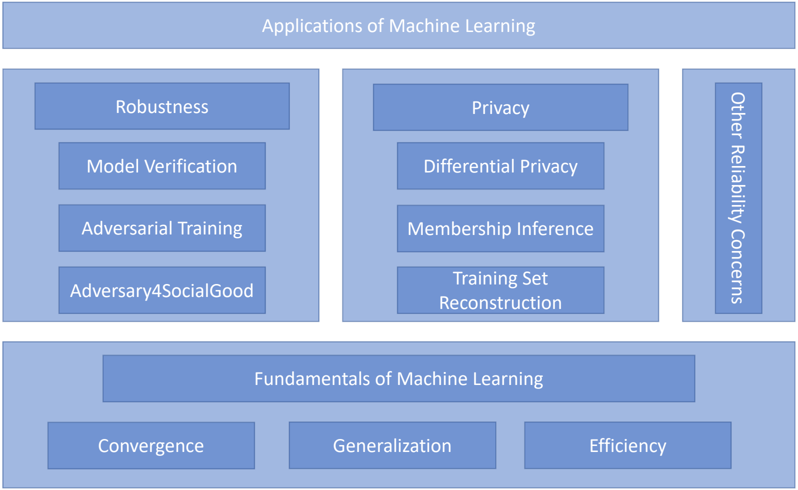
Harnessing Large Language Models in Financial Technologies

Chen Liu

Department of Computer Science
City University of Hong Kong

September 13, 2025
@ EPFL Alumni Great Bay Area Gathering, Shenzhen, China.

Overview of My Research



Sparse Portfolio Optimization

Problem definition

- Select at most m assets from n candidates to maximize investment performance.

Sparse Portfolio Optimization

Problem definition

- Select at most m assets from n candidates to maximize investment performance.

Why sparsity matters?

- better interpretability of selected assets.
- lower transaction costs and easier implementation.

Sparse Portfolio Optimization

Problem definition

- Select at most m assets from n candidates to maximize investment performance.

Why sparsity matters?

- better interpretability of selected assets.
- lower transaction costs and easier implementation.

Evaluation metrics

- Cumulative Wealth (CW): total portfolio return over the horizon.
- Sharpe Ratio (SR): risk-adjusted return per unit volatility.
- Maximum Drawdown (MD): worst-case loss from peak to trough.

Alpha Factor

Definition

- An alpha factor is a mathematical expression that maps historical features (e.g. price, volume, volatility) to a score for each asset.
- Higher alpha factor score $\rightarrow$ More attractive asset under the investment objective.

Evaluation

- The correlation between the ranking by alpha factors and the real ranking.

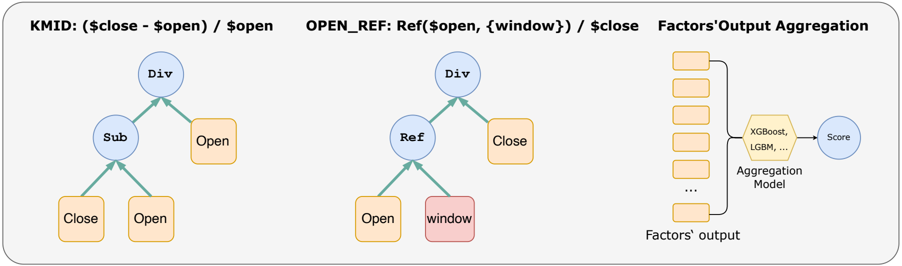


Figure: Example of alpha factors with their tree-structures in Alpha158 (Left) and how multiple factors' outputs are aggregated using models such as XGBoost or LightGBM to produce a final score (Right).

Alpha Factor

Traditional Alpha Factor Pool

Alpha#1: (rank(Ts_ArgMax(SignedPower(((returns < 0) ? stddev(returns, 20) : close), 2.), 5)) - 0.5)

Alpha#2: (-1 * correlation(rank(delta(log(volume), 2)), rank(((close - open) / open)), 6))

Alpha#3: (-1 * correlation(rank(open), rank(volume), 10))

Alpha#4: (-1 * Ts_Rank(rank(low), 9))

Alpha#5: (rank((open - (sum(vwap, 10) / 10))) * (-1 * abs(rank((close - vwap)))))

Alpha#6: (-1 * correlation(open, volume, 10))

Alpha#7: ((adv20 < volume) ? ((-1 * ts_rank(abs(delta(close, 7)), 60)) * sign(delta(close, 7))) : (-1 * 1))

Alpha Factor

Traditional Alpha Factor Pool

<u>Alpha#74:</u> ((rank(correlation(close, sum(adv30, 37.4843), 15.1365)) < rank(correlation(rank(((high * 0.0261661) + (vwap * (1 - 0.0261661)))), rank(volume), 11.4791))) * -1)

<u>Alpha#75:</u> (rank(correlation(vwap, volume, 4.24304)) < rank(correlation(rank(low), rank(adv50), 12.4413)))

<u>Alpha#76:</u> (max(rank(decay_linear(delta(vwap, 1.24383), 11.8259)), Ts_Rank(decay_linear(Ts_Rank(correlation(IndNeutralize(low, IndClass.sector), adv81, 8.14941), 19.569), 17.1543), 19.383)) * -1)

<u>Alpha#77:</u> min(rank(decay_linear(((((high + low) / 2) + high) - (vwap + high)), 20.0451)), rank(decay_linear(correlation(((high + low) / 2), adv40, 3.1614), 5.64125)))

<u>Alpha#78:</u> (rank(correlation(sum(((low * 0.352233) + (vwap * (1 - 0.352233))), 19.7428), sum(adv40, 19.7428), 6.83313))^rank(correlation(rank(vwap), rank(volume), 5.77492)))

<u>Alpha#79:</u> (rank(delta(IndNeutralize(((close * 0.60733) + (open * (1 - 0.60733))), IndClass.sector), 1.23438)) < rank(correlation(Ts_Rank(vwap, 3.60973), Ts_Rank(adv150, 9.18637), 14.6644)))

Limitations of Current Methods

Factor-based Methods

- Interpretable but require heavy manual design.
- Degrade quickly in live trading.
- Sparse decay: performance drops sharply when selecting only few assets.

Traditional Optimization Methods

- Competitive but uninterpretable results.
- Computationally expensive and sensitive to hyper-parameters.

Limitations of Current Methods

Factor-based Methods

- Interpretable but require heavy manual design.
- Degrade quickly in live trading.
- Sparse decay: performance drops sharply when selecting only few assets.

Traditional Optimization Methods

- Competitive but uninterpretable results.
- Computationally expensive and sensitive to hyper-parameters.

Let's design an **adaptive, interpretable and robust** algorithm to find competitive alphas efficiently.

Overall Framework

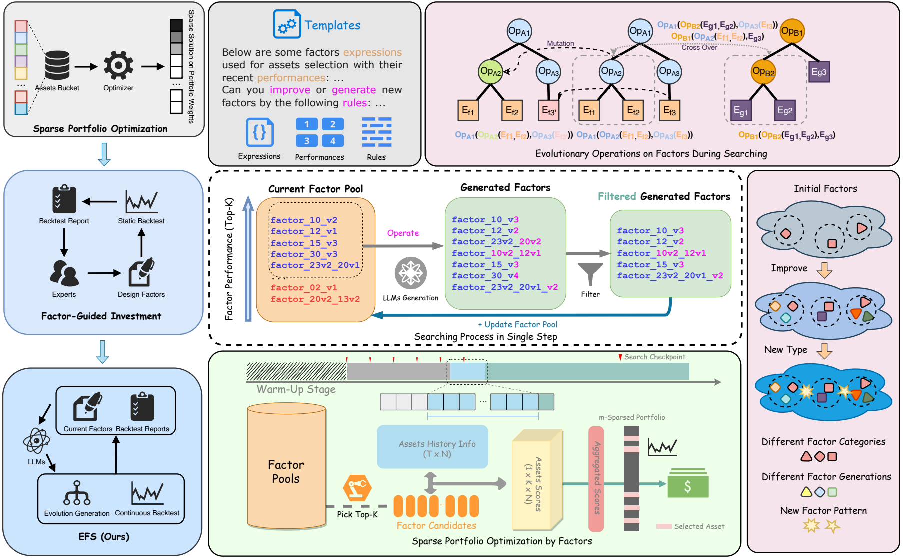


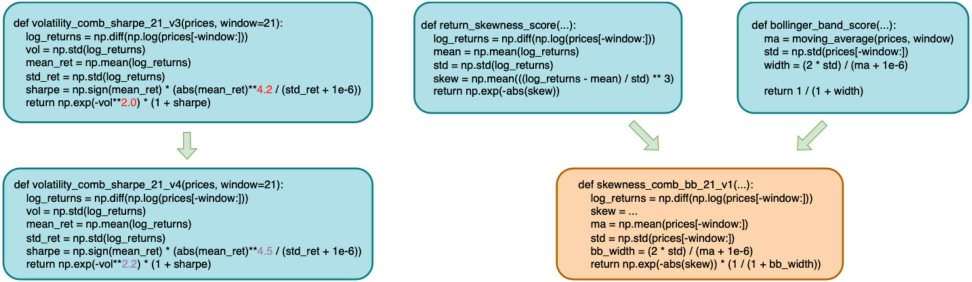
Figure: Evolutionary Factor Search (EFS) framework.

Prompt to Improve Alpha Factor

```
You are a world-class quantitative researcher and Python programmer specializing
in alpha factor design for asset ranking.
Your task is to generate high-quality Python factor functions that are evolved
versions of provided factors.
STRICT REQUIREMENTS:
1. Output ONLY a Python list of function strings - no comments or explanations
2. Each function MUST:
  - Be bug-free and executable
  - Maintain identical input signature: prices, window   - Use only numpy (as np),
don't depend on any external function or variable, you need to do computation all
inside function
  - Handle edge cases (short series, NaNs)
  - Clearly indicate if combining or modifying existing factors
3. Absolute prohibitions:
  - No external functions
  - No hardcoded values that should be parameters
  - No pandas or other libraries
  - No comments in output code
4. Factor name rules: [factor_name_part]_[window_size]_v[version number], the
window_size can only be the following value: 3, 7, 14, 21
5. Value of output factor: For factors, higher value means better asset, please
make sure the output value is positive related to performance of assets.
ACTION SPACE:
1. Improve existing factors by mutation:
  - Modifying parameters (e.g., inner parameters)
  - Adjusting logic
  - Updating inside operators for factors
2. Improve existing factors by crossover:
  - Combining two existing factors to create a new one if you think they can work
together
  - Restart version number from v1 for new factors
IMPROVEMENT CRITERIA:
1. Version increments must show clear:
  - if you improve from a given version, increase 1 to version number, the
```

Figure: Prompt fed to LLMs in the evolution of alpha factors.

Evolution of Alpha Factor



(a) Single Factor Improvement

(b) Factor Crossover

Figure: How alpha factor evolves in LLM-enabled search.

Interpret Alpha Factors

```
def complex_factor_1(prices, window=14):
    import numpy as np
    a = np.asarray(prices)
    if len(a)<max(3,window) or np.any(np.isnan(a[-window:])):
        return 0.0
    w = min(window,3)

    weights = np.exp(np.linspace(-1.5,0.2,w))
    weights /= weights.sum()
    ema = np.dot(a[-w:],weights)
    rel = (a[-1] - ema)/(np.abs(ema)+1e-6)

    ma = np.mean(a[-window:])
    std = np.std(a[-window:])
    width = (2*std)/(ma+1e-6)
    bb_score = 1/(1+width)

    return np.tanh(np.abs(rel)) * bb_score
```

Handle Invalid Input

Capture Movement Signal

Capture Risk Signal

What this factor Capture?

1. Significant Short-Term Price Deviation (Momentum Component). Tanh norm converts deviations to [0,1) range:
Values near 0 → Price hovering near EMA (no momentum)
Values approaching 1 → Strong directional breakout

2. Low Overall Volatility (Stability Component)
Calculates standard Bollinger Band width Inverts volatility to create stability score:
1 = Extremely narrow bands (high stability)
0 = Extremely wide bands (high volatility)

The combined signal suggests:
✔ High-Probability Breakouts
When both conditions align (strong momentum + low volatility), the signal identifies:
(1) Early trend initiation in calm markets
(2) High-potential continuation patterns
(3) Reliable support/resistance breaks
✖ Avoids False Signals
(1) Breakouts during high volatility
(2) Small price movements in stable conditions

```
def complex_factor_2(prices, window=7):
    if len(prices) < window+2: return 0.0
    arr=np.array(prices[-(window+2):])
    if np.isnan(arr).any(): return 0.0
    logrets=np.diff(np.log(arr))

    mean_ret=np.mean(logrets[-window:])
    momentum=(arr[-1]/(arr[-window-1]+1e-8))-1
    mean_abs=np.mean(np.abs(logrets))
    std=np.std(logrets)+1e-8
    v=(mean_abs/(std+1e-8))*np.abs(mean_ret+1.05)

    return mean_ret*momentum*v
```

What This Factor Captures?

Individual Components Effect

7-Day Mean Return - Identifies the prevailing trend direction (positive=uptrend, negative=downtrend)
7-Day Momentum - Measures recent price acceleration strength
Volatility Adjustment - Assesses trend quality by filtering out noisy movements

Combined Effect:
✔ Spots high-probability trends with: Clear direction (Mean Return); Strong momentum (Momentum); Low noise (Volatility Adjustment)

✖ Automatically filters: Choppy, directionless markets; High-volatility false breakouts; Weak, unconvincing trends

Figure: Use LLMs to interpret alpha factors.

Overall Framework

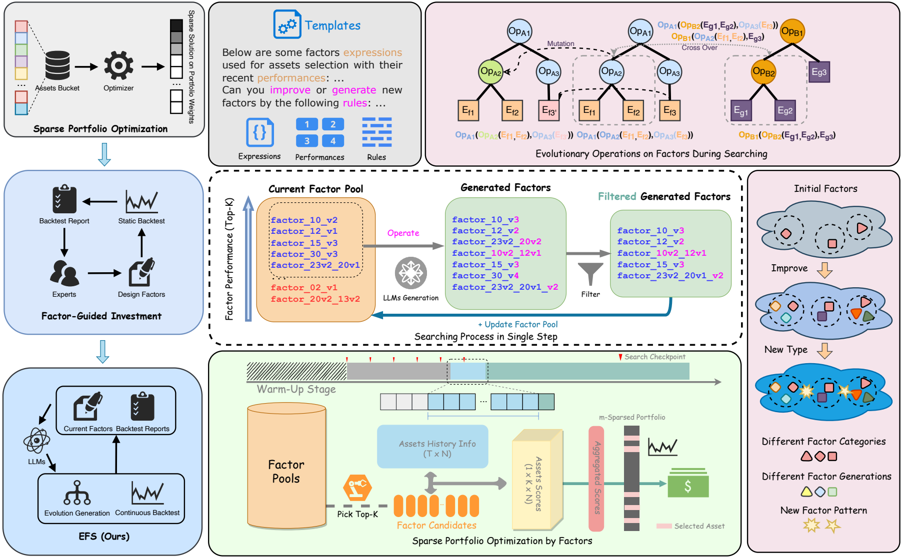


Figure: Evolutionary Factor Search (EFS) framework.

Evolutionary Factor Search (EFS)

Key Features of EFS:

- **LLM-driven factor generation:** prompt LLMs to create *executable* alpha factor formulas.
- **Evolutionary Search:** refine factors iteratively via *mutation* and *crossover*, guided by backtest results.
- **Closed-Loop Feedback:** use performance metrics (CW, SR) to update and prune the factor pool.
- **Sparse Portfolio Construction:** select top-m asset to construct portfolio.
- **Transparency:** factors are human-readable, interpretability and directly deployable.

Results on Real Market

Table: Evaluation of Cumulative Wealth (CW↑), Sharpe Ratio (SR↑), and Maximum Drawdown (MDD↓) on real-market datasets (US50, HSI45 and CSI300) for different model variants. The time frame is from 2019 to 2024 for US50, from 2022 to 2025 for HSI45 and CSI300.

		US50			HSI45			CSI300		
Group	Method	CW↑	SR↑	MDD↓	CW↑	SR↑	MDD↓	CW↑	SR↑	MDD↓
Baseline	1/N	4.562	0.072	0.344	1.333	0.029	0.409	1.087	0.014	0.214
	Min-cVaR	1.779	0.038	0.314	1.628	0.063	0.244	0.992	0.003	0.286
	Max-Sharpe	4.495	0.061	0.461	1.428	0.043	0.300	1.008	0.007	0.333
m=10	LGBM	4.182	0.063	0.332	1.611	0.038	0.367	2.334	0.072	0.225
	XGBoost	6.313	0.077	0.328	1.581	0.035	0.440	1.420	0.032	0.345
	mSSRM-PGA	5.121	0.059	0.569	0.766	-0.003	0.547	0.881	0.002	0.399
	ASMCVaR	10.259	0.073	0.582	2.481	0.052	0.453	1.453	0.030	0.462
	EFS-DeepSeek	**25.101**	0.132	0.288	**3.463**	0.080	0.385	3.437	0.079	0.327
	EFS-GPT	22.905	0.130	0.260	2.789	0.067	0.292	**4.962**	0.098	0.301
m=15	LGBM	3.899	0.062	0.328	1.588	0.037	0.387	1.812	0.055	0.250
	XGBoost	5.607	0.076	0.319	1.586	0.036	0.420	1.348	0.029	0.344
	mSSRM-PGA	4.976	0.062	0.477	0.766	-0.003	0.547	0.787	-0.010	0.384
	ASMCVaR	11.124	0.074	0.566	**2.647**	0.054	0.434	1.658	0.035	0.424
	EFS-DeepSeek	13.978	0.114	0.298	2.364	0.061	0.406	2.510	0.067	0.298
	EFS-GPT	**14.707**	0.117	0.278	2.277	0.058	0.307	**3.218**	0.082	0.246

Ablation Studies

Table: Overall real-market portfolio performance metrics (CW = Cumulative Wealth, SR = Sharpe Ratio, MDD = Maximum Drawdown, RankIC = Rank Information Coefficient, RankICIR = Rank Information Coefficient Information Ratio). The time frame is from 2019 to 2024 for US50, from 2022 to 2025 for HSI45.

	US50					HSI45				
Method	CW↑	SR↑	MDD↓	RankIC↑	RankICIR↑	CW↑	SR↑	MDD↓	RankIC↑	RankICIR↑
Initial Factor	6.254	0.081	0.449	0.005	0.352	1.364	0.031	0.386	0.018	0.950
EFS-DeepSeek	32.993±6.044	0.149±0.003	0.260±0.013	0.027±0.001	1.582±0.050	3.193±0.923	0.076±0.015	0.387±0.026	0.022±0.001	1.412±0.103
w/o Sparse Heuristic	19.248±3.642	0.125±0.003	0.346±0.048	0.020±0.004	1.262±0.265	2.198±1.094	0.051±0.030	0.391±0.051	0.027±0.015	1.508±0.828
w/o Numeric	23.414±3.814	0.133±0.007	0.324±0.042	0.023±0.004	1.334±0.201	2.943±0.516	0.072±0.007	0.372±0.024	0.024±0.002	1.435±0.118
w/o Quality	24.152±7.988	0.133±0.016	0.297±0.052	0.021±0.003	1.178±0.202	2.402±0.404	0.060±0.010	0.429±0.023	0.016±0.003	1.011±0.200
w/o Performance	9.549±5.869	0.094±0.019	0.327±0.053	0.009±0.009	0.487±0.467	1.168±0.036	0.020±0.002	0.369±0.025	0.009±0.002	0.522±0.106
w/o TA Factors	5.367±1.652	0.074±0.011	0.394±0.043	0.002±0.004	0.117±0.241	1.875±1.354	0.037±0.037	0.359±0.062	0.008±0.016	0.494±0.940

Portfolio Performance

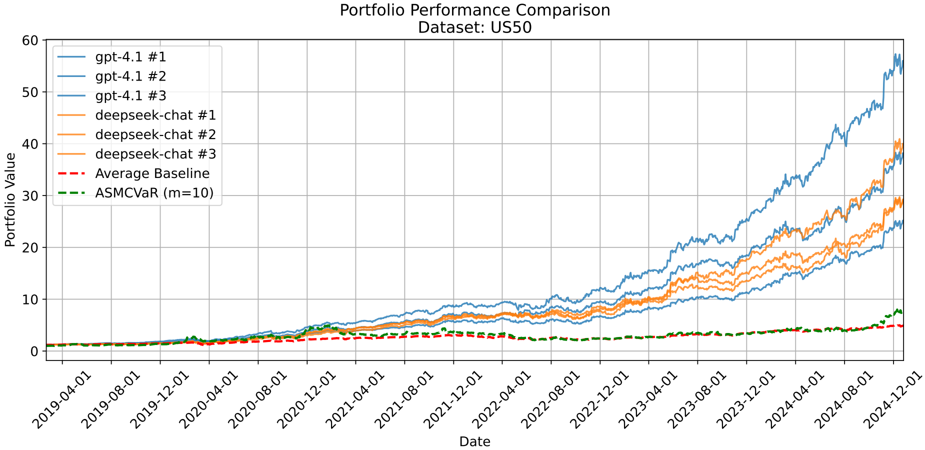


Figure: Portfolio performance comparison across US50, HSI45, and CSI300 datasets. Each plot shows the evolution of LLM-generated portfolios versus baselines and the ASMCVaR benchmark over time.

Portfolio Performance

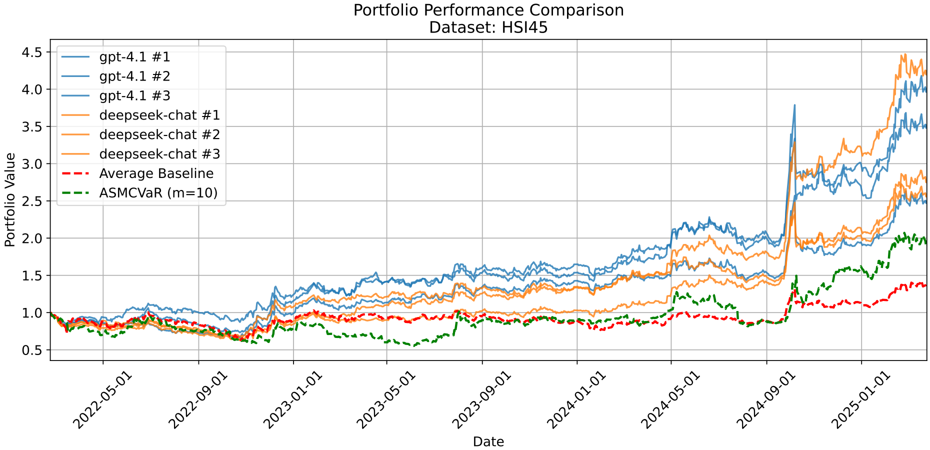


Figure: Portfolio performance comparison across US50, HSI45, and CSI300 datasets. Each plot shows the evolution of LLM-generated portfolios versus baselines and the ASMCVaR benchmark over time.

Portfolio Performance

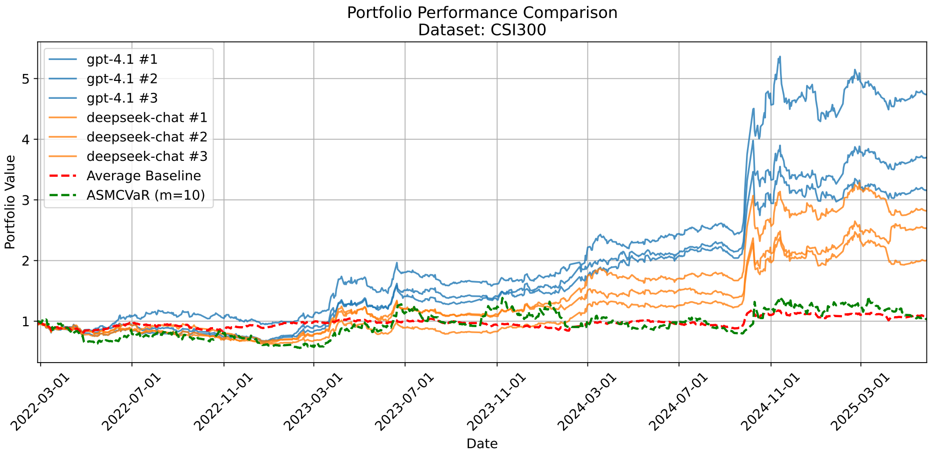


Figure: Portfolio performance comparison across US50, HSI45, and CSI300 datasets. Each plot shows the evolution of LLM-generated portfolios versus baselines and the ASMCVaR benchmark over time.

Portfolio Analysis

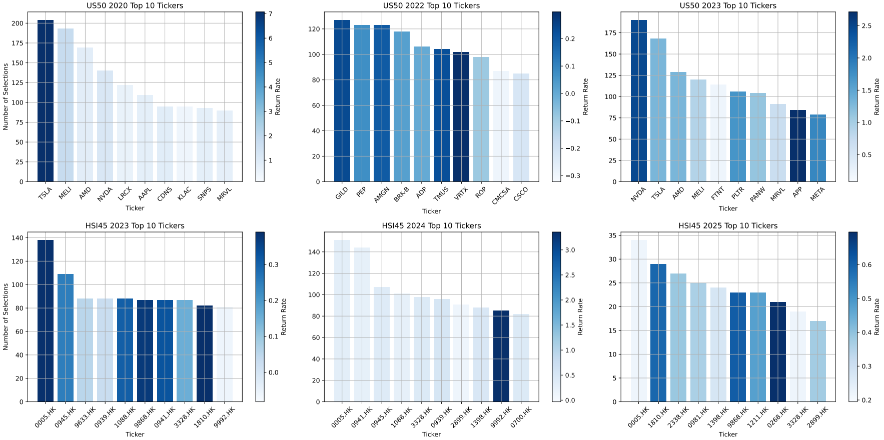
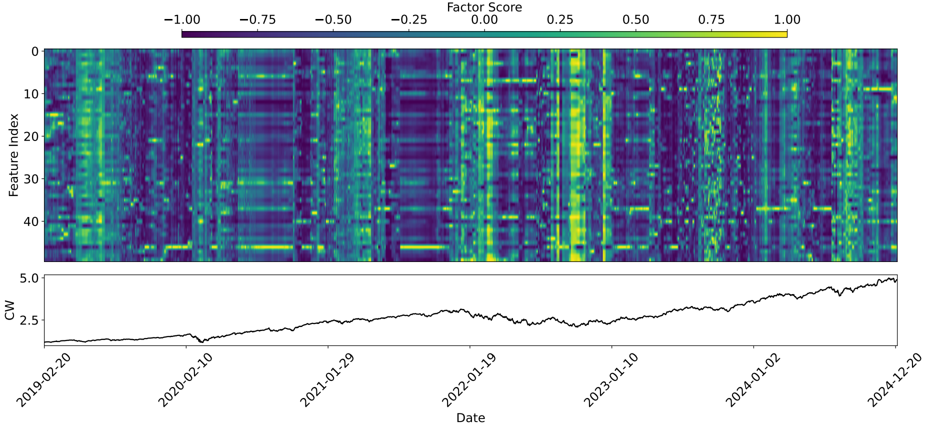


Figure: Annual snapshots of the top 10 most frequently selected assets under EFS in representative years for US50 (top row: 2020, 2022, 2023) and HSI45 (bottom row: 2023, 2024, 2025). Bar height denotes the number of selections within the given year, while the color encodes the annual return rate of each asset. This illustrates EFS's dynamic asset preference and adaptability to varying market environments.

Portfolio Analysis

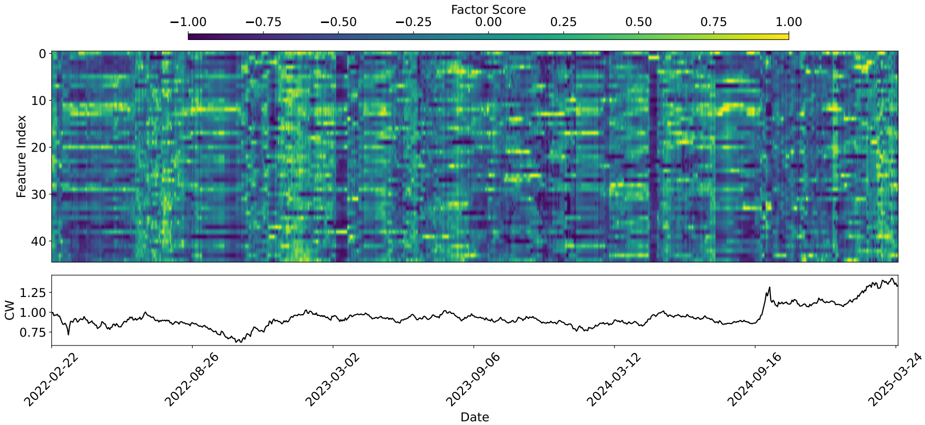


(a) US50 dataset

Figure: Factor score heatmaps and corresponding baseline curves across three datasets.

In general, portfolio holdings are more consistent in bull market.

Portfolio Analysis

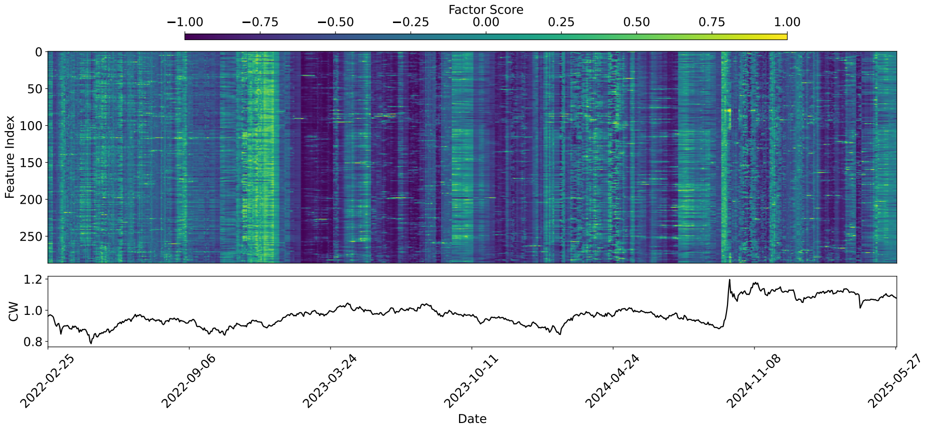


(a) HSI45 dataset

Figure: Factor score heatmaps and corresponding baseline curves across three datasets.

In general, portfolio holdings are more consistent in bull market.

Portfolio Analysis



(a) CSI300 dataset

Figure: Factor score heatmaps and corresponding baseline curves across three datasets.

In general, portfolio holdings are more consistent in bull market.

Transaction Fees

Table: Backtest results under transaction costs $c = 0.1\%$ and $c = 0.2\%$. Metrics shown are Cumulative Wealth (CW), Sharpe Ratio (SR), and Maximum Drawdown (MDD) across datasets US50, HSI45, and CSI300.

c	Method	US50			HSI45			CSI300		
		CW	SR	MDD	CW	SR	MDD	CW	SR	MDD
-	EFS-GPT 4.1	39.746±15.484	0.154±0.013	0.252±0.021	3.338±0.770	0.076±0.012	0.324±0.041	3.862±0.802	0.086±0.011	0.290±0.079
	EFS-DeepSeek	32.709±6.244	0.149±0.003	0.261±0.014	3.203±0.906	0.076±0.015	0.385±0.024	2.451±0.412	0.060±0.010	0.356±0.022
0.1%	EFS-GPT 4.1	25.293±10.802	0.135±0.016	0.256±0.021	2.710±0.571	0.065±0.011	0.340±0.049	2.624±0.425	0.064±0.008	0.354±0.093
	EFS-DeepSeek	22.058±4.053	0.133±0.003	0.265±0.015	2.643±0.775	0.065±0.015	0.410±0.023	1.668±0.227	0.039±0.008	0.434±0.012
0.2%	EFS-GPT 4.1	16.114±7.464	0.117±0.018	0.260±0.022	2.201±0.422	0.054±0.011	0.358±0.056	1.785±0.208	0.043±0.006	0.412±0.103
	EFS-DeepSeek	14.876±2.646	0.117±0.003	0.270±0.016	2.182±0.665	0.054±0.016	0.433±0.022	1.136±0.130	0.018±0.007	0.502±0.009

Different LLMs

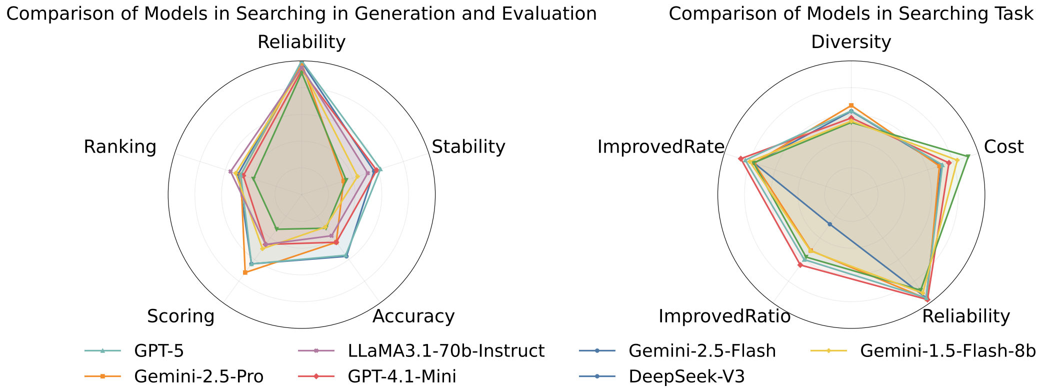


Figure: Radar chart comparison of model performance in alpha factor searching. (Left) Models evaluated on generation and evaluation tasks across reliability, stability, accuracy, scoring, and ranking dimensions. (Right) Models compared on search efficiency metrics including diversity, cost, reliability, improvement rate, and improvement ratio.

Summary

Contributions:

- We propose a unified evolutionary factor search (EFS) framework.
- We adaptively adjust alpha factors in different market regimes.
- The alpha factors found demonstrate good interpretability and competitive performance.

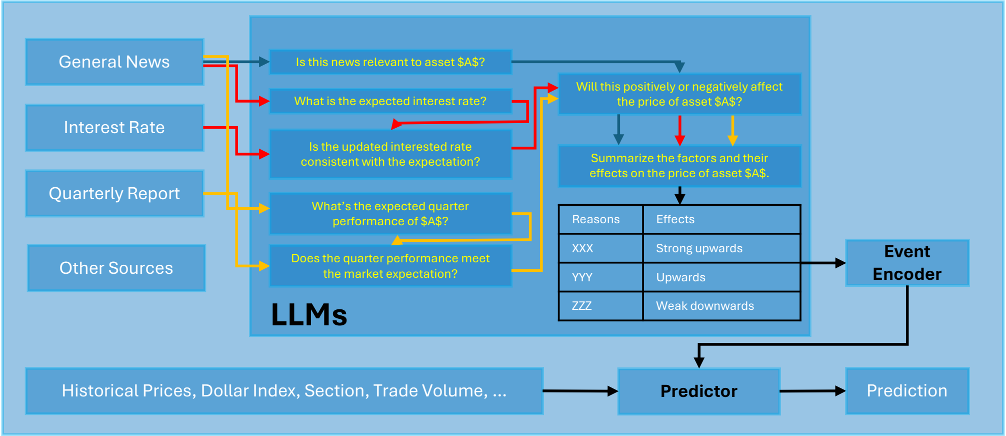
Current Limitations:

- Backtesting is computationally expensive.
- Sensitivity to prompt design and hyper-parameters.
- Limited coverage of multimodal signals (e.g. news, macro events).

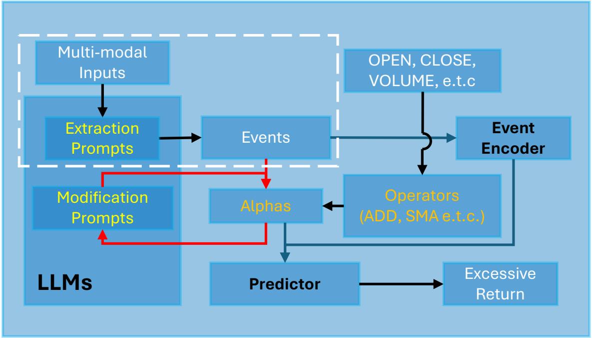
Future Works

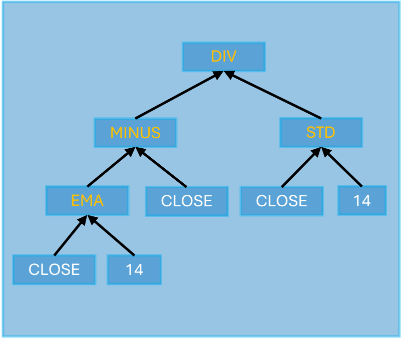
- Multimodal factor mining using news and alternative textual data.
- Enhance prompt robustness with automatic filtering and tuning.

Future Development



Future Development





Acknowledgements

- Prof. Yuan Zhang, Shanghai University of Finance and Economics.
- My research postgraduate students:
  - Haochen Luo
  - Jiandong Chen
- High school summer interns:
  - Ho Tin Ko
  - David Sun

Thank You!