1 Data Acquisition and Pre-Processing

1.1 From Tick Messages to 30-Minute Bars

We download Binance's public `aggTrades` archive for the seven most liquid spot pairs (BTC, ETH, ADA, BNB, DOGE, SOL, XRP) over the interval **1 June 2024 – 1 July 2025**. The `isBuyerMaker` flag preserved in each aggregated trade lets us separate buyer- and seller-initiated volume—crucial for order-flow features.

Download. An asynchronous scraper with eight parallel TCP sockets mirrors $\sim 7\,\mathrm{GB}$ of compressed `aggTrades` archives ($\approx 80\,\mathrm{GB}$ once unzipped for all seven symbols) in under ten minutes on a $100\,\mathrm{Mb/s}$ connection.

Bar construction. Every record is resampled into 1800s = 30-minute bars (Table 1).[1] If the bid–ask spread collapses to zero we enforce the modal positive spread (empirical tick), guaranteeing $\mathrm{ask}_t > \mathrm{bid}_t$.

Table 1: Fields emitted by the resampler.

Field	Type	Meaning
`ts`	datetime	left-closed bar time-stamp
`price_last`	float32	last trade price in the interval
`buy_qty`	float64	buyer-initiated quantity (sum)
`sell_qty`	float64	seller-initiated quantity (sum)
`best_bid`	float32	final quoted bid
`best_ask`	float32	final quoted ask

1.2 Feature Engineering

Per asset we derive three feature families, each computed over **1- and 5-bar windows**. To remove slow regime drift we apply a rolling z-score normalisation with a horizon equal to *60 times* the aggregation window:

$$z_t^{(\ell)} = \frac{x_t^{(\ell)} - \mu_{t,w_\ell}}{\sigma_{t,w_\ell}}, \qquad w_1 = 60,\ w_5 = 300,$$

where $\ell \in \{1, 5\}$ denotes the look-back length in bars and (μ, σ) are the running mean and standard deviation.

(i) **Log-returns** $\Delta \log p$ capture short-term momentum and mean reversion.

(ii) **Signed notional volume** $\log\big[(B_t + S_t)\, p_t\big]$ measures trading activity weighted by price.

[1] The bar width is a trade-off: short enough to retain intraday texture, long enough to avoid excessive commission drag.

(iii) **Order-Flow Imbalance (OFI)**

$$\mathrm{OFI}_t = \frac{B_t - S_t}{B_t + S_t},$$

a proxy for latent liquidity pressure at the top of the book.

Seasonal channels. Crypto trades *24/7*, yet volume and volatility still follow pronounced diurnal and weekly cycles. We encode these calendar patterns with four sinusoidal features broadcast across all assets:

$$\left(\sin \tfrac{2\pi\,\mathrm{tod}}{24\mathrm{h}}, \cos \tfrac{2\pi\,\mathrm{tod}}{24\mathrm{h}}, \sin \tfrac{2\pi\,\mathrm{dow}}{7}, \cos \tfrac{2\pi\,\mathrm{dow}}{7}\right),$$

where tod is seconds since midnight and dow the weekday index $[0, 6]$.

Input tensor. With $m = 7$ tradable assets the model therefore observes

$$X \in \mathbb{R}^{C\times m\times n}, \quad C = 4\ (\text{seasonal})\ +\ 6\ (\text{features}) = 46, \qquad n = 50.$$

That is, each training sample supplied by `RollingWindowDataset` contains

$$46 \times 7 \times \textit{window}$$

numbers: 4 seasonal channels replicated for every symbol plus 3 feature families $\times$ 2 look-back windows $\times$ 7 assets, all over a CNN window which can be 7,36 or 72 in the current config$\times$ bar width history slab.

1.3 Key Pipeline Hyper-parameters

Table 2: Summary of data-pipeline choices.

Component	Setting	Rationale
Symbols	BTC / ETH / ADA / BNB / DOGE / SOL / XRP	liquid, cross-sector
Sample period	1 Jun 2024 – 1 Jul 2025	covers bullish & bearish regimes
Bar width	30 minutes	balances micro-structure detail an
Return / volume / OFI windows	1, 5 bars	capture sub-hour to few-hour swi
Normalisation window	60 $\times$ number of bars	removes low-frequency drift

2 Reinforcement-Learning Architecture

2.1 Problem Setting

At every bar close the agent observes a *state tensor* $X_t \in \mathbb{R}^{C\times m\times n}$ built from the feature cube described in Section 2. It selects a *portfolio vector* $w_t = \left(w_t^{(1)}, \dots, w_t^{(m)}, w_t^{()}\right)$ that allocates the next bar's wealth across the m assets and the cash bench. Admissible actions satisfy $w_t^{(i)} \in [-1, 1]$, $\sum_i w_t^{(i)} = 1$.

Price relatives $y_t = (P_t^{(1)}/P_{t-1}^{(1)}, \ldots, P_t^{(m)}/P_{t-1}^{(m)})$ realise at bar close t. Let $w'_{t-1} = (w_{t-1} \odot y_t)/(w_{t-1} \cdot y_t)$ be the "drifted" weights just before re-balancing, and define the (one-way) turnover

$$\mathrm{turn}_t = \big\| w_t^{(\mathrm{assets})} - w_{t-1}^{\prime(\mathrm{assets})} \big\|_1.$$

The round-trip commission factor applied at bar t is

$$\mu_t = 1 - c\,\mathrm{turn}_t, \qquad 0 < c \ll 1,$$

so the growth during bar t that is attributed to decision w_{t-1} equals

$$r_{t-1} = \log\Big(\mu_t \; w_{t-1}^{\top} \begin{bmatrix} y_t \\ 1 \end{bmatrix}\Big).$$

where w'_{t-1} is the inventory drifted by the realised prices and c the round-trip commission. Maximising the undiscounted sum $\sum_t r_t$ where r_t is the logarithmic return, is equivalent to Kelly growth optimisation and aligns with the Sharpe objective used for evaluation.

2.2 Policy Network: EIIE-CNN

- **Input.** The first convolution ingests the feature cube $X_t \in \mathbb{R}^{C \times m \times W}$. Just before the final layer the previous allocation $w_{t-1} \in \mathbb{R}^m$ is *broadcast* as an extra channel, allowing the network to anticipate re-hedging cost.
- **Spatial inductive bias.** Two one-dimensional convolutions, $\mathrm{Conv}(C{=}46 \to 8,\ k = 3)$ and $\mathrm{Conv}(8 \to 32,\ k = n{-}2)$, sweep along the look-back axis while sharing weights across assets—akin to treating *assets* as image rows and *time* as columns. Instance-norm after each conv stabilises the scale across heterogeneous crypto pairs.
- **Fully convolutional head.** Concatenating the 32 latent maps with the broadcast w_{t-1} (1 map) yields 33 channels; a final $1{\times}1$ convolution produces one raw score $s_t^{(i)}$ per asset. A learnable *asset bias* breaks symmetry, while a separate *cash bias* b is appended before projection so the network can choose *not* to trade.
- **Output projection.**

  (a) Softmax-Cash (long-only baseline): the augmented logits $[s_t; b]$ are sent through softmax to obtain a simplex-valued portfolio w_t.

  (b) ℓ_1-Projection (long–short head): first concatenate $[s_t; b]$, then

$$\lambda \;=\; \tanh\!\big(\|s_t\|_1\big), \qquad w_t^{(\mathrm{assets})} = \frac{\lambda\, s_t}{\|s_t\|_1}, \quad w_t^{()} = 1 - \sum_i w_t^{(i)}.$$

  The tanh factor $\lambda \in (0, 1)$ *softly* scales the weights so that $\|w_t^{(\mathrm{assets})}\|_1 \le 1$; leverage is therefore limited without hard clipping.

The design follows the "Ensemble of Identical Independent Evaluators" principle: a single set of filters evaluates each asset in parallel, promoting parameter-sharing and permutation invariance while keeping cross-asset interactions implicit in the convolutional width dimension.

The architecture is inspired by the "Ensemble of Identical Independent Evaluators" (EIIE) idea: each asset path is evaluated by the *same* set of filters, which improves data efficiency and preserves permutation symmetry.

2.3 Online Learning Algorithm

Time-shifted experience tuples. Because the reward r_{t-1} is only known after y_t arrives, we store $(X_{t-1},\, y_t,\, w_{t-1})$ in a replay buffer of capacity 10^4.[2]

Reward definition. Let

$$\mu_{t-1} \;=\; 1 - c\,\big\|w'_{t-1} - w_{t-1}\big\|_1, \qquad w'_{t-1} = \frac{y_t \odot w_{t-1}}{y_t^\top w_{t-1}},$$

where c is the round-trip commission and w'_{t-1} the **inventory that would result *without* re-balancing**. The factor $\mu_{t-1} \in (0,1]$ is therefore the *fraction of wealth left after paying fees*.

Our per-step reward is the log-growth of wealth, $r_{t-1} = \log\!\big(\mu_{t-1}\, w_{t-1}^\top [y_t; 1]\big)$. Maximising the *undiscounted* sum $\sum_t r_t$ is exactly the **Kelly criterion**—it maximises the long-run geometric growth rate of capital and, under mild regularity, delivers the highest expected utility for any concave utility function.[3]

Off-policy policy-gradient. At each bar we draw a mini-batch of experiences with probability $\propto (1-\beta)^k$ (newer samples get more weight), compute the current deterministic action $w_\theta(X)$, and *directly* maximise the analytic surrogate

$$J(\theta) = \mathbb{E}_{(X,y,w)\sim\mathcal{D}}\Big[\log\!\big(\mu_\theta\, w_\theta^\top y\big) \;-\; \lambda_{\text{turn}}\,\big\|w_\theta - w\big\|_2^2\Big].$$

Here λ_{turn} controls an **L_2** turnover penalty. We favour L_2 over L_1 because it supplies *smooth* gradients – an L_1 term would create kinks that destabilise Adam and require sub-gradient schemes.

No critic is necessary: the objective is already the Monte-Carlo estimate of the Kelly log-growth, so vanilla REINFORCE with variance reduction is sufficient. Gradients are clipped to $\|g\|_2 \le 5$ and updated with Adam.

Why not DDPG / PPO? Unlike many continuous-control tasks, our reward is a **deterministic, differentiable** function of the previous action once the next bar closes: there is no stochastic state transition to model. Using actor–critic methods that learn an additional value baseline would (1) waste data on fitting a function we can compute analytically, and (2) introduce bias from boot-strapping. The deterministic policy-gradient we adopt is therefore both *simpler* and *more sample-efficient*.

[2] See Appendix A for a detailed discussion of experience-replay in finance.

[3] Kelly, *Information Theory and Gambling*, 1956.

2.4 Design Rationale

(i) **Permutation symmetry.** Sharing convolutional kernels across assets prevents the network from over-fitting idiosyncrasies of BTC or ETH and generalises to unseen symbols.

(ii) **Turnover awareness.** Feeding w_{t-1} as an input channel plus the explicit commission factor μ_{t-1} and the L_2 turnover term guides the network to trade *only when edge exceeds fee.*

(iii) **Recency bias in replay.** Geometric sampling emphasises newer market regimes without discarding long-term experience—vital in crypto's non-stationary landscape.

(iv) **End-to-end Kelly objective.** Directly optimising expected log-growth aligns with risk-adjusted return and avoids arbitrary variance penalties or hand-tuned Sharpe targets.

(v) **Smooth regularisation.** L_2 on Δw punishes large reallocations while keeping gradients well-behaved; L_1 would create plateaus and slow learning.

3 Empirical Results and Discussion

3.1 Hyper-parameter interactions (zero-fee back-test)

All metrics in this subsection are evaluated before *commission. Net performance will be lower once a realistic round-trip fee is applied; see Section 5.4 for a cost-adjusted discussion.*

Look-back window n. Across both action heads, `window`= 72 (one trading day of 30-minute bars) produces the most *stable* validation Sharpe, whereas `window`= 6 is too myopic and `window`= 36 tends to over-react to intraday noise. The long–short head is markedly more sensitive: with a short n it exploits fleeting micro-structure artefacts that do *not* survive out-of-sample, yielding negative validation Sharpe in the first block of Table 3.

Learning rate η & batch size B. The grid shows a clear interaction: larger batches tolerate—and indeed *need*—larger η to avoid the "small-gradient trap" inherent in Kelly objectives. Train Sharpe peaks around (B=128, η=5×10^{-5}) for the long–short head and (B=64, η=1×10^{-4}) for softmax, mirroring the rule-of-thumb $\eta \propto \sqrt{B}$ that keeps the SGD noise scale roughly constant.

Turnover penalty and action space. Constraining the portfolio to the unit simplex (SOFTMAX–CASH) acts as an implicit ℓ_1 regulariser on turnover, improving generalisation even *before* fees are charged. Allowing short exposure (L_1-PROJ) unlocks higher *in-sample* Sharpe (train = 2.34) but validation Sharpe deteriorates unless the turnover penalty is increased— evidence that some of the discovered short signals are back-test artefacts.

Take-away. Under zero commission the softmax head already delivers the highest risk-adjusted validation Sharpe. Once a realistic fee is debited (Section 5.4) its advantage widens because the implicit turnover control keeps gross returns above the cost drag, whereas the long–short head's performance collapses.

3.2 Transaction Costs and Model Limitations

Even the best softmax run yields an out-of-sample growth factor $\widehat{G} = 1.24$ *before* any fees but cannot clear the 20 bp round-trip cost typical for Binance market orders ($c = 2 \times 10^{-5}$). In Kelly parlance the implied edge is

$$\Delta g = \mathbb{E}[exp(r_t) - 1] \; = \; \approx \; 7.11 \times 10^{-5},$$

comparable to the fee itself; after costs the growth expectation becomes negative.
Where does the edge vanish?

(i) **Feature sufficiency.** The three hand-crafted families (return, volume, OFI) may be adequate for classical equity micro-alpha but appear too coarse to exploit modern crypto order-flow — especially once signals are aggregated to 30-minute bars.

(ii) **Capacity of the EIIE head.** Convolutional filters are shared across assets and time. While this promotes data efficiency, it also *restricts* the hypothesis class: cross-asset lead–lag patterns cannot be expressed.

(iii) **Objective mis-alignment.** We train on log-growth (Kelly) but validate on Sharpe. Maximising $\sum r_t \, rewardstakingmanysmall, positively\text{-}skewedbets; Sharpepenalisesvolatilitylinearly. Arewardthatdirectly$ cash-fee term, assuming infinite liquidity and no slippage. In practice large trades widen the spread or cross multiple levels; the real cost curve is convex. Ignoring this curvature encourages over-trading.

Take-away. Under a realistic fee schedule the current architecture extracts an edge of the same order as the commission. Overcoming that hurdle likely requires (i) richer state representations (cross-asset attention, latent order-book tensors), (ii) a reward that *directly* targets risk-adjusted return ($\widehat{\text{Sharpe}}$ or VaR-penalised growth), and (iii) an execution layer that models spread impact instead of flat per-share fees.

Table 3: Top-5 long–short grid runs sorted by Train Sharpe.

n	η	β	B	Train $\mathcal{S}$	Val. $\mathcal{S}$	Val. G
72	5×10^{-5}	0.050	128	2.34	−0.67	0.95
36	1×10^{-4}	0.050	256	2.02	−0.55	0.95
72	1×10^{-4}	0.005	128	1.83	**3.15**	**1.28**
72	1×10^{-4}	0.050	256	1.28	1.87	1.15
72	5×10^{-5}	0.005	128	1.27	1.11	1.07

3.3 Future Work

A credible path to net-of-fee profitability must attack *both* sides of the edge equation: extract richer predictive structure *and* internalise execution frictions at training time. Below we sketch concrete directions.

(iv1) **Representation learning beyond EIIE.**

Table 4: Train Sharpe on long–short head as a function of batch size B and learning-rate η.

B	1×10^{-5}	5×10^{-5}	1×10^{-4}
512	−0.094	0.676	0.323
256	−0.146	0.760	0.981
128	**0.221**	**1.014**	**1.108**
64	−0.133	0.657	0.496

Table 5: Train Sharpe on long–short head versus look-back window n and η.

n	1×10^{-5}	5×10^{-5}	1×10^{-4}
72	0.155	**0.977**	0.579
36	−1.150	0.701	0.818
6	0.881	0.653	0.785

- *Cross-asset self-attention.* Model the $m \times n$ cube as a sequence of *asset-tokens*, letting the network learn dynamic pair-wise lead–lag relations (e.g. BTC leading alts during risk-on bursts). A thin multi-head layer on top of the convolutional stem already doubles the hypothesis space.
- *Graph neural networks.* Encode assets as nodes with edges initialised from fundamental similarity (sector, exchange flow) and let message passing refine the correlation graph end-to-end.
- *Dilated TCN / Transformer encoders.* Capture multi-day cycles (funding resets, option expiry) while keeping memory constant; a causal dilated stack can see thousands of bars without exploding n.

2. **Cost-aware action spaces.**

   - *Directly output Δw_t.* Re-parameterising the policy in *change-space* turns the otherwise non-differentiable turnover into a linear control cost inside the network.
   - *Inaction band / fuzzy bandwidth.* Have the actor predict a centre weight plus a tolerance ϵ; rebalance only if $\|w_{t-1} - \hat{w}_t\|_\infty > \epsilon$. The band can be learned jointly and prunes low-information trades during dull regimes.

3. **Execution-layer simulation.**

   - *Differentiable slippage model.* Replace the linear fee μ_t with a convex impact–plus–rebate curve learned from historic quote depth; back-propagating through this surrogate penalises burst trades long before deployment.
   - *Two-tier agent.* A high-level portfolio RL outputs Δw; a low-level micro-agent (e.g. IMPALA or SAC on LOB snapshots) decides limit vs. market style, bridging the back-test/real gap.

4. **Risk-consistent objectives.**

Table 6: Top-5 softmax grid runs sorted by validation Sharpe.

n	η	β	B	Train $\mathcal{S}$	Val. $\mathcal{S}$	Val. G
72	1×10^{-4}	0.005	256	2.15	**1.67**	**1.24**
72	5×10^{-5}	0.005	64	1.53	1.49	1.23
36	1×10^{-4}	0.050	64	1.48	−0.71	0.91
36	5×10^{-5}	0.005	256	1.45	0.93	1.12
36	1×10^{-4}	0.050	128	1.43	−1.29	0.85

Table 7: Train Sharpe on softmax head as a function of batch size B and η.

B	1×10^{-5}	5×10^{-5}	1×10^{-4}
512	0.608	1.091	1.049
256	0.559	0.987	1.096
128	0.543	0.931	**1.284**
64	0.521	**1.048**	1.226

- *Sharpe or Sortino reward.* Maintain an online exponential window of mean/variance and feed the analytically differentiated Sharpe ratio to the policy gradient; this directly optimises what we evaluate.
- *Distributional RL.* Learn $Z_\theta(r)$ and optimise a coherent risk measure such as CVaR_{95}, preventing tail blow-ups masked by the log-sum objective.

5. **Regime-aware meta-policies.**
   - *Volatility/CVI gating.* Train a small ensemble specialised for {high-vol, low-vol, trend, chop}. A lightweight classifier picks the active head on-line, allowing conditional alpha where unconditional alpha is weak.
   - *Rapid fine-tuning.* Use MAML or Reptile so that 5–10 gradient steps on fresh data re-align the policy when regime breaks occur (e.g. ETF approval shocks).

6. **Data enrichment.**
   - Merge *perpetual-futures* funding rates and open interest for leverage sentiment.
   - Add *macro sentiment* (USDT dominance, funding spreads) as global channels—cheap but often highly predictive.

Bottom line. The current model uncovers a statistically significant but fee-sized edge; bridging that gap demands (i) architectures that detect cross-asset structure, (ii) an objective that prices risk and impact *during* training, and (iii) tighter integration with realistic execution mechanics. Only by addressing all three simultaneously can we hope to beat the 20 bp round-trip hurdle on liquid crypto pairs.

Table 8: Train Sharpe on softmax head versus look-back window n and η.

n	1×10^{-5}	5×10^{-5}	1×10^{-4}
72	0.814	1.063	**1.240**
36	0.384	**1.148**	**1.250**
6	0.476	0.832	1.002