

Article

Not peer-reviewed version

---

# Financial Network Analysis Using Polymodel Theory

---

S. Z. <sup>\*</sup>, Zhikang Dong, [Raphael Douady](#)

Posted Date: 25 March 2025

doi: 10.20944/preprints202409.1453.v2

Keywords: polymodel theory; financial network construction



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Article

# Hedge Fund Portfolio Construction Using PolyModel Theory and iTransformer

S. Z<sup>\*</sup>, Zhikang Dong<sup>1</sup> and Raphael Douady<sup>2</sup>

<sup>1</sup> Stony Brook University

<sup>2</sup> University of Paris I: Pantheon-Sorbonne

\* Correspondence: alysia.zhao@gmail.com

**Abstract:** This paper presents a novel approach to financial network analysis by leveraging PolyModel theory. Traditional financial networks often rely on correlation matrices to represent relationships between assets, but these fail to capture the complex, non-linear interactions prevalent in financial markets. In response, we propose a method that quantifies the relationship between financial time series by comparing their reactions to a broad set of environmental risk factors. This method constructs a network based on the inherent similarities in how assets respond to external risks, offering a more robust representation of financial markets. We introduce several network topological properties, such as eigenvalues, degree, and clustering coefficients, to measure market stability and detect financial instabilities. These metrics are applied to a real-world dataset, including the DOW 30, to predict market drawdowns. Our results indicate that this PolyModel-based network framework is effective in capturing downside risks and can predict significant market drawdowns with high accuracy. Furthermore, we enhance sensitivity to smaller market changes by introducing new time-series indicators, which further improve the predictive power of the model.

**Keywords:** PolyModel theory; financial network construction

## 1. Introduction

Portfolio construction remains a central topic in quantitative finance research. Beginning with the Capital Asset Pricing Model (CAPM) [1], the theory of portfolio construction has continuously evolved, incorporating a range of new techniques and theories over time. Data-driven methods, particularly in fields like computer vision [2–12], natural language [13,14], time series analysis [15,16], biomedical research [17–20] audio processing [21–23], content moderation [24], statistics [25,26] and science [27–31], have shown significant advancements. In recent years, those techniques have notably impacted quantitative finance [32,33], from predicting asset prices [34] to hedging risks in derivatives [35].

However, when constructing portfolios, a key problem is that a lot of financial time series data are sparse, making it challenging to apply machine learning methods. Polymodel theory can solve this issue and demonstrate superiority in portfolio construction from various aspects. To implement the PolyModel theory for constructing a hedge fund portfolio, we begin by identifying an asset pool, utilizing over 10,000 hedge funds for the past 29 years' data. PolyModel theory also involves choosing a wide-ranging set of risk factors, which includes various financial indices, currencies and commodity prices. This comprehensive selection mirrors the complexities of the real-world environment. Leveraging on the PolyModel theory, we create quantitative measures such as Longterm Alpha, Long-term Ratio, and SVaR. We also use more classical measures like the Sharpe ratio or Morningstar's MRAR. To enhance the performance of the constructed portfolio, we also employ the latest deep learning techniques (iTransformer) to capture the upward trend, while efficiently controlling the downside, using all the features. The iTransformer model is specifically designed to address the challenges in high-dimensional time series forecasting and could largely improves our strategies. More precisely, our strategies achieve better Sharpe ratio and annualized return. The

above process enables us to create multiple portfolio strategies aiming for high returns and low risks when compared to various benchmarks. The integration of PolyModel theory with machine learning methods facilitates a nuanced and precise understanding of hedge fund returns. This amalgamation enables us to overcome challenges related to hedge fund data, offering a more robust methodology for analyzing hedge fund performance and guiding investment decisions. This is a very meaningful attempt to combine fundamental statistical analysis with latest machine learning techniques.

## 2. PolyModel Theory

The origin of the idea of PolyModel theory and its mathematical foundations can be dated back to [36,37]. Since PolyModel theory is more a framework rather than a single statistical analysis tool, after its first introduction, quite a few extensions and applications have been proposed and studied. For a nice overview of more applications and the history of this theory, one can check [38] while for more concise mathematical description and its implementation, one can consult [39,40].

Before we step into the mathematical descriptions, let's first discuss the core idea and intuition behind PolyModel theory to get a better understand of it.

The core idea of PolyModel theory is to combine a large enough collection of valid description of one aspect of the same target or reality in order to get a as close as possible fully understanding of the target's nature. In financial industry, the target is usually the return of some asset in which one wants to invest.

If we image that the target is alive, like an animal, then PolyModel theory can be regarded as a methodology to observe how this animal reacts to the outside environment, especially, to each single environment factor. If we can capture and understand all its reactions, then we can fully characterize this animal. This idea is, surprisingly, similar to a Python terminology called "Duck Typing": "when an object quacks like a duck, swims like a duck, eats like a duck or simply acts like a duck, that object is a duck." Though coming from very different fields, the two ideas introduced above can both be viewed as an variant of Phenomenology [39]: "Literally, phenomenology is the study of 'phenomena': appearances of things, or things as they appear in our experience, or the ways we experience things, thus the meanings things have in our experience."

After the high-level description of PolyModel theory, we now turn back to its mathematical descriptions and how to construct features with strong description or prediction power.

### 2.1. Mathematical Formulation and Model Estimation

#### 2.1.1. Model Description and Estimation

There are two fundamental components in PolyModel theory:

- A pool of target assets  $\{Y_i\}_{i \in I}$  which are the components of the portfolios one want to construct.
- A very large pool of risk factors  $\{X_j\}_{j \in J}$  which form a proxy of the real-world financial environment.

The mathematical description of the PolyModel theory can be formulated as follows:

For every target  $Y_i, i \in I$ , there is a collection of (relatively simple) regression models:

$$\begin{cases} Y_i = \Phi_{i1}(X_1) + \epsilon_1 \\ Y_i = \Phi_{i2}(X_2) + \epsilon_2 \\ \dots \\ Y_i = \Phi_{in}(X_n) + \epsilon_n \end{cases} \quad (1)$$

where

- $n$  is the number of the risk factors.
- $\Phi_{ij}$  is assumed to capture the major relationship between independent variable  $X_j$  and dependent variable  $Y_i$ ; in practice, it is usually a polynomial of some low degree.

- $\epsilon_j$  is the noise term in the regression model with zero mean; usually it is assumed to be normal distribution but does not have to be.

In practice, we usually assume that

$$\Phi_{ij}(x) = \sum_{k=0}^4 \beta_{ij}^k H_k(x), \quad (2)$$

where  $H_k(x)$  is the Hermitian polynomial of degree  $k$ . Based on authors' practical experience, a polynomial of degree of 4 is flexible enough to capture nonlinear but essential relation between target and risk factor while usually suffer bearable overfitting.

For each target and risk factor pair  $(Y_i, X_j)$ , assume that we have their observations:  $Y_i$  and  $X_j$  for time  $t = 1, 2, \dots, T$ , then we can write each regression model from (1) into matrix format

$$\vec{Y}_i = \mathbf{H}_j^T \vec{\beta}_{ij} + \vec{\epsilon}_{ij}, \quad (3)$$

where

- $\vec{Y}_i$  denotes the vector of the target time series such of return of hedge fund

$$\begin{bmatrix} Y_i(t_1) \\ Y_i(t_2) \\ \vdots \\ Y_i(t_T) \end{bmatrix}.$$

- $H_j$  denotes the following matrix of the risk factor  $X_j$

$$\begin{bmatrix} H_0(X_j(t_1)), H_0(X_j(t_2)), H_0(X_j(t_3)), \dots, H_0(X_j(t_T)) \\ H_1(X_j(t_1)), H_1(X_j(t_2)), H_1(X_j(t_3)), \dots, H_1(X_j(t_T)) \\ \dots \\ H_4(X_j(t_1)), H_4(X_j(t_2)), H_4(X_j(t_3)), \dots, H_4(X_j(t_T)) \end{bmatrix}.$$

which is a  $5 \times T$  matrix, where  $H_k(x)$  is the Hermitian polynomial of degree  $k$ .

- $\vec{\epsilon}_{ij}$  denotes the regression error vector

$$\begin{bmatrix} \epsilon_{ij}(t_1) \\ \epsilon_{ij}(t_2) \\ \vdots \\ \epsilon_{ij}(t_T) \end{bmatrix}.$$

- $\vec{\beta}_{ij}$  is the coefficient vector of length 5

$$\begin{bmatrix} \beta_{ij}^0 \\ \beta_{ij}^1 \\ \vdots \\ \beta_{ij}^4 \end{bmatrix}.$$

Now let's briefly discuss how to estimate the coefficients. From the model description above, we can see that PolyModel theory technically belongs to the realm of statistical regression models, thus, all the common well-established parameter estimation methods can be applied to it. From a practical point of view, we choose to use the Ridge regression [41]

$$\widehat{\vec{\beta}}_{ij,\lambda} := \arg \min_{\{\vec{\beta}_{ij} \in \mathbb{R}^5\}} [(\vec{Y}_i - \mathbf{H}_j^T \vec{\beta}_{ij})^T (\vec{Y}_i - \mathbf{H}_j^T \vec{\beta}_{ij}) + \lambda \|\vec{\beta}_{ij}\|^2], \quad (4)$$

We can see that the fitted coefficients are functions of the hyper-parameter  $\lambda$ ; to determine the optimal value for each simple regression, one can apply any state-of-art hyper-parameter tuning trick

such as grid search plus cross-validation. However, we would like to point out that in PolyModel theory, we need to deal with a huge amount of risk factors, and our polynomial in the regression equation is only of degree 5, thus, our major concern for using ridge regression is to make the matrix  $\mathbf{H}_j \mathbf{H}_j^T + \lambda \mathbf{I}_{5 \times 5}$  invertible, thus, we usually choose a relatively small number as the value of  $\lambda$  for all the target time series and risk factor pairs.

## 2.2. Feature Importance and Construction

One of the major goals of PolyModel theory is to find a set of risk factors which are most important to the target time series after fitting hundreds of simple regressions. In this section, we will first discuss the fundamental statistical quantities based on fitting the numerous simple regressions, then we will use them as building blocks to construct the features which will be used by the machine learning algorithms.

### 2.2.1. Fundamental Statistical Quantities

#### 1. $R^2$ and adjusted $R^2$

As PolyModel is a collection of simple regression models, then it is quite natural to talk about  $R^2$  for every simple regression model.

$R^2$ , also known as coefficient of determination, is one of the most common criteria to check the fitting goodness of a regression model. It is defined as follows:

$$R^2 := \frac{ESS}{TSS} = 1 - \frac{RSS}{TSS}, \quad (5)$$

where, if we denote  $\mathbf{H}_j^T \hat{\beta}_{ij}$  by  $\hat{Y}_i$ , and denote the vector of average of entries of  $\vec{Y}_i$  with the same length by  $\bar{Y}_i$ , then

- ESS is the explained sum of squares which is  $(\hat{Y}_i - \bar{Y}_i)^T (\hat{Y}_i - \bar{Y}_i)$ .
- RSS is the residual sum of squares which is  $(\vec{Y}_i - \hat{Y}_i)^T (\vec{Y}_i - \hat{Y}_i)$ .
- TSS is the total sum of squares which is  $(\vec{Y}_i - \bar{Y}_i)^T (\vec{Y}_i - \bar{Y}_i)$ .

Moreover, it is a well-known fact in regression theory that  $TSS = RSS + ESS$ .

$R^2$  measures how much total uncertainty is explained by the fitted model based on the observed data, thus, the higher  $R^2$  is, the better the model should be. However, this statistic does not take the number of model complexity into consideration, thus, a high  $R^2$  may also indicates overfitting and usually this is the case (for instance, in a one dimension problem given general  $n$  data points, there is usually a degree  $n + 1$  polynomial which can pass through every one of them). Various modifications have been introduced, one very direct generalization is the adjusted- $R^2$ :  $1 - \frac{\frac{RSS}{(n-p)}}{\frac{TSS}{(n-1)}}$

where  $n$  is the number of observations and  $p$  is the number of coefficients in the regression model.

#### 2. Target Shuffling and $P$ -Value Score

To avoid fake strong relationship between target and risk factors, we apply target shuffling which is particular useful to identify "cause-and-effect" relationship. By shuffling the the targets, we have the chance to determine if the relationship fitted by the regression model is significant enough by checking the probability of the  $R^2$  we have seen based on the observations.

The procedure can be summarized as follows:

- Do random shuffles on the target time series observations many times, say  $N$  times. For each  $X_j$ , let we assume that there are  $T$  data points  $\{(Y_i(t_k), X_j(t_k))\}_{k=1}^T$ . We fix the order of  $X_j(t_k)$ , and we do  $N$  times of random shuffle of  $Y_i(t_k)$ . In this way, we try to break any relation from the original data set and create any possible relations between the target and risk factor.
- For each newly ordered target observations  $\{(Y'_i(t_k), X_j(t_k))\}_{k=1}^T$ , we can fit a simple regression model and calculate the  $R^2$ . Then we get

$$R^2_{shuff} = \{R^2_{(1)}, R^2_{(2)}, \dots, R^2_{(N)}\}.$$

Thus, we have a population of  $R^2$  based on above procedures.

- Evaluate the significance of the  $R^2$  calculated from the original data, for instance, we can calculate the p-value of it based on the  $R^2$  population from last step. Here we assume that our original  $R^2$  for target asset  $Y_i$  and risk factor  $X_j$  is denoted as  $R_{ij}^2$ . Then, we could define

$$p_{ij} = P(R^2 > R_{ij}^2).$$

- We compute  $-\log(p_{ij})$  and call it  $P$ -Value Score of target asset  $Y_i$  and risk factor  $X_j$  which indicates the importance of the risk factor  $X_j$  to the target asset time series  $Y_i$ .

The higher the  $P$ -Value Score is, the more important the risk factor is. As we also need to take different regimes over the time into the picture, at each time stamp, we only look at the past 3 years' return data, and thus, we can have a dynamic  $P$ -Value Score series for each target asset  $Y_i$  and risk factor  $X_j$  pair.

### 2.2.2. Feature Construction

Now we are ready to construct the features based on the statistical quantities introduced above and the data themselves. We will briefly discuss how to construct them and their meanings. More details can be found in [40].

#### 1. Sharpe Ratio

It is one of the most common statistical metric to estimate the performance of a portfolio. Roughly speaking, it is the ration between the portfolio return and its volatility, thus, usually is regarded as a measure of the ratio between reward and risk.

Assume  $R$  represents the return of the target portfolio,  $R_f$  represents the return of the benchmark financial time series, for instance, RFR. Then Sharpe Ratio is defined as

$$\text{Sharpe Ratio} := \frac{E(R-R_f)}{\sqrt{\text{var}(R-R_f)}}.$$

In practice, one may also ignore the benchmark if it is very small or static. Notice that Sharpe Ratio is a feature that is only dependent on target portfolio itself.

#### 2. Morningstar Risk-adjusted Return (MRaR)

This is another feature mostly dependent on the target portfolio itself. Given the target portfolio (e.g. hedge fund return  $Y_i$ ), denote its return at time  $t$  as  $r_t$ ; denote the return of benchmark at time  $t$  as  $r_f$ , the MRaR over  $n$  months is defined as follows [42]

$$\text{MRaR} = \left( \frac{1}{n} \sum_{t=1}^n (1 + r_{Gt})^{-\gamma} \right)^{-\frac{n}{\gamma}} - 1,$$

$$r_{Gt} = \left( \frac{1+r_t}{1+r_f} \right) - 1,$$

where  $n$  is the total number of months in calculation period;  $r_{Gt}$  is the geometric excess return at month  $t$ ;  $\gamma$  is the risk aversion parameter, and *Morningstar*<sup>TM</sup> uses 2. Investors can adjust the value of  $\gamma$  according to their own risk flavors.

As mentioned in [43], the main assumption is that investors are rational and willing to give up a small portion of their expected return to achieve a better certainty. This is metric is similar to Sharpe ratio but has more advantages. More discussions on its advantages can be found in [44].

#### 3. StressVaR (SVaR)

SVaR can be regarded as a good alternative risk measure instead of VaR, in fact, it can be regarded as a factor model-based VaR. However, its strength resides in the modeling of nonlinearities and the capability to analyze a very large number of potential risk factors[45].

There are three major steps in the estimation of StressVaR of a hedge fund  $Y_i$ .

- (a) Most relevant risk factors selection: for each risk factor  $X_j$ , we can calculate the  $P$ -Value Score of it with respect to  $Y_i$ . Recall Section 2.5.2, this score can indicate the explanation power of risk factor  $X_j$ , and the application of target shuffling improves the ability of our

model in preventing discovering non-casual relations. Once a threshold of  $P$ -Value Score is set, we can claim that all the risk factors  $X_j$  whose  $P$ -Value Score is above the threshold are the most relevant risk factors, and denote the whole set of them as  $\Gamma_i$ .

- (b) Estimation of the Maximum Loss of  $Y_i$ : For every risk factor  $X_j \in \Gamma_i$ , using the fitted polynomial for the pair  $(Y_i, X_j)$ , we can predict the return of  $Y_i$  for all risk factor returns from 1st to 99th quantiles of the risk factor distributions. In particular, we are interested in the potential loss of  $Y_i$  corresponding to  $\alpha\% = 98\%$  of the factor returns. Once this is estimated for one factor  $X_j$ , we can define  $SVaR_{ij}$  for the pair  $(Y_i, X_j)$  as follows:

$$SVaR_{ij} := \sqrt{\hat{Y}_{i,j,max}^2 + \sigma(Y_i)^2 \cdot (1 - R^2) \cdot \xi^2}$$

where

- $\hat{Y}_{i,j,max}$  is the maximum potential loss corresponding to  $\alpha$  quantile of risk factor  $X_j$ .
- $\sigma(Y_i)^2 \cdot (1 - R^2)$  is unexplained variance under the ordinary least square setting which can be estimated by the following unbiased estimator if penalty terms are added to the regression models

$$\frac{\sum(Y_i - \hat{Y}_i)^2}{n-p},$$

where  $p$  is the degree of freedom of the regression model.

- $\xi = \varphi^{-1}(\alpha) \approx 2.33$  where  $\varphi$  is the cumulative distribution function (cdf) of standard normal distribution.

- (c) Calculation of StressVaR: The definition of StressVaR of  $Y_i$  is

$$SVaR_i := \max_{j \in \Gamma_i} SVaR_{ij}.$$

#### 4. Long-term alpha (LTA)

For the given hedge fund and risk factor pair  $(Y_i, X_j)$ , assume we already fitted the regression polynomial  $\Phi_{ij}(x)$ . Assume that  $\theta_{j,q}$  represents the  $q$ -quantile of the empirical distribution of  $X_j$  where  $q = 1\%, 16\%, 50\%, 84\%, 99\%$ . They are calculated using the very long history of the factor. The extremes 1% and 99% are computed by fitting a Pareto distribution on the tails.

Then we define

$$LTA(Y_i, X_j) := \sum_{q=1\%}^{99\%} w_q \Phi_{ij}(\theta_{j,q}),$$

subject to  $E(X_j) = \sum_{q=1\%}^{99\%} w_q \theta_{j,q}$ , where  $w_q$  correspond to Lagrange method of interpolating an integral and are hyper-parameters.

The global LTA (long-term average) is the median of the factor expectations for selected factors.  $LTA_i$  for  $Y_i$  is defined as the 50th quantile among all the  $LTA(Y_i, X_j)$  values, where  $X_j \in \Gamma_i$  represents the selected ones.

#### 5. Long-term ratio (LTR)

Once we get the  $LTA_i$  and  $SVaR_i$  for  $Y_i$ ,  $LTR_i$  is simply defined as

$$LTR_i := \frac{LTA_i}{SVaR_i}.$$

#### 6. Long-term stability (LTS)

For fund  $Y_i$ ,  $LTS_i := LTA_i - \kappa \cdot SVaR_i$  where  $\kappa$  is a hyper-parameter whose value is set to 5%.

Besides the features constructed above, we also include some more standard features for our financial time series research: asset under management (AUM) of each hedge fund, volume of each hedge fund, and historical returns for each hedge fund and risk factor. All of them will be used as input features when applying machine learning techniques below.

### 3. Methodology

Given the carefully chosen risk factor pool and the set of hedge funds to invest, we first apply PolyModel theory to construct the features introduced in the previous section. Notice that these

features can be regarded as a dynamical encoding of the hedge funds' returns and their interactions with the whole financial environment.

We then will apply modern machine learning algorithms to predict the performance of each hedge fund. We particularly choose to apply transformer techniques in our prediction due to its string performance in time series related forecasting researches during recent years [46]. Moreover, we will apply one of its latest variants called inverted transformer in our study.

In the rest of this section, we first introduce inverted transformer, then discuss how to apply it to our hedge fund performance prediction task in details.

### 3.1. Inverted Transformers (iTransformer)

Inverted Transformers (iTransformer) [47] is designed for multivariate time series forecasting. We combine this method with PolyModel theory to generate effective portfolio construction. Suppose we extract  $N$  features with  $T$  timesteps, denoted as  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_T\} \in \mathbb{R}^{T \times N}$ . Based on those historical observations, we can forecast the future  $S$  time steps target  $\mathbf{Y} = \{\mathbf{x}_{T+1}, \dots, \mathbf{x}_{T+S}\} \in \mathbb{R}^{S \times N}$ . Instead of regarding multivariate features of the same time step as a temporal token, the iTransformer tokenize the whole time series input of each feature as the token, which focus on representation learning and correlation measurement of multivariate time series.

$$\mathbf{h} = \text{Embedding}(\mathbf{X}), \quad (6)$$

where  $\mathbf{h} = \{\mathbf{h}_1, \dots, \mathbf{h}_N\} \in \mathbb{R}^{N \times D}$ . We use multi-layer perceptron (MLP) to project raw time series data into  $D$ -dimensional latent space. [47] shows that the temporal information has been processed by MLP, the position embedding in original Transformer [48] is not necessary anymore.

We apply Layer normalization (LN) [49] to token  $\mathbf{h}$  across time steps. Unlike the common Transformer frameworks, which apply LN across different features, iTransformer [47] normalizes each feature token to a standard Gaussian distribution, which helps keep patterns in each feature. [50,51] also prove that this technique are helpful in solving non-stationary time series problem.

$$\mathbf{H} = \frac{\mathbf{h}_n - \text{Mean}(\mathbf{h}_n)}{\sqrt{\text{Var}(\mathbf{h}_n)}}, n = 1, \dots, N. \quad (7)$$

The original Transformer [48] uses the attention mechanism to process temporal information for encoded features. The iTransformer [47] uses this attention mechanism to model feature correlations since each token represents the whole time series data of a feature. Suppose there are linear projections  $\mathbf{W}_Q \in \mathbb{R}^{D \times d_k}$ ,  $\mathbf{W}_K \in \mathbb{R}^{D \times d_k}$  and  $\mathbf{W}_V \in \mathbb{R}^{D \times d_k}$ . We can obtain query, key and value matrices as  $\mathbf{Q} = \mathbf{H}\mathbf{W}_Q$ ,  $\mathbf{K} = \mathbf{H}\mathbf{W}_K$  and  $\mathbf{V} = \mathbf{H}\mathbf{W}_V$ . Then, the self-attention mechanism is computed as

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V}. \quad (8)$$

Traditional transformer models typically utilize temporal tokens, analyzing all features at a single timestamp, which can limit their ability to effectively learn dependencies. One approach to address this limitation involves patching, where data points along the time axis are grouped prior to tokenization and embedding. However, this method may suffer from insufficiently large grouping ranges, failing to capture all necessary dependencies. In contrast, the iTransformer adopts an innovative approach by viewing the time series from an inverted perspective. This allows it to consider the same feature across multiple timestamps, significantly enhancing its capacity to discern dependencies and multivariate correlations. This distinct capability positions the iTransformer as a superior alternative in scenarios demanding nuanced temporal analysis.

### 3.2. Hedge Fund Performance Prediction

We apply iTransformer algorithm directly in our research. The input features are those described in section 2.2.2. Regarding the output, for each target hedge fund, we predict the probability of the

trend rather than the value of its return, in particular, we assume that there are three status of the return trend: up, down and unchanged (we set a prior threshold for the hedge fund return. If the absolute value of the return is smaller than the threshold, we define its status as unchanged. Otherwise, the status is up if the return is positive and the status is down if the return is negative).

We apply the implementation of iTransformer from [47] in a straight forward manner where interested readers can find all the technical details. Thus, rather than more discussions on iTransformer, we will discuss why we choose the trend rather than the value of hedge fund returns as our prediction output.

As already pointed out in some recent research such as [52,53], it is more useful to correctly classify the trend of returns rather than to provide a predicted result which is close to the real return. For instance, one has a portfolio and can predict its return as close as the realized one but with an opposite sign, this may cause a significant negative impact on one's pnl and is not favored.

Moreover, our target assets are hedge funds whose returns usually have very large magnitude, thus, once we can predict the return status correctly and select those hedge funds whose next returns are positive, we will have a good chance to achieve a reasonably high total return. On the other side, PolyModel theory is quite good at identifying risk factors which may cause large drops of the target assets. Thus, the combination of these two theories can give us a better chance to create a portfolio with large positive return and small drawdown.

#### 4. Portfolio Construction

Based on the theories and methodologies introduced in previous sections, we are ready to construct our portfolio. We rebalance our portfolio monthly. Before the end of each month, we apply iTransformer to predict the probability on whether the return of hedge fund  $Y_i$  for the next month is positive which is denoted as  $p_i$ . We select the top 50% hedge funds with the largest probabilities of having a positive return for the next month. We keep those hedge funds which are currently held in our portfolio if they are selected, and sell the in-selected ones in our hands. The collected cash are reinvested evenly to buy the rest selected hedge funds which are not in current portfolio. We call this strategy simple average portfolio (SA). A second proposed strategy, which is denoted as weighted average portfolio (WA), is almost identical to SA except that the weights of the selected fund in the portfolio are based on the their AUM.

#### 5. Experiments and Results

In this section, we will give an overview of the data used for our study, the benchmarks to compare with and the performance of our portfolio. The same set of data and benchmarks are also used in [52].

##### 5.1. Data Description

As mentioned in the introduction of PolyModel theory, there are two datasets: risk factors and target hedge funds. The data sets cover a long period from April 1994 to May 2023. These data will be used to construct the features introduced in section 2.2.2, and the set of hedge fund will be used to construct the portfolio. Below let's look at the snapshots of some of the representatives of these two data sets.

Regarding risk factors, our study incorporates an extensive universe comprising hundreds of risk factors from different domains, including equities, coupons, bonds, industrial indexes, and more. We list some of the risk factors:

**Table 1.** List of the Risk Factors for Hedge Funds Portfolio Construction

Label	Code
T-Bil	INGOVS USAB
SWAP 1Y Zone USA In USD DIRECT VAR-LOG	INMIDR USAB
American Century Zero Coupon 2020 Inv (BTTTX) 1989	BTTTX
COMMODITY GOLD Zone USA In USD DIRECT VAR-LOG	COGOLD USAD
EQUITY MAIN Zone NORTH AMERICA In USD MEAN VAR-LOG	EQMAIN NAMM
...	...

we collect more than 10,000 hedge funds' data, including their monthly returns and AUMs. The selected hedge funds encompass a diverse range of strategies and characteristics. In terms of investment strategy, we have included fixed income, event driven, multi-strategy, long-short equities, macro, and various others. Geographically, the hedge funds under consideration span global, Europe, north America, Asia, and other regions. Here are some of the representatives:

**Table 2.** List of Hedge Funds

Fund Name
400 Capital Credit Opportunities Fund LP
Advent Global Partners Fund
Attunga Power & Enviro Fund
Barington Companies Equity Partners LP
BlackRock Aletsch Fund Ltd
Campbell Managed Futures Program
...

## 5.2. Benchmark Description

We select two fund of fund portfolios as the benchmarks, they are listed in Hedge Fund Research (HFR) [54], and let's quote their descriptions here directly:

- **HFRI Fund of Funds Composite Index (HFRIFOF)**

"Fund of Funds invest with multiple managers through funds or managed accounts. The strategy designs a diversified portfolio of managers with the objective of significantly lowering the risk (volatility) of investing with an individual manager. The Fund of Funds manager has discretion in choosing which strategies to invest in for the portfolio. A manager may allocate funds to numerous managers within a single strategy, or with numerous managers in multiple strategies. The minimum investment in a Fund of Funds may be lower than an investment in an individual hedge fund or managed account. The investor has the advantage of diversification among managers and styles with significantly less capital than investing with separate managers. PLEASE NOTE: The HFRI Fund of Funds Index is not included in the HFRI Fund Weighted Composite Index."

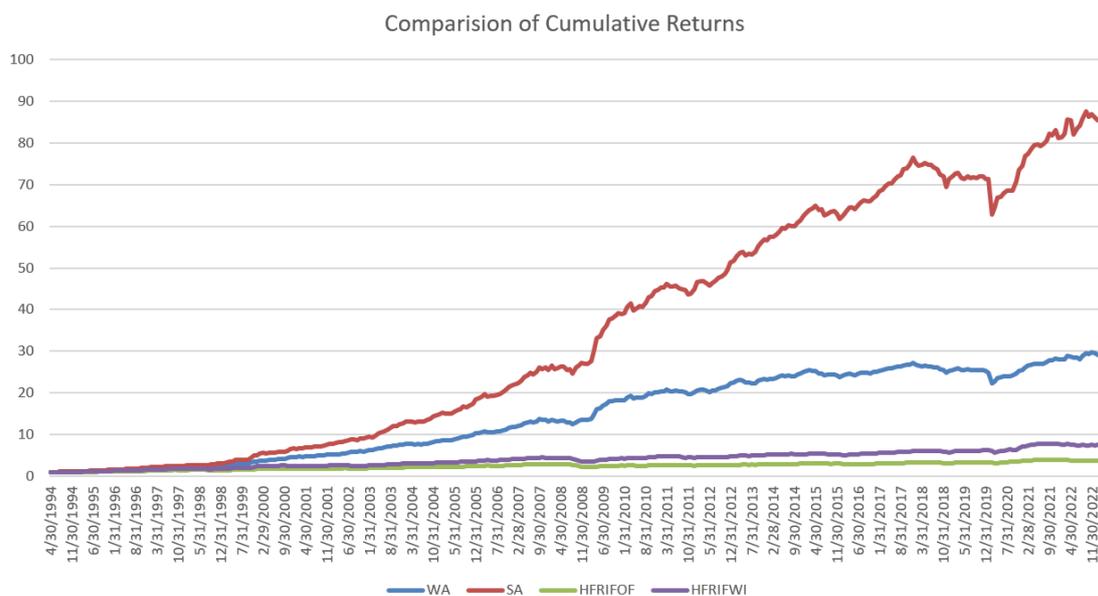
- **HFRI Fund Weighted Composite Index (HFRIFWI)**

"The HFRI Fund Weighted Composite Index is a global, equal-weighted index of single-manager funds that report to HFR Database. Constituent funds report monthly net of all fees performance in US Dollar and have a minimum of \$50 Million under management or \$10 Million under

management and a twelve (12) month track record of active performance. The HFRI Fund Weighted Composite Index does not include Funds of Hedge Funds."

### 5.3. Performance of the Constructed Portfolio

We follow the strategy discussed in section 4 to construct our portfolios. To calculate the features based on PolyModel theory, we use the past 36 months data to compute features such as SVaR and LTS for the next month's prediction purpose. We compare the performance of our strategies against the two benchmarks from section 5.2, assuming that we start with 1 dollar at 4/30/1994; the four portfolios are SA and WA, which are based on the selection method discussed in Section 4, and the two benchmarks HFRIFOF and HFRIFWI:



**Figure 1.** This figure plots the cumulative returns of the 4 strategies.

We can see that SA has the best performance regarding the cumulative return; WA is more stable and suffers much less drawdown than SA. Both strategies outperform the benchmarks significantly. It supports the power of the combination of PolyModel feature construction and deep learning techniques.

## 6. Conclusion

In this work, we considered the problem of portfolio construction when the available data is sparse. Especially, we considered to construct a portfolio of hedge funds.

To resolve this issue, we proposed the combination of PolyModel theory and iTransformer for hedge funds selection; the proposed strategies achieved much higher returns than the standard fund of fund benchmarks. This research also shows the power of combining domain knowledge and modern deep learning techniques.

## References

1. Fama, E.F.; French, K.R. The capital asset pricing model: Theory and evidence. *Journal of economic perspectives* **2004**, *18*, 25–46.
2. Li, J.; Li, D.; Savarese, S.; Hoi, S. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In Proceedings of the International conference on machine learning. PMLR, 2023, pp. 19730–19742.
3. Dong, Z.; Kim, J.; Polak, P. Mapping the Invisible: Face-GPS for Facial Muscle Dynamics in Videos. In Proceedings of the 2024 IEEE First International Conference on Artificial Intelligence for Medicine, Health and Care (AIMHC). IEEE, 2024, pp. 209–213.

4. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning transferable visual models from natural language supervision. In Proceedings of the International conference on machine learning. PMLR, 2021, pp. 8748–8763.
5. Kim, J.; Dong, Z.; Polak, P. Face-GPS: A Comprehensive Technique for Quantifying Facial Muscle Dynamics in Videos. In Proceedings of the Medical Imaging Meets NeurIPS: An official NeurIPS Workshop, 2023.
6. Dan, H.C.; Yan, P.; Tan, J.; Zhou, Y.; Lu, B. Multiple distresses detection for Asphalt Pavement using improved you Only Look Once Algorithm based on convolutional neural network. *International Journal of Pavement Engineering* **2024**, *25*, 2308169.
7. Dan, H.C.; Lu, B.; Li, M. Evaluation of asphalt pavement texture using multiview stereo reconstruction based on deep learning. *Construction and Building Materials* **2024**, *412*, 134837.
8. Zhang, D.; Zhou, F.; Jiang, Y.; Fu, Z. Mm-bsn: Self-supervised image denoising for real-world with multi-mask based on blind-spot network. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 4189–4198.
9. Zhang, D.; Zhou, F. Self-supervised image denoising for real-world images with context-aware transformer. *IEEE Access* **2023**, *11*, 14340–14349.
10. Dong, Z.; Beedu, A.; Sheinkopf, J.; Essa, I. Mamba Fusion: Learning Actions Through Questioning. *arXiv preprint arXiv:2409.11513* **2024**.
11. Fu, H.; Patel, N.; Krishnamurthy, P.; Khorrami, F. CLIPScope: Enhancing Zero-Shot OOD Detection with Bayesian Scoring. *arXiv preprint arXiv:2405.14737* **2024**.
12. Dong, Z.; Hao, W.; Wang, J.C.; Zhang, P.; Polak, P. Every Image Listens, Every Image Dances: Music-Driven Image Animation. *arXiv preprint arXiv:2501.18801* **2025**.
13. Lyu, W.; Zheng, S.; Pang, L.; Ling, H.; Chen, C. Attention-Enhancing Backdoor Attacks Against BERT-based Models. In Proceedings of the The 2023 Conference on Empirical Methods in Natural Language Processing.
14. Yang, R.; Wang, X.; Jin, Y.; Li, C.; Lian, J.; Xie, X. Reinforcement subgraph reasoning for fake news detection. In Proceedings of the Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2022, pp. 2253–2262.
15. Sun, Y.; Yang, S. Manifold-constrained Gaussian process inference for time-varying parameters in dynamic systems. *Statistics and Computing* **2023**, *33*, 142.
16. Lu, J.; Han, X.; Sun, Y.; Yang, S. CATS: Enhancing Multivariate Time Series Forecasting by Constructing Auxiliary Time Series as Exogenous Variables. *arXiv preprint arXiv:2403.01673* **2024**.
17. Tang, M.; Gao, J.; Dong, G.; Yang, C.; Campbell, B.; Bowman, B.; Zoellner, J.M.; Abdel-Rahman, E.; Boukhechba, M. SRDA: Mobile Sensing based Fluid Overload Detection for End Stage Kidney Disease Patients using Sensor Relation Dual Autoencoder. In Proceedings of the Conference on Health, Inference, and Learning. PMLR, 2023, pp. 133–146.
18. Kumar, S.; Datta, D.; Dong, G.; Cai, L.; Boukhechba, M.; Barnes, L. Leveraging mobile sensing and bayesian change point analysis to monitor community-scale behavioral interventions: A case study on covid-19. *ACM Transactions on Computing for Healthcare* **2022**, *3*, 1–13.
19. Dong, G.; Cai, L.; Kumar, S.; Datta, D.; Barnes, L.E.; Boukhechba, M. Detection and analysis of interrupted behaviors by public policy interventions during COVID-19. In Proceedings of the 2021 IEEE/ACM Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE). IEEE, 2021, pp. 46–57.
20. Wei, Y.; Gao, M.; Xiao, J.; Liu, C.; Tian, Y.; He, Y. Research and implementation of cancer gene data classification based on deep learning. *Journal of Software Engineering and Applications* **2023**, *16*, 155–169.
21. Gong, Y.; Chung, Y.A.; Glass, J. Ast: Audio spectrogram transformer. *arXiv preprint arXiv:2104.01778* **2021**.
22. Dong, Z.; Liu, X.; Chen, B.; Polak, P.; Zhang, P. Musechat: A conversational music recommendation system for videos. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 12775–12785.
23. Liu, X.; Dong, Z.; Zhang, P. Tackling data bias in music-avqa: Crafting a balanced dataset for unbiased question-answering. In Proceedings of the Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2024, pp. 4478–4487.
24. Xin, W.; Wang, K.; Fu, Z.; Zhou, L. Let community rules be reflected in online content moderation. *arXiv preprint arXiv:2408.12035* **2024**.
25. Chen, Y.; Xu, T.; Hakkani-Tur, D.; Jin, D.; Yang, Y.; Zhu, R. Calibrate and Debias Layer-wise Sampling for Graph Convolutional Networks. *Transactions on Machine Learning Research*.

26. Xu, T.; Zhu, R.; Shao, X. On variance estimation of random forests with Infinite-order U-statistics. *Electronic Journal of Statistics* **2024**, *18*, 2135–2207.
27. Abramson, J.; Adler, J.; Dunger, J.; Evans, R.; Green, T.; Pritzel, A.; Ronneberger, O.; Willmore, L.; Ballard, A.J.; Bambrick, J.; et al. Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature* **2024**, pp. 1–3.
28. Dong, Z.; Polak, P. CP-PINNs: Changepoints Detection in PDEs using Physics Informed Neural Networks with Total-Variation Penalty. In Proceedings of the Machine Learning and the Physical Sciences Workshop, NeurIPS 2023, 2023.
29. Dong, Z.; Polak, P. CP-PINNs: Data-Driven Changepoints Detection in PDEs Using Online Optimized Physics-Informed Neural Networks. In Proceedings of the 2024 Conference on AI, Science, Engineering, and Technology (AIxSET). IEEE, 2024, pp. 90–97.
30. Lin, F.; Guillot, K.; Crawford, S.; Zhang, Y.; Yuan, X.; Tzeng, N. An Open and Large-Scale Dataset for Multi-Modal Climate Change-aware Crop Yield Predictions. In Proceedings of the Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), 2024, pp. 5375–5386.
31. Lin, F.; Crawford, S.; Guillot, K.; Zhang, Y.; Chen, Y.; Yuan, X.; et al. MMST-ViT: Climate Change-aware Crop Yield Prediction via Multi-Modal Spatial-Temporal Vision Transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2023, pp. 5751–5761.
32. Papanicolaou, A.; Fu, H.; Krishnamurthy, P.; Healy, B.; Khorrami, F. An optimal control strategy for execution of large stock orders using long short-term memory networks. *Journal of Computational Finance* **2023**, *26*.
33. Papanicolaou, A.; Fu, H.; Krishnamurthy, P.; Khorrami, F. A Deep Neural Network Algorithm for Linear-Quadratic Portfolio Optimization With MGARCH and Small Transaction Costs. *IEEE Access* **2023**, *11*, 16774–16792.
34. Kumar, D.; Sarangi, P.K.; Verma, R. A systematic review of stock market prediction using machine learning and statistical techniques. *Materials Today: Proceedings* **2022**, *49*, 3187–3191.
35. Buehler, H.; Gonon, L.; Teichmann, J.; Wood, B. Deep hedging. *Quantitative Finance* **2019**, *19*, 1271–1291.
36. Cherny, A.; Douady, R.; Molchanov, S. On measuring nonlinear risk with scarce observations. *Finance and Stochastics* **2010**, *14*, 375–395.
37. Coste, C.; Douady, R.; Zovko, I.I. The stressvar: A new risk concept for extreme risk and fund allocation. *The Journal of Alternative Investments* **2010**, *13*, 10–23.
38. Douady, R. Managing the downside of active and passive strategies: Convexity and fragilities. *Journal of portfolio management* **2019**, *46*, 25–37.
39. Barrau, T.; Douady, R. *Artificial Intelligence for Financial Markets: The Polymodel Approach*; Springer Nature, 2022.
40. Zhao, S. PolyModel: Portfolio Construction and Financial Network Analysis. PhD thesis, Stony brook University, 2023.
41. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*; Springer series in statistics, Springer, 2009.
42. MRAR Illustrated.
43. Morningstar Risk-Adjusted Return.
44. The Morningstar Rating for Funds.
45. Coste, C.; Douady, R.; Zovko, I.I. The StressVaR: A New Risk Concept for Superior Fund Allocation. *arXiv preprint arXiv:0911.4030* **2009**.
46. Wen, Q.; Zhou, T.; Zhang, C.; Chen, W.; Ma, Z.; Yan, J.; Sun, L. Transformers in time series: A survey. *arXiv preprint arXiv:2202.07125* **2022**.
47. Liu, Y.; Hu, T.; Zhang, H.; Wu, H.; Wang, S.; Ma, L.; Long, M. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625* **2023**.
48. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Advances in neural information processing systems* **2017**, *30*.
49. Ba, J.L.; Kiros, J.R.; Hinton, G.E. Layer normalization. *arXiv preprint arXiv:1607.06450* **2016**.
50. Kim, T.; Kim, J.; Tae, Y.; Park, C.; Choi, J.H.; Choo, J. Reversible instance normalization for accurate time-series forecasting against distribution shift. In Proceedings of the International Conference on Learning Representations, 2021.
51. Liu, Y.; Wu, H.; Wang, J.; Long, M. Non-stationary transformers: Exploring the stationarity in time series forecasting. *Advances in Neural Information Processing Systems* **2022**, *35*, 9881–9893.

52. Siqiao, Z.; Dan, W.; Raphael, D. Using Machine Learning Technique to Enhance the Portfolio Construction Based on PolyModel Theory. *Research in Options* 2023 **2023**.
53. Vuletić, M.; Prenzel, F.; Cucuringu, M. Fin-gan: Forecasting and classifying financial time series via generative adversarial networks. *Quantitative Finance* **2024**, pp. 1–25.
54. Hedge Fund Research, <https://www.hfr.com/hfri-indices-index-descriptions>.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.