人工知能学会第二種研究会資料
金融情報学研究会
SIG-FIN-036-31

大規模言語モデルを用いた株式投資戦略の自動生成におけるフィードバック設計

Feedback Design for the Automatic Generation of Stock Investment Strategies

河村 飛来 [1,4], *　久保 健治 [2,4], 中川 慧 [3,4],
Hirai KAWAMURA [1,4], Kenji KUBO [2,4], Kei NAKAGAWA [3,4],

[1] 東京大学 医学部
[1] Faculty of Medicine, The University of Tokyo
[2] 東京大学 工学系研究科
[2] Graduate School of Engineering, The University of Tokyo
[3] 大阪公立大学 経営学研究科
[3] Graduate School of Business, Osaka Metropolitan University
[4] 株式会社松尾研究所
[4] Matsuo Institute, Inc.

Abstract:
As large language models (LLMs) have become more powerful, their use in investment strategies has grown rapidly. However, we still lack a sufficient empirical understanding of how effective LLMs are in the process of improving investment strategies. In particular, there has been no systematic investigation of how feedback should be presented to LLMs to improve investment strategies. Motivated by this question, this study constructs an automated framework for generating stock investment strategies using LLMs and empirically examines the impact of feedback design on strategy improvement. Specifically, we conducted iterative strategy improvement experiments under multiple conditions based on two dimensions: the scope of information provided (basic information only vs. basic information plus additional information) and the format of presentation (text only vs. text plus plots). The experimental results show that differences in feedback design exert a modest influence on the improvement process, but their impact on performance gains is limited. In contrast, differences in the models used led to larger variations in performance improvement. These findings suggest that the success of strategy improvement may depend more strongly on model-specific characteristics than on the fine details of feedback design.

1　はじめに

大規模言語モデル（Large Language Models：LLMs）は，自然言語処理における汎用的な推論，生成能力を背景に，金融分野においてもデータ分析，トレーディング，文章作成といった多様な業務へ適用範囲を拡げている [1, 2]．例えば，金融ニュースを対象としたLLMベースのセンチメント分析が，銘柄の翌日リターンと統計的に関連することが報告されている [3]．また，金融ニュースをインプットに運用コメントを自動生成する研究もある [4]．

この点，LLMを分析あるいは生成器ではなく意思決定する主体として位置づけ，意思決定プロセスそのものをエージェント化する枠組みも提案されている [5, 1, 6]．また，投資戦略生成の観点では，LLMは非構造データの解釈だけでなく，定性的仮説を定量的なシグナル（アルファ）へ変換し，探索するエージェントとして用いられている [7, 8]．

しかし，戦略のパフォーマンスの定量指標や可視化結果を与えた時，LLMがどのようなフィードバックを生成し，戦略の改善に繋げるかというフィードバック

*連絡先：河村 飛来，東京大学
〒 113-8654 東京都文京区本郷７丁目３－１
E-mail: kawamura-hirai323@g.ecc.u-tokyo.ac.jp
本稿の内容は筆者らが所属する組織を代表するものではなく，本稿の全ての誤りは，筆者らの責に属するものである．

能力そのものを評価する研究は限定的である．実務的な定量運用の現場では，戦略改善は重要なタスクではあるが，既存研究 [7, 8] や文献 [9, 10] は主としてアルファ生成やファクターの構築に焦点を当てることが多く，フィードバック能力を系統的に評価する余地が残る．また，[1] においても，戦略の頑健性，金融市場の複雑なダイナミクスへの適応という要件を重要課題として挙げており，フィードバックによる戦略改善はこれらと関連する要素である．

本研究では，LLM を用いた株式投資戦略の自動生成フレームワークを構築し，フィードバック設計が戦略改善に与える影響を実証する．具体的には，提示する情報の範囲（基本情報のみまたは基本情報と追加情報）および提示形式（テキストのみ又はテキストとプロット）という 2 軸に基づき，複数の条件下で戦略改善タスクを反復的に実行した．実験の結果，フィードバック設計の差異は改善プロセスに一定の影響を与えるものの，パフォーマンスの改善に対する効果は限定的であった．一方で，使用するモデルの違いは，パフォーマンスの改善においてより大きな差異をもたらした．この結果は，戦略改善の成否がフィードバックの細かな設計よりも，モデル固有の特性に強く依存する可能性を示唆する．

2 問題設定

本研究では，投資戦略の自動生成プロセスにおけるフィードバック設計の違いが，戦略改善プロセスにどのような影響を与えるかを検証する．特に，(i) どのモデルを用いるかというモデル選択，(ii) どの情報を与えるかという情報の範囲，(iii) どの形式で与えるかという提示形式，の 3 点に着目する．

モデル選択に関しては，モデルのパラメータ規模や学習データ，推論特性の違いが，フィードバックの解釈能力や改善方針の策定に影響を与える可能性がある．

また，提示する情報の範囲も重要であり，リターンやシャープレシオといった基本的なパフォーマンス指標のみでは，戦略がそのパフォーマンスを生み出している要因を構造的に把握することは難しい．つまり，観測されたパフォーマンスが持続的な予測力に基づくものなのか，あるいは特定のリスク特性やポジションの偏りに起因するものなのかは，追加の指標がなければ十分に評価できない．したがって，追加情報の有無は，LLM が改善方針を表面的なパラメータ調整として捉えるのか，あるいは戦略の構造的修正として捉えるのかに影響を与える可能性がある．

さらに，バックテスト結果をテキストで要約して提示する場合，入力トークン数を抑制できる一方で，ある特定期間のパフォーマンスに対する統計量として情報が圧縮されるという課題がある．その結果，リターンやドローダウンの時間的推移，パフォーマンスの安定性や各種レジームへの依存性といった動的特性が十分に反映されない可能性がある．

これに対し，時系列データをテキストで提示すれば，動的情報を保持できるが，トークン数が大幅に増加し，コンテキスト長や計算コストの制約に直面する[1]この制約を緩和する手段として，時系列情報をプロットとして画像入力する方法が考えられる．画像であれば，どのようなタイムスパンであれ，時系列的な変動パターンを比較的コンパクトに提示でき，テキストによる詳細な記述に比べて効率的に動的情報を伝達できる可能性がある．

以上を踏まえ，本研究ではモデル選択に加えて，フィードバック設計を次の 2 軸で整理する．

- 情報の範囲：基本情報のみ／基本情報＋追加情報
- 提示形式：テキストのみ／テキスト＋プロット

ここで，各情報は以下のように定義する．

- 基本情報：リターン，ボラティリティ，シャープレシオ，最大ドローダウン，トータルコスト，生特徴量の統計量[2]
- 追加情報：IC（Infomation Coeficient：情報係数），ネットエクスポージャー，ファクターエクスポージャー

基本情報は戦略のパフォーマンスを示す指標であり，追加情報はシグナルの予測力やリスク構造を補足的に示す指標である．生特徴量の統計量を含めるのは，全ての特徴量が NaN や 0 である場合や，分布が著しく偏っている場合にそれをフィードバックすることを目的としている．

これら 2 軸の組み合わせにより，表 1 に示す 2 × 2=4 条件の設計を考え，そのうち、3 条件（P1–P3）を比較対象とする[3]．各条件は 3.2 節で定義するプロンプトに対応する．

それぞれの条件下で，以下の問題 1 を反復的に与え，戦略の改善プロセスを観察する．

問題 1. 初期戦略のバックテスト結果を分析し，実運用に耐えうる水準に達していないと判断される場合には，改善案を提示する．さらに，その改善案に基づいて戦略の実装（Python コード）を修正する．

以上の反復的試行を通じて，情報の範囲および提示形式の違いが，戦略改善のプロセスにどのような差異をもたらすかを比較・検証する．

[1]本研究で用いた 9 年分の日次データだと，Base64 でエンコードした画像のテキスト入力，時系列の数値列のテキスト入力，画像入力で 1 桁程度ずつトークン数のオーダーが違う．

[2]生特徴量とはポートフォリオウェイト算出前の特徴量のことを指す．

[3]「基本情報のみ」×「テキスト＋プロット」の条件に関しては検証を行なっていない．

表 1: フィードバックに含める情報

	テキスト	テキスト＋プロット
基本情報のみ	P1	—
基本＋追加情報	P2	P3

3 手法

本手法では，LLM とチャットする方式で戦略を改善していく．具体的には，初期提案生成，コード生成，フィードバック（改善提案）生成，エラー修正のそれぞれのプロンプトを用意し，LLM に対していずれかを与えることでチャットを成立させる．コード実行の成功回数が 10 回に到達する，あるいは LLM が「APPROVED」と出力した場合にループを打ち切る．

3.1 初期戦略の設定

各 LLM モデルの改善能力の比較を公平に行うため，事前にいくつかの初期戦略を生成し，全 LLM で共通して使用する．具体的には,「初期提案生成プロンプト，LLM 返答（改善案提案），コード生成プロンプト，LLM 返答（コード改善）」までの一連のやり取りをチャット履歴に事前に組み込んでおき，そこにフィードバック生成プロンプトを投げかけて以下の改善試行が始まる．初期戦略は全て，まず新規特徴量を生成し，それを元にポートフォリオウェイトを計算するという構成になっている．

3.2 フィードバック生成

第 2 章で導入したように，以下の 3 通りのプロンプトを試した．

Prompt 1 (P1) 基本的なバックテスト指標と生特徴量の各種統計量を含めたプロンプトである．

> Below are the backtest results of the strategy proposed above through 2014 to 2022. Based on the metrics, provide a comprehensive analysis and propose improvements. Do not write any code yet.
>
> The metrics include:
>
> **Backtest Metrics**: Total costs, Annualized return, Annualized volatility, Sharpe ratio, Max drawdown
> **Feature Statistics**: count, mean, std, min, 1%, 5%, 50%, 95%, 99%, max, skew, kurtosis, missing ratio, zero ratio
>
> If the strategy passes the criteria for production use, ONLY output "APPROVED".
> —
> ### Backtest Metrics
> {backtest_results}
> —
> ### Feature Statistics
> {feature_stats_text}

Prompt 2 (P2) P1 に加えて，クオリティ指標として日次 IC の平均・標準偏差と ICIR[4]を，リスク指標としてネットエクスポージャーの期間平均値とファクターエクスポージャーの累積値を含めたプロンプトである．

> Below are the backtest results of the strategy proposed above through 2014 to 2022. Based on the metrics, provide a comprehensive analysis and propose improvements. Do not write any code yet.
>
> The metrics include:
>
> **P&L Metrics**: Total costs, Annualized return, Annualized volatility, Sharpe ratio, Max drawdown
> **Net Exposure**: Average net exposure across time
> **Factor Exposure**
> - Relative factor exposure for following 17 style factors: BPR, EarningsYield, Size, MidCap, ShortTermMomentum, MidTermMomentum, LongTermMomentum, Beta, ResidualVolatility, EarningsQuality, EarningsVariability, InvestmentQuality, Leverage, Profitability, DividendYield, Growth, Liquidity
> **IC and ICIR**: Mean Daily IC, Daily IC Standard Deviation, ICIR
> **Feature Statistics**: count, mean, std, min, 1%, 5%, 50%, 95%, 99%, max, skew, kurtosis, missing ratio, zero ratio
>
> If the strategy passes the criteria for production use, ONLY output "APPROVED".
> —
> ### Backtest Metrics
> {backtest_results}
> —

[4]IC の平均を IC の分散で除した値として定義した．

Net Exposure
- Average net exposure across time: {net_exposure_mean:.6f}

—

Factor Exposure (Relative value)
{factor_exposure_text}

—

IC and ICIR
- Mean IC: {ic_mean:.6f}
- IC Standard Deviation: {ic_std:.6f}
- ICIR: {icir:.6f}

—

Feature Statistics
{feature_stats_text}

Prompt 3 (P3) P1 に加えて，累積リターン，ドローダウン，累積 IC，ネットエクスポージャー．累積ファクターエクスポージャーの時系列推移を示す画像データを含めたマルチモーダルなプロンプトである．

Below are the backtest results of the strategy proposed above. Based on the plots and metrics, provide a comprehensive analysis and propose improvements. Do not write any code yet.

The results include plots and metrics as follows:

1. Plots
Three figures are provided.
Figure 1: Equity curve (with and without transaction costs), Drawdown, Net exposure
Figure 2: Cumulative IC
Figure 3: Cumulative factor exposure

2. Metrics
Backtest Metrics: Total costs, Annualized return, Annualized volatility, Sharpe ratio, Max drawdown
Feature Statistics: count, mean, std, min, 1%, 5%, 50%, 95%, 99%, max, skew, kurtosis, missing ratio, zero ratio

If the strategy passes the criteria for production use, ONLY output "APPROVED".

—

Backtest Metrics
{backtest_results}

—

Feature Statistics
{feature_stats}

3.3 コード生成と実行

フィードバックに従って，特徴量とポートフォリオウェイトを計算する Python コードを生成する．具体的には，株価・出来高やその他データと各種パラメータを引数にとり，日付，証券番号，特徴量（`feature` 列），ポートフォリオウェイト（`weight` 列）を出力とする `compute_feature` 関数を生成する．

次に，生成したコードを実行し，特徴量およびポートフォリオウェイトの計算を行なう．コード実行時にエラーが生じた場合には，エラーメッセージを元にコードを修正し，再度コードを実行する．コード実行時には 10 分間のタイムアウトを設定し，タイムアウトした場合には通常のエラー対応と同様，エラー内容（タイムアウト）を提示した上でコードを修正し，再度実行する．

最後に，計算したポートフォリオのウェイトを元に，バックテストを実行する．この後は 3.2 節のフィードバック生成に戻り，一連の戦略改善プロセスが進行する．

4 実証分析

本章では，前章手法を金融セクター[5]を除く TOPIX 500 のデータを用いて分析する．

4.1 対象データおよび期間

株価・出来高，セクター情報，ファンダメンタルズ指標，空売り指標，マクロ指標などを含む 80 のデータを 2014 年から 2022 年まで日次で準備した．

4.2 手順

まず，LLM モデルの選定を行なった．表 2 は検証に用いた LLM である．本研究では 3 つの LLM ファミリー（GPT，Gemini，Claude）から合計 8 モデルを使用し，全てのモデルにおいてパラメータはデフォルト設定を用いた．

次に，初期戦略の生成を行った．同一ファミリーのモデルが作成した戦略の方が改善しやすいといったバイアスを抑えるため，各 LLM ファミリーから標準的と考えられる 1 モデルずつを選定し，合計 3 モデルで初期戦略を構築した．選定したモデルは GPT-5，Gemini

[5] 東証業種別株価指数 TOPIX-17 シリーズのうち，銀行，金融，不動産の構成銘柄に属するものとした．

表 2: 使用したモデル一覧

ファミリー	該当モデル
GPT	GPT-5 nano, GPT-5 mini, GPT-5
Gemini	Gemini 3 Flash Preview, Gemini 3 Pro Preview
Claude	Claude Haiku 4.5, Claude Sonnet 4.5, Claude Opus 4.5

3 Flash Preview，Claude Sonnet 4.5 である．初期戦略の生成における取引条件は以下のように設定した．

- 決済タイミングは翌営業日の寄りとする．
- 各日付においてユニバース内から生特徴量ベースで上位 5%以内をロング（買い），下位 5%以内をショート（売り）する．
- 取引コストとして片道 5bps（= 0.05%）を想定する．

各モデルが生成した初期戦略の概要は以下のとおりである．

GPT-5: FX Risk Underreaction (FXUR)　個別株リターンを TOPIX および為替変化率に対して最小二乗法により線形回帰をすることで理論リターンを推定し，実際のリターンとの差分を主要シグナルとする．さらに為替変化率や信用倍率で調整した後，サイズ，バリュー，モメンタム，ボラティリティ，クオリティの 5 ファクター[6]に対して中立化を行う．

Gemini 3 Flash Preview: Intraday Institutional Divergence (IID)　後場の出来高集中を伴うセクター相対リターンを捉える戦略であり，サイズ，バリュー，モメンタムの 3 ファクターに対して中立化を行う．

Claude Sonnet 4.5: Session Momentum Divergence Alpha (SMDA)　前場と後場におけるリターンおよび出来高の乖離をもとに，その加速および累積からシグナルを構築する戦略であり，サイズ，バリュー，モメンタム，ボラティリティ，クオリティの 5 ファクターに対して中立化を行う．

なお，これらの戦略の改善前のバックテスト結果は図 1 に Original として示してある．

この後，これらの初期戦略に対して，選定した 8 モデルすべてで 3 節の手法に則ってフィードバックおよび改善を行った．なお，モデルが所定のフォーマット以外で「APPROVED」と出力してきた場合には，そこで改善プロセス終了とみなした．

[6]ファクターとは，株式のリターンやリスクに影響を与える共通の要因のことであり，それらを中立化することで各銘柄固有の超過リターン（アルファ）を獲得することを目指す．

4.3　評価指標

フィードバックの情報が増えることによって，(i) パフォーマンスがどのように変わるのか，(ii) 実装が量的にどのように変わるのか，(iii) 実装が質的にどのように変わるのか，という点を今回の評価対象とする．1 点目はプロットと P&L の年率改善幅（%）で評価し，2 点目と 3 点目については，モデル・初期戦略ごとに実装コードを LLM に与え，その時系列変化を記述させることで評価した．特に，2 点目は以下に定義する実質的変更率を参考に評価した．この指標は，各モデルが戦略に対してどの程度意味のある変化を加えたかを測定することを目的としている．

$$\text{実質的変更率} = \frac{\text{実質的変更回数} + 0.5 \times \text{中程度変更回数}}{\text{総バージョン数}}$$

実質的変更率では，ウェイト計算への影響を考慮して変更の程度を以下のように定義した．

- 実質的：新機能の追加，アルゴリズムの変更，パラメータ値の変更，計算式の修正，戦略ロジックの変更
- 中程度：関数の分離統合，意味的に重要な変数名の変更，コード構造のリファクタリング
- 表面的（＝変更なし）：コメント，docstring の修正，フォーマット調整，軽微な変数名変更，空白行の調整

4.4　結果と考察

図 1 にプロンプト・初期戦略ごとの改善後の P&L 推移を，表 3 に P&L の年率改善幅（%）を示す．図 1 において，初期戦略は灰色，GPT は青系統，Gemini は緑系統，Claude は赤系統で示した．また，表 3 では年率改善幅の平均値順に列をソートしている．全体的に，Claude，Gemini，GPT の順番に改善後のパフォーマンスが良い傾向があり，同じファミリーのモデルが作った初期戦略の方が改善しやすいという傾向は見られない．

表 4 に，基本情報のみをテキストで提示する設定（P1）から，基本情報に追加情報を加えたテキスト提示（P2）への切替による P&L の年率改善幅の変化を示す．また，表 5 には，P1 から，追加情報に加えてプロットも提示する設定（P3）への切替による P&L の年率改善幅の変化を示す．全体の平均値を見ると，P1 から P2 への切替ではむしろマイナスに変化し，P1 から P3 への切替においても改善は確認されない．正負限らず大きな効果が観察されたモデルもあるが，モデル単位で見ると試行回数が十分とは言い難く，また今回は生成するテキストのランダムさを表す temperature をデフォルトの 1.0[7]に設定したため，試行を繰り返す

[7]0.0 から 2.0 までの範囲で設定されることが多く，0.0 の時には理論上毎回同じ出力となる．

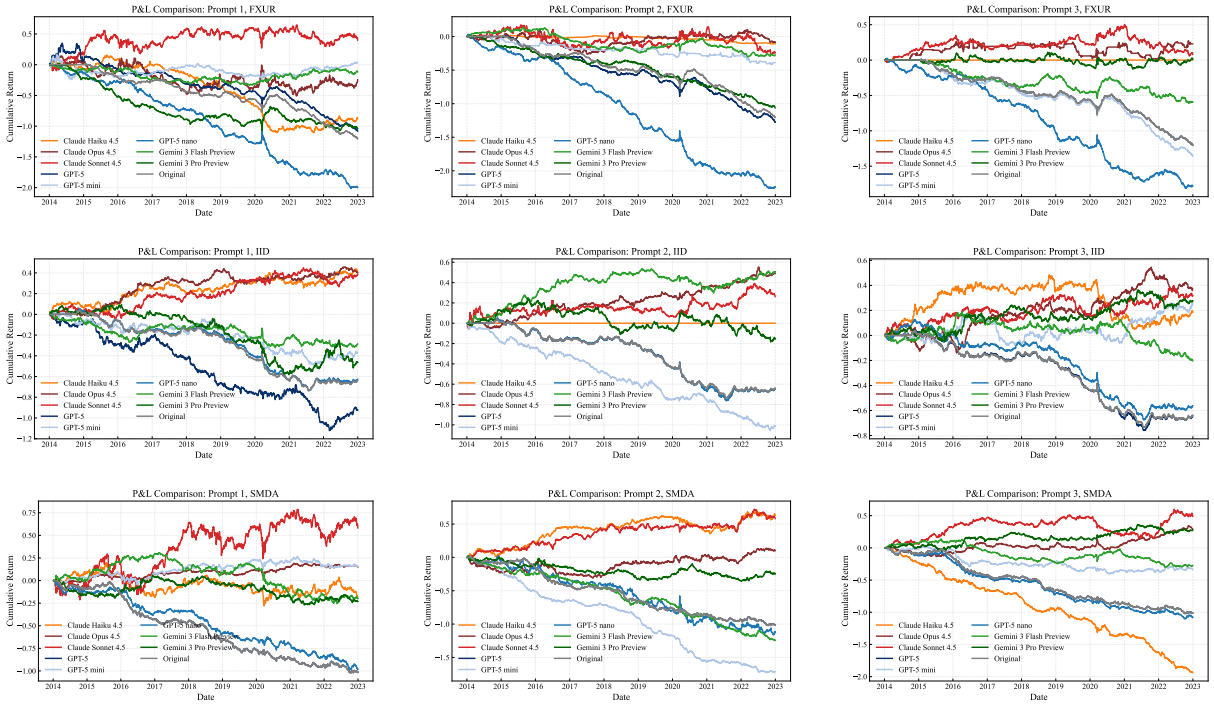

図 1: プロンプト・戦略ごとの改善後の P&L の推移
注：初期戦略は灰色，GPT は青系統，Gemini は緑系統，Claude は赤系統で示す．

ことで別の値に収束していく可能性もある．この点を解明するためには，temperature を 0.0 に設定するか，あるいは十分な回数の試行を繰り返す必要がある．

表 6 に，モデル×プロンプト別の実質的変更率（%）を示す．モデル内におけるプロンプト間の差分は一部で確認されるものの，それ以上にモデル間の差分が顕著であり，全体としては Gemini，Claude，GPT の順に値が低下する傾向が見られる．Claude および Gemini ではファミリー内で似た値を示す一方，GPT では同一ファミリー内においても GPT-5 mini，GPT-5 nano，GPT-5 の順に値が低下する傾向が確認された．Claude Haiku 4.5 では，P1・P2 と P3 の間で実質的変更率に特に大きな差が生じているが，これは P3 において収益性の低さを指摘し，改善を試みず同一コードの出力を継続したことに起因する．また，Gemini ファミリーはほぼすべての条件で 100%という極めて高い実質的変更率を示したが，これは戦略の改善というタスクの枠組み自体を逸脱し，戦略探索タスクへ移行した結果である．

プロンプトごとに実装された手法を見ると，P1 では古典的なファクターや手法を探索する傾向が見られ，P2 ではスタイル・ファクターに対する中立化に関連した実装が増加し，P3 では IC や VIX を用いた動的なゲーティングでレジーム変化に適応しようとする実装が観察された．この点は，プロンプトの差分に素直に対応していると言える．参考として，モデル×プロンプト別の実装手法の特徴と代表例を Appendix の表 A.1 に掲載しておく．

全体として見ると，P&L の改善幅やコードの実質的変更率に対してはモデル選択が大きな影響を持つ一方で，実装手法の内容に関してはプロンプトが大きな影響を持つことがわかる．また，実装手法の内容とパフォーマンスの改善幅には関連がないように思われる．

考察として，以下の 3 つのことが言える．

1. パフォーマンス改善幅の差異は，モデル固有の挙動特性に起因する．
2. コード変更の量は主としてモデル選択によって決定される．
3. コード変更の質は主としてフィードバック設計によって規定される．

まず 1 点目について，パフォーマンス差は単なる性能水準の優劣ではなく，挙動特性の違いに起因すると考えられる．今回，特に Claude 系モデルの成績が良好であり，GPT 系モデルは相対的に劣後した．Claude（および GPT-5 mini）は既存戦略の構造を保持しつつ，局所的なロジック修正やパラメータ調整を積み上げることで改善を図る傾向が見られた．これは探索空間を限定しながら，各点で勾配が大きいと推測される方向を何度か試して妥当な方向に進むプロセスであり，反復回数に対して安定的な改善率が期待できる．一方，Gemini は初期戦略と無関係な戦略の探索を行う傾向が

表 3: 初期戦略に対する P&L の年率改善幅（%）

	FXUR			IID			SMDA			
	P1	**P2**	**P3**	**P1**	**P2**	**P3**	**P1**	**P2**	**P3**	**平均**
Claude Sonnet 4.5	17.83	10.72	14.28	11.37	10.02	10.63	17.74	17.73	16.71	**14.12**
Claude Opus 4.5	10.65	12.46	15.94	11.59	12.54	11.22	12.95	12.39	14.52	**12.69**
Claude Haiku 4.5	3.67	12.21	13.39	11.61	7.10	9.18	9.23	18.27	-10.27	**8.27**
Gemini 3 Pro Preview	1.87	1.59	13.58	1.97	5.38	10.18	8.70	8.50	14.35	**7.35**
Gemini 3 Flash Preview	12.10	10.34	6.77	3.92	12.74	4.85	9.05	-2.56	8.23	**7.27**
GPT-5 mini	13.83	9.08	-1.70	3.04	-4.15	9.94	12.97	-7.79	7.54	**4.75**
GPT-5	1.42	-0.78	0.00	-3.15	-0.05	-0.05	0.00	0.00	0.00	**-0.29**
GPT-5 nano	-8.83	-11.55	-6.46	0.13	-0.08	0.86	0.34	-1.21	-0.74	**-3.06**

注 1：赤色は「APPROVED」となった戦略.
注 2：平均は全 9 条件の単純平均値．列は平均値の降順にソートしてある.

表 4: P1 から P2 への切替による P&L の年率改善幅の変化（%）

モデル	**FXUR**	**IID**	**SMDA**	**平均**
Claude Haiku 4.5	8.54	−4.50	9.03	4.36
Gemini 3 Pro Preview	−0.27	3.41	−0.20	0.98
Claude Opus 4.5	1.81	0.95	−0.56	0.73
GPT-5	−2.20	3.09	0.00	0.30
GPT-5 nano	−2.73	−0.21	−1.55	−1.50
Gemini 3 Flash Preview	−1.75	8.82	−11.62	−1.52
Claude Sonnet 4.5	−7.11	−1.36	−0.01	−2.83
GPT-5 mini	−4.75	−7.19	−20.76	−10.90
平均	−1.06	0.38	−3.21	**−1.30**

注 1：各セルは P&L の年率改善幅（%）について「P2 の値 − P1 の値」を計算したものを示す．正値は P2 が P1 を上回ることを意味する.
注 2：平均は単純平均値．列は平均値の降順にソートしてある.

表 5: P1 から P3 への切替による P&L の年率改善幅の変化（%）

モデル	**FXUR**	**IID**	**SMDA**	**平均**
Gemini 3 Pro Preview	11.71	8.21	5.65	8.52
Claude Opus 4.5	5.29	−0.37	1.57	2.16
GPT-5 nano	2.37	0.72	−1.09	0.67
GPT-5	−1.42	3.09	0.00	0.56
Gemini 3 Flash Preview	−5.33	0.94	−0.82	−1.74
Claude Sonnet 4.5	−3.55	−0.74	−1.03	−1.77
Claude Haiku 4.5	9.71	−2.43	−19.51	−4.08
GPT-5 mini	−15.52	6.90	−5.43	−4.68
平均	0.41	2.04	−2.58	**0.00**

注 1：各セルは P&L の年率改善幅（%）について「P3 の値 − P1 の値」を計算したものを示す．正値は P3 が P1 を上回ることを意味する.
注 2：平均は単純平均値．列は平均値の降順にソートしてある.

表 6: モデル×プロンプト別の実質的変更率（%）

モデル	P1	P2	P3	平均
G3 Pro	100.0%	100.0%	100.0%	**100.0%**
G3 Flash	100.0%	100.0%	100.0%	**100.0%**
Opus	82.6%	86.5%	90.5%	**86.5%**
GPT-5m	87.0%	81.5%	79.6%	**82.7%**
Sonnet	78.0%	79.5%	73.5%	**77.0%**
Haiku	83.3%	80.0%	50.0%	**71.1%**
GPT-5n	51.9%	31.5%	24.1%	**35.8%**
GPT-5	22.2%	9.3%	24.1%	**18.5%**
平均	**75.6%**	**71.0%**	**67.7%**	**71.5%**

注 1：赤色はモデル内の最大値，青色はモデル内の最小値．
注 2：平均は単純平均値．列は平均値の降順にソートしてある．
注 3：変更の程度の分類が曖昧なものもあり，各数値は評価者により変動しうる．

あり，戦略改善タスクからの逸脱も観察された．これは局所最適からの脱出という観点では有利であり得るが，短期的にはパフォーマンスの分散を高める．すなわち，改善幅の期待値は高いが分散も大きく，十分な反復回数を確保できない状況では Claude に劣後する可能性がある．GPT（GPT-5 および GPT-5 nano）は既存ロジックを大きく変更しない保守的傾向が顕著であり，結果として Claude および Gemini の双方に劣る成績となった．

次に 2 点目について，本検証ではモデルごとに実質的変更率に明確な差が見られた．すなわち，大胆に構造を変更するモデルと，既存ロジックを維持するモデルとが存在した．全体として Gemini は実質的変更率が高く，GPT は実質的変更率が低かったが，このような傾向は，GPT は文脈拘束が強い一方，Gemini は文脈拘束が弱いという直感的印象と整合する結果であった．これは，例えば RLHF[8]などを通して獲得される，どれほど文脈整合性を重要視するかといったモデル固有の性質などが，コード変更の程度を規定した可能性を示唆する．

最後に 3 点目について，本検証ではプロンプトごとに実装される手法に違いが見られた．ファクターエクスポージャーを与えた場合（特に P2）に中立化に関する実装が増加する傾向や，時系列プロットを与えた場合（P3）にレジーム適応のための実装が増加する傾向が見られ，与える情報がフィードバックを経て実装される手法に質的な変化をもたらすことが示された．

[8]Reinforcement Learning from Human Feedback（人間のフィードバックによる強化学習）の略であり，LLM の出力に対する人間の評価を用いて応答を調整する手法を指す．

5 まとめ

本研究では，フィードバック設計の違いが株式投資戦略の改善プロセスに与える影響を検証した．具体的には，提示する情報の範囲（基本情報のみ／基本情報＋追加情報）および提示形式（テキストのみ／テキスト＋プロット）という 2 軸に基づき，3 種類の条件下で反復的な改善プロセスを比較した．

実験の結果，パフォーマンスの改善幅およびコードの変更量には使用する LLM モデルの違いが顕著な効果を及ぼす一方で，コード変更の質，つまり提案される手法の内容にはフィードバック設計が大きな影響を及ぼすことがわかった．すなわち，フィードバック設計の差分よりも，モデル選択が改善プロセスの到達水準をより強く規定している可能性がある．

この結果は，LLM を用いた株式投資戦略の自動生成プロセスにおいて，フィードバック設計以上にモデル選択が主要な設計要素となり得ることを示唆している．

今後の課題としては，観測されたモデル差が単なる挙動特性の差ではなく，改善プロセスのアーキテクチャやプロンプトとの親和性に起因する可能性を検証することが挙げられる．この点を明らかにすることは，各モデルの戦略改善におけるポテンシャルを引き出す上で重要な課題となる．

A 実装手法の補足詳細

表 A.1 は Gemini を用いて戦略コード間の差分を抽出し，表にまとめたものである．

参考文献

[1] Yifei Dong, Fengyi Wu, Kunlin Zhang, Yilong Dai, Sanjian Zhang, Wanghao Ye, Sihan Chen, and Zhi-Qi Cheng. Large language model agents in finance: A survey bridging research, practice, and real-world deployment. *Findings of the Association for Computational Linguistics: EMNLP*, Vol. 2025, pp. 17889–17907, 2025.

[2] 中川慧, 平野正徳, 高野海斗. 本邦金融分野における大規模言語モデルに関するサーベイと展望.

[3] Kemal Kirtac and Guido Germano. Sentiment trading with large language models. *Finance Research Letters*, Vol. 62, p. 105227, 2024.

[4] Kaito Takano, Kei Nakagawa, and Yugo Fujimoto. Generation of market comments and outlooks in mutual fund disclosure documents using llm. 人工知能学会論文誌 (Web), Vol. 39, No. 4, pp. 23–1, 2024.

[5] Yangyang Yu, Zhiyuan Yao, Haohang Li, Zhiyang Deng, Yuechen Jiang, Yupeng Cao, Zhi Chen, Jordan Suchow, Zhenyu Cui, Rong Liu, et al. Fincon: A synthesized llm multi-agent system with conceptual verbal reinforcement for enhanced financial decision

表 A.1: モデル×プロンプト別の実装手法の特徴と代表例

モデル	Prompt 1	Prompt 2	Prompt 3
Claude 4.5	**王道ファクターの網羅：**200 日線，ROE，FCF 利回り，Z スコアなど，伝統的クオリティ・バリュー戦略が中心.	**中立化の徹底：**スタイル・セクター中立化，為替ベータ推定，VIX レジームなど，外部環境への耐性を強化.	**動的ゲーティング：**ローリング IC によるレジーム制御，確信度加重，戦略の取捨選択（棄却判断含む）.
Claude Opus	**複合アルファの形成：**PEAD（決算後ドリフト），信用取引フロー，モメンタムの平滑化など，市場の歪みを重視.	**非線形・詳細分析：**2 次サイズ項（Size_sq），リッジ回帰，決算スケジュール減衰など，数学的深度が増す.	**最適化計算の導入：**SLSQP（二次計画法），多因子回帰，ドローダウンベースのスケーリングなど，高度な運用管理.
Claude Sonnet	**実用的なフィルター群：**TOPIX トレンド，売買代金加重，ストップ高安回避など，実運用上の制約を重視.	**ポートフォリオ特性制御：**リスクパリティ（ERC），ROIC 導入，ボラティリティ・ターゲットによるエクスポージャー調整.	**計算効率と柔軟性：**Pandas 互換性向上，イントラデイ・モメンタム中立化，動的流動性スケーリング.
Gemini 3 Flash	**ユニークな命名と発想：**GQA（Growth/Quality/Anchor），ベラシティ（CF 裏付け），機関投資家確信度（MIC）など，独自指標の提案.	**効率性とマクロ連動：**SOX/QQQ ベータ，資本規律重視，ボリューム・スタビリティなど，外部市場との相関を分析.	**アルゴリズムの高速化：**OLS 高速化（2 × 2 方程式ソルバー，クラメールの公式），ハッシュ関数による銘柄グループ化.
Gemini 3 Pro	**センチメントと需給：**イントラデイ・センチメント，マージン・クラウディング，インプライド EPS の算出.	**4 ピラー（柱）構築：**クオリティ・バリュー・トレンド・セーフティの統合，ペイン・リバージョン（過熱感）.	**平滑化と整合性：**イントラデイ・シャープレシオ，インデックス不整合の修正，多期間モメンタム分解.
GPT-5	**マクロガードレールの導入：**USDJPY ベータ，VIX フロア，リテール混雑のペナルティなど，マクロリスクを遮断.	**高度な中立化技術：**リッジ回帰によるスタイル中立化，乖離アクセラレーション，データクリッピング.	**リファクタリングと純化：**L1 ノルムによる再正規化，出力カラムの固定，コード構造の極限までの整理.
GPT-5 mini	**堅牢な実装基盤：**ロバスト Z スコア（MAD），スタッガード・リバランス，シグナル持続性（Persistence）.	**微細なパラメータ調整：**ベータの縮小推定（Shrinkage），ボラティリティ・フロア，関数のモジュール化.	**実行安定性の追求：**Future-Warning 回避，IC ゲーティング，反復的比例配分によるウェイト正規化.
GPT-5 nano	**コードの軽量化と高速化：**transform 活用，ベクトル化演算，最小二乗法（np.linalg.lstsq）の直接利用.	**インターフェースの汎用化：**デフォルト引数制御，パラメータの汎用名化（param1-8），型変換の厳格化.	**ロジックの再編：**クロスセクショナル・ウィンゾライジング，ダイバージェンス加速度，未使用引数の削除.

making. *Advances in Neural Information Processing Systems*, Vol. 37, pp. 137010–137045, 2024.

[6] Kaito Takano, Masanori Hirano, and Kei Nakagawa. Modeling hawkish-dovish latent beliefs in multi-agent debate-based llms for monetary policy decision classification. In *International Conference on Principles and Practice of Multi-Agent Systems*, pp. 488–505. Springer, 2025.

[7] Saizhuo Wang, Hang Yuan, Leon Zhou, Lionel Ni, Heung Yeung Shum, and Jian Guo. Alpha-gpt: Human-ai interactive alpha mining for quantitative investment. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 196–206, 2025.

[8] Lang Cao. Chain-of-alpha: unleashing the power of large language models for alpha mining in quantitative trading. *arXiv preprint arXiv:2508.06312*, 2025.

[9] Richard C Grinold and Ronald N Kahn. Active portfolio management. 2000.

[10] Marcos Lopez De Prado. *Advances in financial machine learning*. John Wiley & Sons, 2018.