multi-agent deep reinforcement learning for high-frequency multi-market making. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, pages 2409–2411, 2023.

[17] Leslie Lamport. *Time, clocks, and the ordering of events in a distributed system*, volume 21. 1978. doi:10.1145/359545.359563.

[18] LOBSTER. Lobster: Limit order book system — the efficient reconstructor. https://data.lobsterdata.com/index.php, 2025. Accessed: 2025-12-03.

[19] James Lucas. Lecture notes: Expectation maximization and gaussian mixture models. CSC411 Machine Learning, 2018. URL: https://www.cs.toronto.edu/~jlucas/teaching/csc411/lectures/lec15_16_handout.pdf.

[20] Thomas Lux and Michele Marchesi. Scaling and criticality in a stochastic multi-agent model of a financial market. *Nature*, 397(6719):498–500, 1999. doi:10.1038/17290.

[21] Albert J Menkveld and Marius A Zoican. Need for speed? exchange latency and liquidity. *The Review of Financial Studies*, 30(4):1188–1228, 2017. doi:10.1093/rfs/hhx006.

[22] Szabolcs Mike and J. Doyne Farmer. An empirical behavioral model of liquidity and volatility, 2007. URL: https://arxiv.org/abs/0709.0159, arXiv:0709.0159.

[23] Ciamac C. Moallemi and Mehmet Saglam. The cost of latency in high-frequency trading. *SSRN Electronic Journal*, February 2013. URL: https://ssrn.com/abstract=1571935, doi:10.2139/ssrn.1571935.

[24] Washim Uddin Mondal and Vaneet Aggarwal. Reinforcement learning with delayed, composite, and partially anonymous reward, 2023. URL: https://arxiv.org/abs/2305.02527, arXiv:2305.02527.

[25] Andrew Y Ng and Stuart Russell. Algorithms for inverse reinforcement learning. In *Proceedings of the 17th International Conference on Machine Learning*, pages 663–670. Morgan Kaufmann, 2000.

[26] Richard S Sutton and Andrew G Barto. Reinforcement learning: An introduction. 2018.

[27] Chuanhai Wu. On the convergence properties of the em algorithm. *The Annals of Statistics*, 11(1):95–103, 1983. doi:10.1214/aos/1176346060.

[28] Lei Xu and Michael I Jordan. Convergence analysis of the em algorithm for gaussian mixtures. *Neural Computation*, 8(1):129–151, 1996. doi:10.1162/neco.1996.8.1.129.

[29] Brian Ziebart. *Maximum Entropy Inverse Reinforcement Learning*. PhD thesis, Carnegie Mellon University, 2008. URL: https://www.cs.cmu.edu/~bziebart/publications/thesis-bziebart.pdf.

Appendix A: Background and Preliminaries

This appendix provides essential background material on limit order books, tick-level market events, sources of latency, asynchronous decision-making, and the core ideas behind the EM algorithm and maximum-entropy inverse reinforcement learning. These concepts form the foundation on which the Inverse Delayed Reinforcement Learning (IDRL) framework is constructed.

A.1 Limit Order Books

Modern electronic financial markets match buyers and sellers through a *Limit Order Book* (LOB), a dynamic data structure that records all currently active buy and sell limit orders for a given asset [13]. The LOB is partitioned into a **bid side** and an **ask side**:

- **Bid side:** Orders to buy, sorted in *descending* price.
- **Ask side:** Orders to sell, sorted in *ascending* price.

Within each price level, orders follow a first-in, first-out (FIFO) priority rule. This immediately implies two key market reference points:

$$\text{Best Bid} = \max\{\text{bid prices}\}, \qquad \text{Best Ask} = \min\{\text{ask prices}\},$$

and the *spread* is

$$\text{Spread} = \text{Best Ask} - \text{Best Bid}.$$

A narrow spread reflects high liquidity, while a wide spread indicates lower market efficiency. The best bid and ask form the *touch*, which is the primary microstructure signal for short-term price dynamics [8].

Historically, LOBs evolved from paper-based ledgers to fully electronic matching engines capable of processing millions of events per second. Today, all short-term price formation emerges directly from LOB dynamics [2].

A.2 Tick-by-Tick Events

The LOB evolves through a stream of time-stamped *tick events*. Each event represents an atomic update to the book. Common event types include:

- **New order:** Insert a new limit order at a given price and size.
- **Modify:** Adjust the quantity or price of an existing order.
- **Cancel:** Remove an active order from the book.
- **Trade:** Execute matching buy and sell orders.
- **Exchange cancel/conflict cancel:** Remove orders for regulatory or technical reasons.

Reconstructing the LOB from tick data involves applying each event in strict time order to an initially empty book. Formally, the LOB at time t is a functional of all past events:

$$\mathcal{B}(t) = \mathcal{F}(E_1, E_2, \ldots, E_{n_t}),$$

where E_i is the i-th tick event and n_t is the number of events up to time t.

Tick-level data forms the basis of empirical market microstructure analysis. It enables fine-grained measurement of order flow, depth dynamics, queue competition, and short-term predictability [22], all of which are essential inputs for feature engineering in IDRL.

A.3 Sources of Latency

In electronic markets, *latency* refers to the delay between observing a market event and responding with a trading action [6]. Latency is not uniform across participants; it arises from multiple sources:

Network Latency

Signals travel through fiber-optic or microwave links at near-light speeds. Colocated trading firms achieve sub-microsecond delays; remote participants experience significantly longer transmission paths.

Processing Latency

Interpreting market data requires parsing feeds, updating the local LOB representation, and applying strategy logic. HFT firms deploy optimized hardware (FPGAs, kernel-bypass networking) to reduce this delay, while general-purpose systems incur higher overhead.

Reaction Latency

Automated algorithms react in microseconds; human traders require tens or hundreds of milliseconds to cognitively process information and respond.

The combination of these latencies results in heterogeneous, dynamic reaction times across market participants [11]. In the IDRL framework, these delays appear as *latent stochastic variables*.

A.4 Asynchrony in Electronic Markets

Financial markets are intrinsically *asynchronous*. No two participants share the same view of the LOB at the same moment [17]. Formally:

- Each participant maintains a *local* LOB state $\mathcal{B}^{(i)}(t)$.
- Due to delays, $\mathcal{B}^{(i)}(t)$ corresponds to the true LOB at time $t - \Delta_i(t)$, where $\Delta_i(t)$ is participant i's latency.
- Event ordering may differ across participants.

Thus, the market resembles a distributed system with *no global clock*. This has two major implications:

1. Actions are often taken in response to *stale* states.

2. Fast participants can strategically exploit slower ones (latency arbitrage) [21].

This motivates the core idea of IDRL: actions at time t should be aligned not with the state at t, but probabilistically with states from earlier times $\{t - \Delta\}$.

A.5 The Expectation–Maximization Algorithm

The Expectation–Maximization (EM) algorithm computes maximum-likelihood estimates in models with latent variables [27]. Let $X = \{x_i\}$ be observed data and $Z = \{z_i\}$ latent variables. The complete-data likelihood is $p_\theta(X, Z)$, while the observed likelihood is

$$p_\theta(X) = \sum_Z p_\theta(X, Z).$$

Direct maximization of $\log p_\theta(X)$ is difficult due to the $\log \sum$ structure. EM alternates between:

- **E-step:**
$$q^{(t+1)}(Z) = p(Z \mid X, \theta^{(t)}),$$
computing posterior responsibilities.

- **M-step:**
$$\theta^{(t+1)} = \arg\max_\theta \mathbb{E}_{q^{(t+1)}}[\log p_\theta(X, Z)],$$
maximizing the expected complete-data log-likelihood.

Monotonic ascent follows from Jensen's inequality:

$$\log p_{\theta^{(t+1)}}(X) \geq \log p_{\theta^{(t)}}(X).$$

In IDRL, EM simultaneously infers: - delay responsibilities $\gamma_{t,\Delta}$, - strategy responsibilities $w_{t,i}$, - reward parameters $\theta^{(i)}$, under the time-bracketed MaxEnt IRL model.

A.6 Maximum Entropy Inverse Reinforcement Learning

Maximum-entropy IRL models expert behaviour as the solution to a reward-maximizing stochastic policy of the form [29]:

$$\pi_\theta(a \mid s) = \frac{\exp\left(\theta^\top f(s, a)\right)}{Z_\theta(s)},$$

where $f(s, a)$ are feature vectors and Z_θ is the partition function ensuring normalization.
The objective is to find θ such that:

$$\mathbb{E}_{\pi_\theta}[f] \approx \mathbb{E}_{\text{expert}}[f].$$

In IDRL, the feature expectations become *delay-conditioned*, and the MaxEnt objective is embedded inside an EM loop, enabling simultaneous inference of delays, strategies, and rewards.

Appendix B: Full Proofs of Theorem A and Theorem B

B.1 Notation and Preliminaries

Let $\mathcal{D} = \{(s_t, a_t)\}_{t=1}^{T}$ denote the sequence of observed states and actions. The delay grid is

$$\Delta \in \{\Delta_1, \ldots, \Delta_K\},$$

and for each candidate delay Δ, define the delayed feature vector

$$f_{t,\Delta} := f(s_{t-\Delta}, a_t) \in \mathbb{R}^m.$$

For any $\theta \in \mathbb{R}^m$, the MaxEnt scoring function and partition function are

$$R_\theta(s, a) = \theta^\top f(s, a), \qquad Z_\theta(s) = \sum_{a'} \exp(\theta^\top f(s, a')).$$

The MaxEnt policy is

$$p_\theta(a \mid s) = \frac{\exp(\theta^\top f(s, a))}{Z_\theta(s)}.$$

In the single-strategy model, the EM responsibilities are

$$\gamma_{t,\Delta} = \frac{\pi_\Delta \exp(\theta^\top f_{t,\Delta})/Z_\theta(s_{t-\Delta})}{\sum_{\Delta'} \pi_{\Delta'} \exp(\theta^\top f_{t,\Delta'})/Z_\theta(s_{t-\Delta'})}. \tag{B.1}$$

In the multi-strategy model with strategies $i = 1, \ldots, M$, the gating softmax is

$$w_{t,i} = \frac{\exp(\phi_i^\top h_t)}{\sum_{j=1}^{M} \exp(\phi_j^\top h_t)}. \tag{B.2}$$

Joint responsibilities are

$$\gamma_{t,i,\Delta} = \frac{w_{t,i}\, \pi_\Delta^{(i)}\, \exp(\theta_i^\top f_{t,\Delta})/Z_{\theta_i}(s_{t-\Delta})}{\sum_{i',\Delta'} w_{t,i'}\, \pi_{\Delta'}^{(i')}\, \exp(\theta_{i'}^\top f_{t,\Delta'})/Z_{\theta_{i'}}(s_{t-\Delta'})}. \tag{B.3}$$

Strategy responsibilities and conditional delay responsibilities are:

$$\Gamma_{t,i} = \sum_{\Delta} \gamma_{t,i,\Delta}, \qquad \gamma_{t,\Delta|i} = \gamma_{t,i,\Delta}/\Gamma_{t,i}. \tag{B.4}$$

The Q-functions for the M-steps are:

$$Q(\theta) = \sum_{t,\Delta} \gamma_{t,\Delta} \big[\theta^\top f_{t,\Delta} - \log Z_\theta(s_{t-\Delta})\big], \tag{B.5}$$

$$Q(\theta_i) = \sum_{t,\Delta} \gamma_{t,i,\Delta} \big[\theta_i^\top f_{t,\Delta} - \log Z_{\theta_i}(s_{t-\Delta})\big], \qquad Q(\phi) = \sum_{t,i} \Gamma_{t,i} \log w_{t,i}. \tag{B.6}$$

B.2 Auxiliary Lemmas

Lemma B.8 (Convexity of Log-Partition)**.** *For any fixed state s, the function $\theta \mapsto \log Z_\theta(s)$ is convex, twice differentiable, and satisfies*

$$\nabla \log Z_\theta(s) = \mathbb{E}_{p_\theta(\cdot|s)}[f(s,a)],$$

$$\nabla^2 \log Z_\theta(s) = \mathrm{Cov}_{p_\theta(\cdot|s)}(f(s,a)) \succeq 0.$$

Proof. Standard exponential family results. □

Lemma B.9 (Strict Concavity of Single-Strategy Q-function)**.** *If the feature matrix $\{f_{t,\Delta}\}$ has full rank on the support of $\gamma_{t,\Delta}$, then $Q(\theta)$ in (B.5) is strictly concave in θ.*

Lemma B.10 (ELBO Decomposition for Delay-Latent Model)**.** *For any auxiliary distribution $q(z)$,*

$$\log p_\theta(\mathcal{D}) = \mathcal{F}(q,\theta) + \mathrm{KL}\big(q(z) \,\|\, p_\theta(z \mid \mathcal{D})\big),$$

where $\mathcal{F}$ is the EM lower bound.

B.3 Proof of Theorem A (Single-Strategy IDRL)

Theorem B.3 (Theorem A: Concavity, Monotonicity, Convergence)**.** *The single-strategy IDRL Q-function (B.5) is strictly concave in θ; the EM algorithm produces a sequence $\{\theta^{(k)}\}$ such that*

$$\log p_{\theta^{(k+1)}}(\mathcal{D}) \geq \log p_{\theta^{(k)}}(\mathcal{D});$$

every limit point of $\{\theta^{(k)}\}$ is a stationary point of the observed-data likelihood.

B.3.1 Concavity

Differentiating (B.5),

$$\nabla_\theta Q = \sum_{t,\Delta} \gamma_{t,\Delta} \Big(f_{t,\Delta} - \mathbb{E}_\theta[f \mid s_{t-\Delta}] \Big). \tag{B.7}$$

Using Lemma B.1,

$$\nabla^2_\theta Q = -\sum_{t,\Delta} \gamma_{t,\Delta}\, \mathrm{Cov}_{p_\theta(\cdot|s_{t-\Delta})}(f(s_{t-\Delta},a)) \preceq 0. \tag{B.8}$$

If the features are not degenerate, the Hessian is strictly negative definite on the relevant subspace, establishing strict concavity.

B.3.2 EM Monotonicity

Using the ELBO decomposition of Lemma B.3:

$$\log p_{\theta^{(k)}}(\mathcal{D}) = \mathcal{F}(q^{(k+1)}, \theta^{(k)}),$$

where $q^{(k+1)}(z) = p(z \mid \mathcal{D}, \theta^{(k)})$. The M-step maximizes $\mathcal{F}(q^{(k+1)}, \theta)$, so

$$\mathcal{F}(q^{(k+1)}, \theta^{(k+1)}) \geq \mathcal{F}(q^{(k+1)}, \theta^{(k)}). \tag{B.9}$$

Using the KL decomposition again:

$$\mathcal{F}(q^{(k+1)}, \theta^{(k+1)}) \leq \log p_{\theta^{(k+1)}}(\mathcal{D}). \tag{B.10}$$

Combining yields the monotonicity result.

B.3.3 Convergence

By Wu (1983), if each M-step performs a continuous maximization of a concave function and the parameter domain is closed and bounded (or regularized), then EM converges to the set of stationary points. Strict concavity ensures uniqueness of each M-step solution.

B.3.4 KKT Conditions for M-step

Maximizing $Q(\theta)$ requires solving $\nabla_\theta Q(\theta^*) = 0$, i.e.,

$$\sum_{t,\Delta} \gamma_{t,\Delta}\, f_{t,\Delta} = \sum_{t,\Delta} \gamma_{t,\Delta}\, \mathbb{E}_{\theta^*}[f \mid s_{t-\Delta}]. \tag{B.11}$$

Since the domain is unconstrained ($\theta \in \mathbb{R}^m$), KKT reduces to stationarity.

B.4 Multi-Strategy Preliminaries

The joint responsibilities (B.3) and gating factors (B.2) define the multi-strategy Q-functions:

$$Q(\theta_i) = \sum_{t,\Delta} \gamma_{t,i,\Delta} \Big[\theta_i^\top f_{t,\Delta} - \log Z_{\theta_i}(s_{t-\Delta})\Big], \tag{B.12}$$

$$Q(\phi) = \sum_{t,i} \Gamma_{t,i} \log w_{t,i}. \tag{B.13}$$

B.5 Auxiliary Lemmas for Multi-Strategy Model

Lemma B.11 (Blockwise Concavity in θ_i)**.** *For each fixed i, $Q(\theta_i)$ is strictly concave in θ_i.*

Lemma B.12 (Concavity of Gating Subproblem)**.** *The gating objective (B.13) is concave in ϕ because* log*-softmax log-likelihoods are concave.*

Lemma B.13 (ELBO for Mixture-of-Experts IDRL)**.** *The multi-strategy incomplete log-likelihood can be decomposed as*

$$\log p_{\theta,\phi}(\mathcal{D}) = \mathcal{F}(q,\theta,\phi) + \mathrm{KL}\big(q(z) \,\|\, p_{\theta,\phi}(z \mid \mathcal{D})\big). \tag{B.14}$$

B.6 Proof of Theorem B (Multi-Strategy IDRL)

Theorem B.4 (Theorem B: Multi-Strategy EM Convergence)**.** *For the mixture-of-experts IDRL model with Q-functions (B.12)-(B.13):*

1. *Each θ_i-subproblem is strictly concave.*
2. *The gating M-step is concave in ϕ.*
3. *EM monotonically increases the observed-data likelihood.*
4. *(θ_i, ϕ) converges to a stationary point of the likelihood.*

B.6.1 Blockwise Concavity in θ_i

Using the same Hessian argument as in (B.8),

$$\nabla^2_{\theta_i} Q(\theta_i) = -\sum_{t,\Delta} \gamma_{t,i,\Delta} \operatorname{Cov}_{\theta_i}[f \mid s_{t-\Delta}] \preceq 0, \tag{B.15}$$

and strict concavity follows from feature non-degeneracy.

B.6.2 Concavity of Gating Update

The gating objective is

$$Q(\phi) = \sum_t \sum_i \Gamma_{t,i} \Big(\phi_i^\top h_t - \log \sum_j e^{\phi_j^\top h_t} \Big). \tag{B.16}$$

The Hessian of $\log \sum_j e^{\phi_j^\top h_t}$ is positive semidefinite, hence $Q(\phi)$ is concave.

B.6.3 EM Monotonicity

Using Lemma B.6, for each EM iteration:

$$\mathcal{F}(q^{(k+1)}, \theta^{(k+1)}, \phi^{(k+1)}) \geq \mathcal{F}(q^{(k+1)}, \theta^{(k)}, \phi^{(k)}), \tag{B.17}$$

and the ELBO decomposes to the incomplete-data likelihood as before, establishing monotonicity.

B.6.4 Convergence

By the block EM convergence theorem of Xu & Jordan (1996), because each block update maximizes a concave function over a closed parameter set, the sequence converges to the stationary set of the likelihood.

B.6.5 KKT Conditions for Gating Step

Let

$$L(\phi, \lambda) = Q(\phi) + \sum_t \lambda_t \Big(\sum_i w_{t,i} - 1 \Big). \tag{B.18}$$

Stationarity requires:

$$\frac{\partial L}{\partial \phi_i} = \sum_t \big(\Gamma_{t,i} - w_{t,i} \big) h_t = 0. \tag{B.19}$$

Dual feasibility and primal feasibility hold by softmax construction.

□