AI-Driven Cryptocurrency Hedge Fund — Vision & Architecture

Fedorov Artem Maximovich

Hedge Fund Model:

Vision & Design Axes

We see modern AI driven crypto hedge fond as multi-agent system, that fuses deep learning, RL, and probabilistic modeling; data sources include market/derivatives, on-chain flow, and news/sentiment on a unified, right-aligned timeline. We divide our approach to modelling strategies into two main domain:

Alpha approach:

1) RL, full-environment-aware with news integration (RAG) and policy fine-tuning on historical environments:

$$Objective : \max J(\pi_\theta) = \mathbb{E}\left[\sum_t \gamma^t r_{t+1}\right], \text{with state } s_t = [x^{\text{mkt}}t, , e^{\text{news}}t]\ .$$

2) Probabilistic ML for market dynamics

$$Rt+1 \sim p\theta(\cdot \mid x_t),\ \text{position sizing } w_t = \text{clip}!\left(\beta, \frac{\mathbb{E}[R_{t+1} \mid x_t]}{\hat{\sigma}_t}, , -w_{\max}, , w_{\max}\right)$$

Autonomy:

1) Fully autonomous policy → automatic execution under hard risk budgets.

2) Expert system → policy output as a recommendation to an operator/broker.

Hedge Fund Model:

Strategy Families & Algorithms

- **News/Semantics (RAG → signals):** Topic clustering (*LDA / Summarization-BERT*), dense embeddings (*SBERT*), relevance indices via GLAD-like Bayesian optimization; then horizon-specific agents (RNN/CNN) trained with PPO/A2C on $[x_t^{\text{mkt}}, e_t^{\text{news}}]$.

- **Neural dynamics (continuous-time)**: Diffusion models / neural operators for learning drift and diffusion: $dS_t/S_t = \mu_\theta(x_t), dt + \sigma_\theta(x_t), dW_t$, or an operator $\mathcal{G}\theta : (xt - W : t) \mapsto (\mu, \sigma)$.
  Use for (i) trajectory simulation, (ii) risk management (*VAR/ES*), (iii) portfolio control at large N.

- **Classical ML / quant baselines**: *KNN* and tree/linear ensembles for short-horizon patterns; dispersion-based portfolios (*IVol, GMV, HRP, ERC*) both as benchmarks and as base experts in ensembles.

- ***Usage of technical analysis tools***: Usage of MA/EMA, RSI/ROC, MACD, oscillators, volatility metrics, information retrieved via technical analysis and motif detection algorithms — included as informative features and stability anchors.

  More on that on next slide

Hedge Fund Model:

High-Information Features (for RL & Expert Systems)

Effective learning in financial environments begins with the definition of the observable state. We want well compressed information, stored in low dimensional state, to help relatively large model to fit small historical data on training environment. We consider next tools:

- **Motif detection & shapelets:** Structural motifs in r_t, spreads, volumes; binary/numeric features increase sample efficiency.
- **Deterministic TA patterns:** Interpretable, low-variance features that act as weak learners/filters.
- **Volatility & regimes:** EWMA/GARCH, regime labels (HMM / Bayesian change-point), turbulence indices, autocorrelation profiles.
- **Risk-aware features:** Drawdown, expected shortfall (CVaR), ulcer index, serenity index, recovery factor; derivatives signals (funding, basis, open interest, lead–lag with spot/cash).
- **Unified feature vectors:** $x_t = \left[x_t^{\text{mkt}}, x_t^{\text{deriv}}, x_t^{\text{onchain}}, e_t^{\text{news}}, m_t^{\text{risk}}\right]$ — a high-informativeness design that supports training larger architectures on limited data.

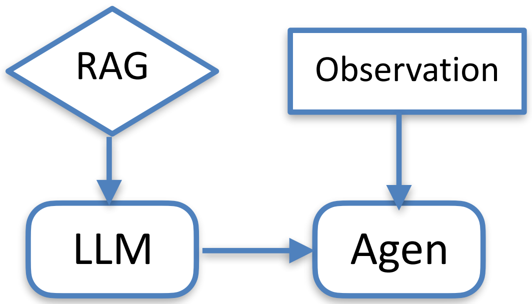
Merging of all computed indexes might significantly increase data dimension => make observation space much more complex
To mitigate this we apply feature selection mechanics, based on single regressor predictive capabilities, to take only top-n features.

Hedge Fund Model:

Multi-Agent Interaction. Mixture-of-Experts paradigm

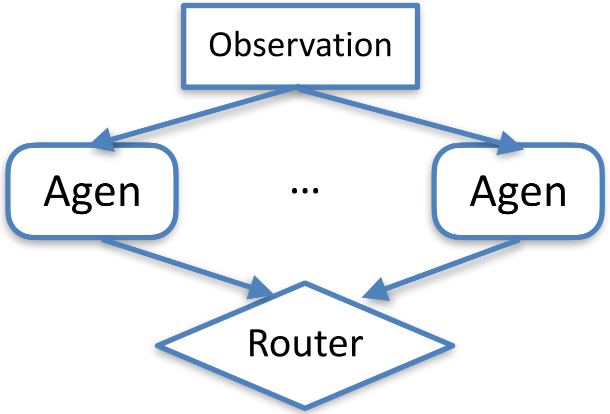
Heterogeneous and non-stationary markets favor an architecture that aggregates specialized decision rules. A Mixture-of-Experts framework assigns routing weights $\alpha_k(x_t) = \mathrm{softmax}(Ux_t)$ and forms the composite action $a_t = \sum \alpha_k(x_t), a_k(x_t)$, where experts capture distinct regimes and horizons. We separate MoE from transfer-learning technics though, that we incorporate too in such tasks as News aggregation and information extraction via LLMs.

Scenario A



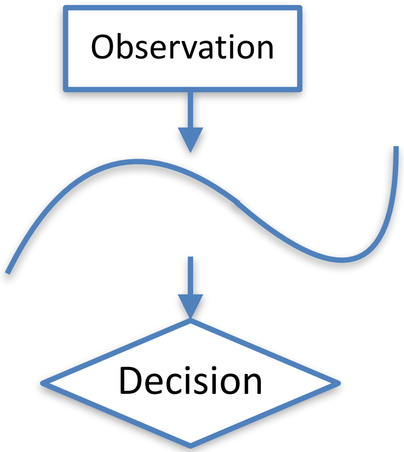
Semantic extractor → NN. RAG produces e_t^{news}; feed into signal NNs/ RL agents.

Scenario B



Classical MoE with router.
Routing weights $\alpha_k(x) = \mathrm{softmax}(Ux)$ final action $a = \sum \alpha_k(x), a_k(x)$, where a_k are independent experts (RL/ML/rules)

Scenario C



Neural operator + probabilistic head.
Neural operator outputs (μ, σ) for a stochastic differential model; an action head learns $\pi_\theta(a \mid \mu, \sigma, x)$ and selects a risk-normalized action.

Risk Management:

Mitigation of risks. Metrics & Estimation

Risk mitigation:

AI handles risk by making it a **first-class control signal**: it now casts volatility and tail risk, estimates liquidity, and detects regimes to **gate exposure** as part of its inner decision making mechanism. For that we employ variety of different risk estimators and treat them as multi criteria optimisation problem on train environment. Agent proposals are scaled by control multipliers—e.g., **vol targeting** $\alpha_t = \min 1{,}\sigma^*/(\hat{\sigma}_t + \varepsilon)$ and **uncertainty throttles** $\eta_t \propto 1/(u_t + \varepsilon)$ —then **projected to hard constraints** $\mathrm{ES}q(w) \le \mathrm{ES}^*,\ |w_i| \le w\max,\ \|w\|_1 \le L$, with liquidity caps.

Core risk metrics:

- **Volatility:**
  A. EWMA $\hat{\sigma}_t^2 = \lambda \hat{\sigma}_{t-1}^2 + (1-\lambda) r_t^2$;
  B. GARCH(1,1) $\sigma_t^2 = \omega + \alpha r_{t-1}^2 + \beta \sigma_{t-1}^2$;
  C. realized vol $\mathrm{RV}t = \sum rt, i^2$;
  D. Parkinson $\hat{\sigma}_{HL}^2 = \frac{1}{4\ln 2}(\ln(H/L))^2$.
- **Tail & path risk:** $\mathrm{VaR}q, \mathrm{ES}q = \mathbb{E}[R \mid R \le \mathrm{VaR}q]$; drawdown $\mathrm{DD}t = \dfrac{\max \tau \le tV\tau - V_t}{\max \tau \le tV\tau}$; Calmar $= \dfrac{\text{Ann. Return}}{\text{MaxDD}}$.
- **Regime signals:** turbulence $T_t = (x_t - \mu_t)^\top \Sigma_t^{-1}(x_t - \mu_t)$; $p(z_t)$ via HMM/change-points.
- **Cost pressure:** turnover; fees/slippage model $c_t = c_{\text{fee}} \cdot \mathrm{turnover}t + c\text{slip} \cdot |\Delta w_t|$.

Risk Management:

Volatility and Liquidity

A. **Volatility indexes we implement in our policy training/inference:**
   A. **EWMA** $\hat{\sigma}_t^2 = \lambda \hat{\sigma}_{t-1}^2 + (1 - \lambda) r_t^2$**;**
   B. **GARCH(1,1)** $\sigma_t^2 = \omega + \alpha r_{t-1}^2 + \beta \sigma_{t-1}^2$**;**
   C. **realized vol** $\mathrm{RV}t = \sum rt, i^2$**;**
   D. **Parkinson** $\hat{\sigma}_{HL}^2 = \frac{1}{4 \ln 2} (\ln(H/L))^2$**.**

B. **Liquidity estimates we propose as efficiency markers (most commonly used).**
   A. **Amihud illiquidity** $\mathrm{ILLIQ}_t = \frac{|r_t|}{\text{DollarVolume}_t}$;
   B. **quoted/effective bid–ask spread**;
   C. **order-book depth and fill ratio**;
   D. **Kyle's** λ (price impact per unit volume);

Portfolio management:

What is an Optimal Portfolio

An optimal portfolio chooses weights w that maximize risk-adjusted utility under realistic costs and constraints. In a in canonical form of single-period view this reads as:

$U(w) = w^\top \mu - \frac{\lambda}{2}, w^\top \Sigma w - \mathrm{costs}(w)$ subject to $\mathbf{1}^\top w = 1$ and leverage liquidity limits. For example, this optimisation problem lays in basis of dispersion algorithms like *IVol, GMV, HRP, ERC,* that solve direct optimisation problem over time-running estimations on short term historic data.

We can derive different objectives, tilting behavior of final strategy:

1) Sharpe-optimal portfolios maximize $\frac{w^\top \mu}{\sqrt{w^\top \Sigma w}}$;
2) mean–CVaR designs minimize $\mathrm{CVaR}_q(R_p)$ at a target return;
3) bounded-Kelly maximizes $\mathbb{E}[\ln(1 + w^\top r)]$ but caps leverage to avoid ruin.

Each of such core principals can be developed into its own Machine Learning Optimisation problem => can describe desired loss

Portfolio management:

Risk-Based Portfolios & Rebalancing

Risk-based constructions reduce reliance on fragile mean estimates. Inverse-Vol sets $w_i \propto 1/\sigma_i$; Equal Risk Contribution (ERC) equalizes marginal risk shares with $\mathrm{RC}_i = w_i(\Sigma w)_i/\sqrt{w^\top \Sigma w}$; Hierarchical Risk Parity (HRP) allocates top-down along a tree to avoid correlation crowding. Rebalancing is a control problem: a time cadence (every n bars) is combined with deviation bands and volatility targeting $\alpha_t = \min 1,,\sigma^*/(\hat{\sigma}_t + \varepsilon)$, adapting exposure to regime while containing costs. Constraints stay active throughout: tail-risk bounds (e.g., ES_q), leverage caps, and capacity limits $|\Delta N_i| \le \rho, \mathrm{ADV}_i$.

Portfolio outcomes are dominated by estimation error in (μ, Σ). We stabilize inputs with shrinkage covariance (e.g., $\Sigma \leftarrow (1-\delta)\hat{\Sigma} + \delta, \Sigma_0$), factor structures, and Bayesian return views (Black–Litterman) that pull aggressive means toward informed priors. The optimizer itself is regularized—adding ℓ_1/ℓ_2 penalties to avoid concentration—and made tradable via explicit fee and slippage terms and turnover bands, so the solution remains stable out of sample instead of overreacting to noise.

Portfolio management:

Statistical Portfolio with AI Assist

Statistical approach:

AI improves the *inputs* and the *orchestration* while the optimizer remains transparent. Agents forecast $\hat{\mu}_t, \hat{\Sigma}_t$, and regime/liquidity states; generative models (diffusion or neural operators) provide scenario sets. We then solve a robust problem

$\min_{w} \max_{\mu \in \mathcal{U}_\mu,\, \Sigma \in \mathcal{U}_\Sigma} \frac{1}{2} w^\top \Sigma w - \theta, w^\top \mu$ (or other functional objective)

where uncertainty sets $\mathcal{U}$ come from ensembles or quantiles. A light meta-policy selects among IVol/HRP/ERC/mean–CVaR depending on the inferred regime; an execution overlay projects decisions to hard constraints and applies volatility targeting. This path is interpretable and auditable, **can end up as fully functional expert system,** at the cost of slightly slower adaptation than a fully end-to-end policy.

Portfolio management:

Pure RL Approach

Pure RL approach:

This approach can be though as the opposite to statistical. Here the policy maps state directly to allocations: $s_t = [x_t, \hat{\sigma}_t, \text{regime}, \text{liquidity}] \mapsto a_t = w_t$. The reward is risk-aware, for example

$$r^{\text{pm}}t + 1 = w_t^{\top} rt + 1 - \lambda_{\text{ES}} \cdot \max(0, \text{ES}q, t - \text{ES}^*) - \lambda\Delta w \|\Delta w_t\|_1,$$

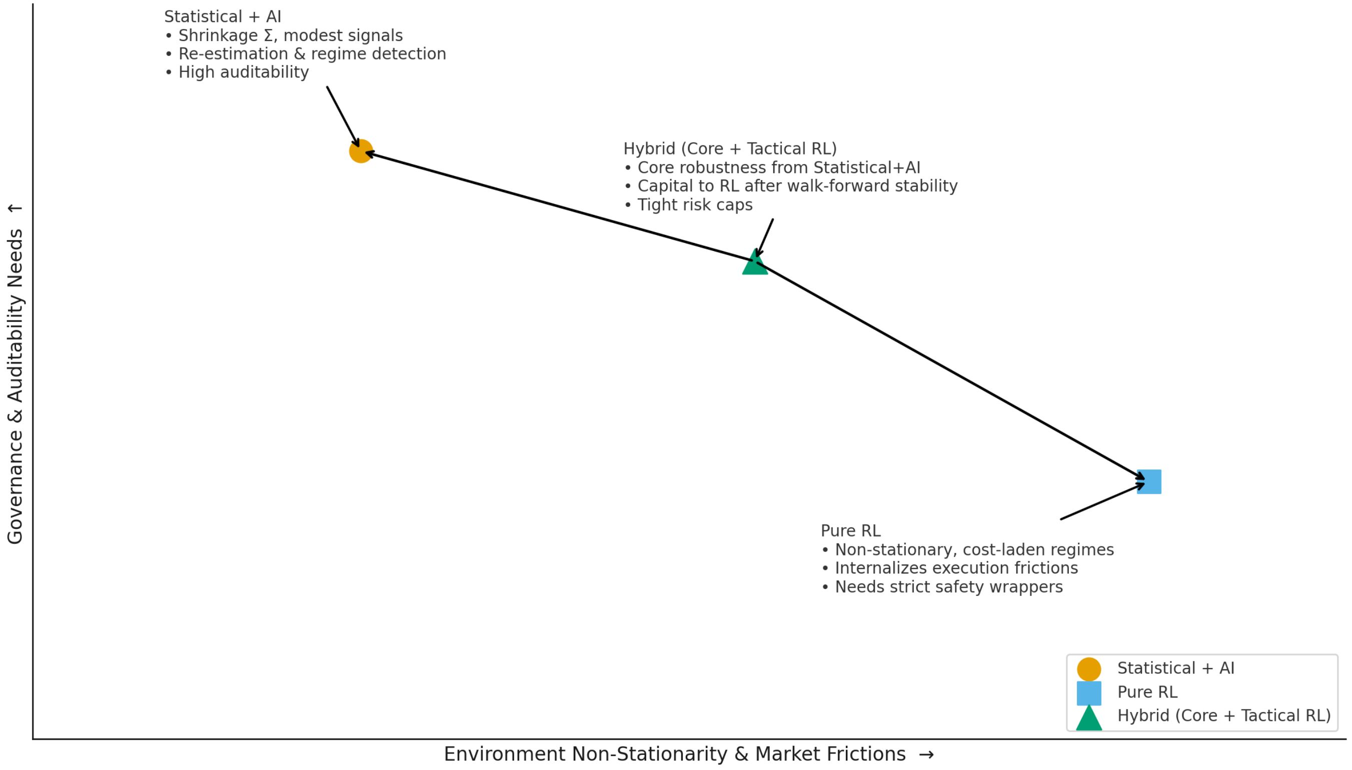
and actions are projected each step to a feasible set with $\text{ES}q(w) \le \text{ES}^*$, $|w_i| \le w_{\max}$, $\|w\|_1 \le L$, and capacity $|\Delta N_i| \le \rho, \text{ADV}_i$.

We are fully free to choose any reward model, add any model architecture, as long as it does comply to action space topology. Training uses A2C/PPO/SAC with walk-forward splits, multi-seed stability checks, and block-bootstrap confidence intervals on out-of-sample performance. Guardrails—projection, vol targeting, drawdown governors, kill-switches—are non-negotiable.

Portfolio management:

RL vs Statistical modeling

The statistical+AI route excels when governance and auditability matter and when shrinkage-stabilized $\hat{\Sigma}_t$ plus modest predictive signals are available; it adapts by re-estimation and regime detection. Pure RL is attractive in highly non-stationary, friction-laden settings where the state-to-allocation mapping is complex; it internalizes costs but requires stricter safety wrappers and richer data. In practice, we combine them: a statistical+AI core portfolio for robustness, with tactical capital allocated to RL policies that prove stable under walk-forward testing and stringent risk caps.



Project architecture: