FolioLLM: Constructing portfolio of ETFs using Large Language Models

Stanford CS224N Custom Project

Andrey Popov (SUID: apopov), Oleg Roshka (SUID: oros)
Department of Computer Science
Stanford University
`apopov@stanford.edu, oros@stanford.edu`

Abstract

The primary goal of the project is to develop FolioLLM, a domain-specific large language model designed to assist investors and financial professionals in constructing portfolios of exchange-traded funds (ETFs). The project aims to fine-tune an LLM to understand user preferences, qualitative and quantitative investment features of funds, and portfolio optimization techniques, providing personalized investment advice. Secondary objectives include evaluating FolioLLM's performance, comparing it with existing baseline and fine-tuned models (both general and finance-specific LLMs), utilizing standard evaluation metric as well as ones from the investment domain. In addition we will explore integration of retrieval-augmented generation, Low-Rank Adaptation and Kolmogorov-Arnold Network techniques for enhancing model performance.

1 Key Information

- Mentor: Sonia Chiu
- External Collaborators: Ilia Badanin
- Team Contributions: Oleg worked on model pipeline construction and implementation of LoRA and KAN methods as well as model training. Andrey worked on data gathering and preparation, optimization and model evaluation. Ilia has advised on model structure and worked on classificator/ticker extractor and prompt construction.

2 Introduction

Widespread application of Large Language Models (LLMs) seem to have touched almost all areas and industries in life. Finance is no exception to this rule and for a while people have been trying to apply LLMs to solve specific tasks in the financial industry. Van Capelle (Van Cappelle, 2023) highlighted significant benefits for asset management industry, ranging from sentiment analysis, forecasting company data and zero-shot classifiers for thematic researches. Moreover, he demonstrated that LLMs can generate robust alpha, if applied to the investment strategy and forecasting. Sentiment analysis has been a long-standing area of application for LLMs, where it has been demonstrated that pre-trained FinBert model significantly improves accuracy on the finance-related tasks (Araci, 2019).

While LLM based models have been finding their use in consulting areas across different industries, one might want to explore the opportunity of applying LLMs within the wealth management industry. There have been several attempts to construct an investment portfolio using LLMs such as Cascio, 2023. The author pointed a significant potential in generative ability of the GPT model to construct abstract investment portfolio fitting into a particular theme, also highlighting that LLMs aren't performing well when it comes to precise asset allocation. Another important point raised is absence of relevant market data and inability to apply traditional optimization methods within LLM framework.

Perhaps one of the most relevant to our task papers has been published by industry professionals. Guo et al., 2023 have created a combined LLM suitable for asset management companies by fine-tuning a combination model on domain-specific data. Their approach involved continuous pre-training and supervised fine-tuning to adapt the model to the asset management domain. They also developed a comprehensive evaluation framework to assess the model's performance on financial exams, practical tasks, open question answering, and safety considerations.

Inspired by the work of Guo et al., 2023 we propose to develop a similar domain-specific LLM focused on ETFs and portfolio construction. By leveraging the advancements in LLMs and adapting them to the specific requirements of the ETF and portfolio management domain, we aim to create a model that can provide accurate, personalized, and actionable insights to investors and financial professionals.

3 Related Work

In the broader context of LLMs in finance, there have been notable works that target a wider scope compared to our ETF/portfolio-specific model. Wu et al., 2023 introduced BloombergGPT, a large language model specifically designed for the finance domain. BloombergGPT was trained on a vast corpus of financial data, including news articles, analyst reports, and company filings. The model demonstrates strong performance in various financial tasks, such as sentiment analysis, named entity recognition, and question answering. While BloombergGPT focuses on a broader range of financial applications, its architecture and training methodology can provide valuable insights for our model, particularly in terms of domain-specific pre-training and fine-tuning strategies.

Similarly, Yang et al., 2023 developed FinGPT, an open-source financial large language model. FinGPT was trained on a diverse set of financial data sources, including corporate filings, earnings call transcripts, and financial news. The model showcases impressive performance in tasks such as financial named entity recognition, sentiment analysis, and financial question answering. Although FinGPT is not specifically designed for portfolio management, its architecture and training approach can be adapted to our case. For instance, we can leverage the pre-training techniques used in FinGPT to enhance our model's understanding of financial concepts and market dynamics.

In the bigger overview of finance-oriented LLMs, Zhao et al., 2024 has pointed that LLMs would allow creating a wholistic investment approach in the fields of quantitative trading and portfolio management. He pointed on the opportunities for broader Finance-related GPT models like BloombergGPT (Wu et al., 2023) and FinGPT (Yang et al., 2023) to be applied to portfolio optimization process, potentially mitigating issues of a more traditional approaches.

Within portfolio management industry, new developments have been concentrated around implementing machine learning methods to overcome pitfalls of traditional portfolio theory based on Markowitz' pivotal work on mean-variance portfolio optimisation (Markowitz, 1952). Latest significant papers (López de Prado, 2020), (López de Prado, 2018) were aimed at applying PCA methods to assets' correlation matrix to identify underlying features and cluster assets accordingly in order to apply Nested Clustering Optimization (NCO) and Equal-Risk Parity (ERP) to compute optimal assets allocation. We aim to integrate one of the above mentioned portfolio optimization techniques in our LLM model.

4 Approach

This section details our approach to developing the FolioLLM, which includes a combination of pre-training, fine-tuning, and retrieval-augmented generation techniques. The main steps of our methodology are as follows:

Baseline Selection: Similar to Zhao et al., 2024 we started looking at the available pre-trained language model, such as BERT, Zephyr-7b, GPT-2 and Alpaca. In addition we also explored finance-focused models such as FinGPT and FinGU. Evaluating the size of models, their relevance of the task and time required for the training, we have focused on fine-tuning FinGU model with 0.5B parameters, and trained it on a vast amount of synthetic data.

Data Preparation and Augmentation: We used a template-based augmentation technique for data preparation, leveraging large language models (LLMs) like GPT-4 and Claude. This approach involved generating synthetic training data by applying transformation templates to raw ETF data. Specifically, we converted tabular datasets into text format to structure the data appropriately for training. The process included extracting relevant features such as ETF ticker, name, manager, returns, expense ratios, and other quantitative and qualitative attributes from the dataset.

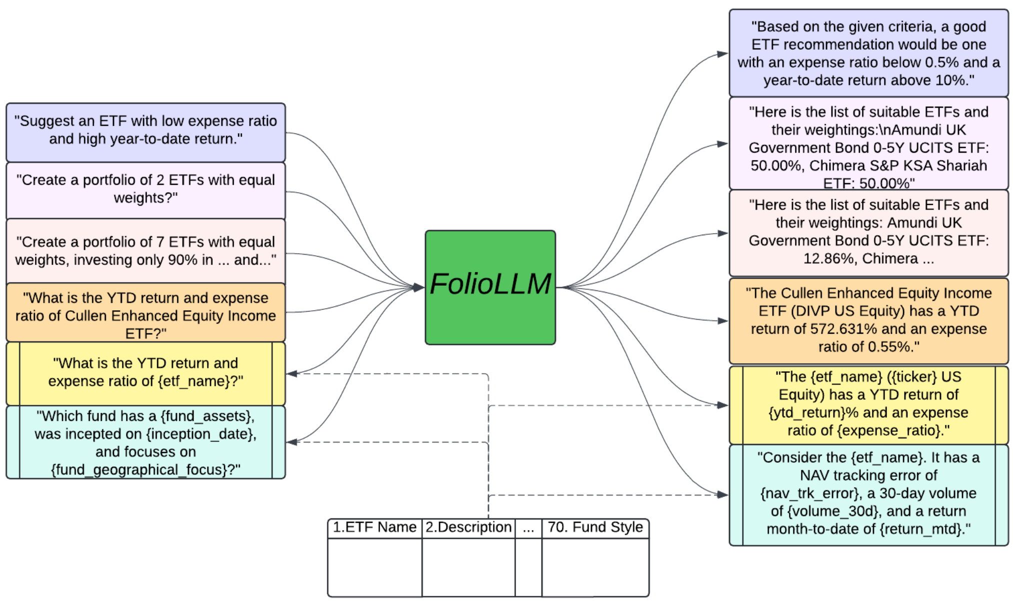


Figure 1: Prompt-Response training pairs.

Fine-Tuning Approaches

Classical Fine-Tuning We started by fine-tuning the entire model on our dataset of ETF data. This allowed the model to update all of its parameters to better fit the target domain.

LoRA We also experimented with the Low-Rank Adaptation (LoRA) technique (Hu et al., 2021). LoRA injects trainable low-rank decomposition matrices into the model, allowing efficient fine-tuning without updating all of the pre-trained weights. We tried different rank values r for the LoRA matrices, hypothesizing that a low rank may be sufficient to capture the important features for our ETF-focused task.

Modified LoRA with KAN Building on the LoRA approach, we proposed a modified version where we replace one of the LoRA matrices (either A or B) with a KAN (Kolmogorov-Arnold Network) layer (Liu et al., 2024). The KAN layer is designed to better extract nuanced relationships between the most important features, which we believed could be beneficial for modeling the complex interactions in ETF data. For more details please refer to the Appendix 4.

For KAN, we mainly focused on experiments with the ReLU kernel, where ϕ is defined as $\mathrm{ReLU}(x)$. The mathematical formulation for this modified LoRA approach is:

$$\Delta W = \alpha \cdot AB$$

where W_s represents the pre-trained weights, α is a scaling factor, B is the LoRA matrix, and A is the KAN layer.
The KAN layer is formulated as follows:

$$KAN(x) = (\Phi_3 \circ \Phi_2 \circ \Phi_1)(x)$$

where $\Phi_q = \sum_{p=1}^{n} \phi_{q,p} \rho(x_p)$ and $\phi_{q,p} = \mathrm{ReLU}(x)$.

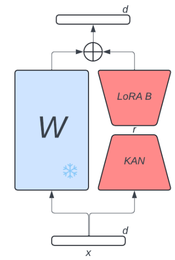


Figure 2: LoRA-KAN Combination.

Using RAG and Classification Techniques

In our project, we employ a combination of Retrieval-Augmented Generation (RAG) and classification techniques to enhance the context for inference and to extract ETF tickers for invoking classical portfolio optimization routines.

Enhancing Context for Inference

To improve the model's ability to provide accurate and insightful responses, we utilize RAG. The model leverages a pre-trained sentence transformer (Reimers & Gurevych, 2019) to generate embeddings for the user queries. These embeddings are then used to retrieve relevant ETF information from a FAISS index (Johnson et al., 2019, Douze et al., 2024). By incorporating this relevant information, the model can generate responses that are contextually enriched and more informative.

The process is as follows:

1. The user's input query is encoded using the SentenceTransformer model to obtain query embeddings.
2. These embeddings are used to perform a search on the FAISS index to retrieve the most relevant ETF descriptions.
3. The retrieved ETF information is then included in the context provided to the model for generating responses.

This approach ensures that the generated responses are grounded in up-to-date and pertinent ETF data, thereby enhancing the overall inference process.

Extracting ETF Tickers for Portfolio Optimization

For portfolio optimization, the system employs a classification technique to determine whether the user's query requires an optimization routine. A linear layer model, trained for sequence classification, predicts the query's category (Devlin et al., 2018). If the query is classified as one requiring optimization, the following steps are executed:

1. The query is encoded and the relevant ETF tickers are extracted from the FAISS index.
2. The extracted tickers are used to invoke a classical portfolio optimization routine, implemented as a custom optimizer module (Markowitz, 1952). Please refer to the Appendix 3 for the details of Mean-Variance Optimization.
3. The optimization results, which include the recommended ETF allocations, are then integrated into the response provided to the user.

By effectively classifying queries and extracting relevant ETF tickers, the system can seamlessly integrate advanced optimization techniques into its responses, offering users actionable investment advice based on their input.

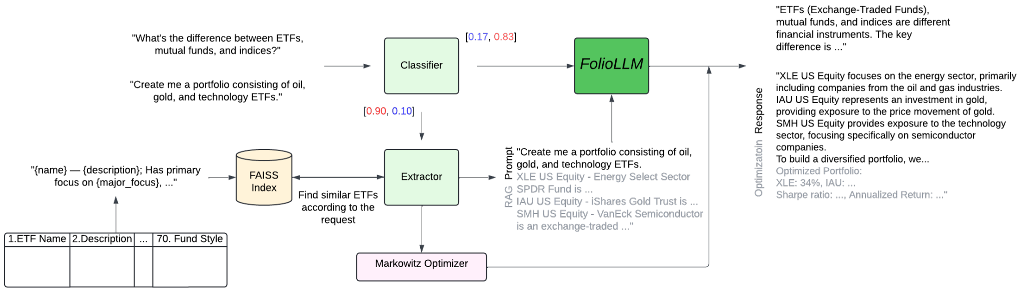


Figure 3: Inference: Integration of RAG, classification, and ETF ticker extraction for portfolio optimization.

5 Experiments

5.1 Data

Hoffmann et al., 2022 highlighted that the dataset plays a more crucial role than model size. Using financial data providers (Bloomberg, Refinitiv, AllFunds) we have gathered quantitative and qualitative data on 12,224 Exchange-Traded Funds (ETFs) with 70 fields, split between qualitative (including short-fund description) and quantitative data. We have performed data clean-up, removing fields which contains significant amount of unavailable data to enhance fine-tuning procedure. In order to provide data in a suitable format, we have run a code to convert available tabular dataset into the json format.

Feature	**Details**
Ticker	SPY US
Name	SPDR S&P 500 ETF Trust
Description	SPDR S&P 500 ETF Trust is an exchange-traded fund incorporated in the USA. The ETF tracks the S&P 500 Index...
Type	ETF
Manager	State Street Corp
...	...
Class Assets, mm	514,613.75
Expense Ratio	0.09%
Year-To-Date Return	9.92%

Table 1: Example Fund Data

Price history: In addition to the quantitative and qualitative data, for every fund we have obtained price history for all the funds mentioned above. This data is utilized to compute financial metrics and resolve optimization portfolio construction task described in Markowitz, 1952.

We created a set of **Prompt/Response pairs** aimed at evaluating the basic competency of our model in understanding and generating financial information. These test prompts, were not seen during the training phase and serve to assess the model's domain-specific performance. We have also designed complex prompt-response pairs, aiming at training the model to give list of ETF tickers and their portfolio allocations. Please refer to Appendix 2 for examples of base and advanced prompt-response pairs.

5.2 Evaluation method

We employed several evaluation metrics to comprehensively assess the performance of FolioLLM:

Language Model Metrics

- **BERT Score:** Measures the similarity between generated text and reference text using BERT embeddings, providing precision, recall, and F1 scores.
- **Cosine Similarity:** Evaluates the semantic similarity between generated and reference texts.
- **Perplexity:** Indicates how well the model predicts a sample, with lower perplexity suggesting better performance.

Finance-Specific Metrics

In addition to the traditional metrics, we employed the following financial metrics to evaluate quality of LLM responses:

Annualized Return measures the yearly return of the portfolio:

$$\text{Annualized Return} = (1 + R_{daily})^{252} - 1$$

where R_{daily} is the average daily return of the portfolio.

Annualized Volatility measures the yearly risk or volatility of the portfolio:

$$\text{Annualized Volatility} = \sigma_{daily}\sqrt{252}$$

where σ_{daily} is the standard deviation of daily returns.

Harmonic Portfolio Symmetry (HPS) quantifies the alignment between the sum of individual ETF assets and a predetermined total across multiple portfolios:

$$\text{HPS} = \frac{1}{M}\sum_{j=1}^{M}\left|T_j - \sum_{i=1}^{n_j} w_{ij}\right|$$

where M is the number of modeled portfolios, T_j is the specified total for the j-th portfolio, w_{ij} is the weight of the i-th ETF in the j-th portfolio, and n_j is the number of ETFs in the j-th portfolio.

Sharpe Ratio evaluates portfolio performance by comparing the excess return over the risk-free rate to its standard deviation:

$$\text{Sharpe Ratio} = \frac{E[R_{model} - R_f]}{\sigma_{model}}$$

where R_{model} are returns of the portfolios, R_f is the risk-free rate, assumed constant, and σ_{model} is the standard deviation of the portfolios' returns. Explanation:

5.3 Experimental details

The experiments were conducted using the following setup:

- **Model Configuration:** We fine-tuned the FinGU model with 0.5B parameters using a combination of synthetic and real ETF data.
- **Training Environment:** Training was performed on NVIDIA A100/H100 GPUs using the Hugging Face Trainer API integrated with custom tokenization and data collators on the Modal infrastructure.
- **Hyperparameters:** Learning rate was set to 2×10^{-5}, with a batch size of 1 per device, and training was conducted for 3 epochs.
- **Fine-Tuning:** We experimented with classical fine-tuning and LoRA with different ranks as well as the modified LoRA-KAN approach discussed earlier. Prompt tuning techniques were employed, and advanced configurations such as gradient checkpointing and data augmentation via templates generated by GPT-4 and Claude were applied.

In order to monitor training process, we have utilized weights and biases library. The training process have demonstrated significant reduction of a gradient at the initial epochs, followed by the period of minimal gradient changes,

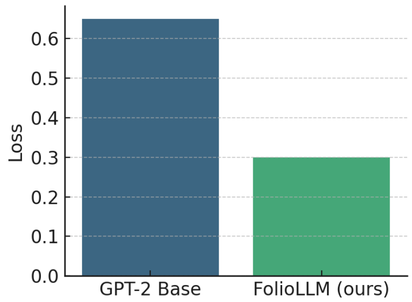


Figure 4: Loss comparison between models

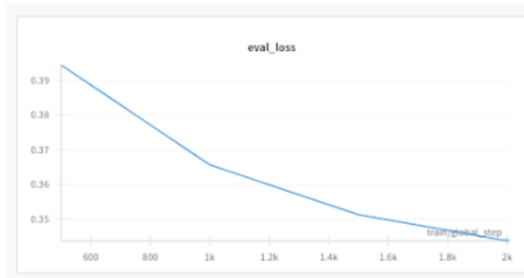


Figure 5: FolioLLM (KAN-LoRA) eval loss over epochs

5.4 Results

The quantitative results of our experiments are summarized in Table 2. The evaluation metrics include BERT Score, Cosine Similarity, and Perplexity, comparing the performance of the base GPT-2 model, the FinGU model, and the fine-tuned FolioLLM model. We have also run a test set of portfolio construction questions and reviewed suggested model portfolios based on the above financial metrics to understand adequacy of financial advice produced by the model.

Model	BERT Recall ↑	Perplexity ↓	Cosine Similarity ↑
GPT-2 Base	0.8472	11.2344	0.2619
FinGU	0.8695	6.1734	0.2842
FolioLLM (ours)	**0.8703**	4.0720	**0.3245**
FolioLLM (KAN-LoRA)	0.8269	**2.5271**	0.1227

Table 2: Comparison of different models.

Model	Avg Return ↑	Avg Volatility ↓	Avg Sharpe Ratio ↑	HPS ↓
FinGU	N/A[1]	N/A[1]	N/A[1]	N/A[1]
GPT 3.5	8.15%	14.45%	0.535	**0.087**
FolioLLM (ours)	**31.19%**	**13.50%**	**1.948**	0.115

Table 3: Financial Results of Response Portfolios

6 Analysis

While the generic NLP metrics such as BERT Score, Cosine Similarity, and Perplexity proved useful in evaluating the overall performance of our models, a more nuanced evaluation of the domain-specific aspects could only be achieved through human interaction and qualitative analysis. During our qualitative review, we observed very distinct impacts based on the fine-tuning approach employed.

For instance, the KAN-LoRA method with a relatively dense KAN network yielded more concise responses and improved awareness of different ETFs' characteristics such as geography and sector compared to the standard LoRA approach. However, this method also had some drawbacks. For portfolio prompts without any LoRA, the model provided detailed information about portfolio composition, but often failed to recognize requests for different portfolio weights, pulling in a lot of linked information without tailoring it to the specific request. This sometimes resulted in responses containing irrelevant information and unexpected characters.

With the simple LoRA approach, the responses improved notably, but the additional information about asset weights and portfolio composition disappeared. When using prompt-based LoRA, the responses contained some additional information and, in certain cases, portfolio allocations, but often included inaccurate asset allocations, imprecise fund tickers, and occasional unexpected characters.

[1] FinGU was unable to produce any results

The LoRA-KAN approach significantly limited the amount of output data, often responding only with fund tickers. However, it produced very accurate responses for certain types of requests, such as those focused on geographical or specific asset class funds. Due to limited time and resources, we could not comprehensively analyze different options with varying KAN sizes and mixtures of kernels, or experiment with replacing LoRA B instead of LoRA A. These areas present opportunities for further exploration in future work.

Analysing financial metrics, one can see that the original FinGU model was incapable of producing portfolios of specific ETFs. On the other side, GPT-3.5 demonstrated good understanding of the requested number of funds and their weights in the portfolio. However, due to the absence of financial data and optimization techniques, it primarily was producing portfolios of most widely-recognized ETFs in the market (e.g. SPY, QQQ). FolioLLM was capable of leveraging training pair and datasets to identify niche ETFs matching the requested criterias and by applying optimization produced portfolios with higher return and lower volatilities, producing better Sharpe ratio (see Appendix 5).

7 Conclusion

In this project, we have fine-tuned our FolioLLM model and explored several novel approaches, including the utilization of KAN (Kolmogorov-Arnold Network) in combination with LoRA (Low-Rank Adaptation). While the performance of these techniques showed promise in our domain-specific problem, we do not categorically claim they are superior to other fine-tuning approaches.

By choosing a relatively small model size, we were able to generate coherent overviews and optimization suggestions. In a short time-frame, we created a synthesized dataset that allowed us to achieve high evaluation metrics, surpassing the performance of comparable models trained on higher-quality datasets. This suggests that our data augmentation and fine-tuning strategies were effective in adapting the model to the ETF and portfolio management domain.

Looking ahead, we believe this model is capable of achieving even greater results with a larger dataset and higher-quality inputs. The nuanced impact we observed based on the fine-tuning approach indicates that further exploration of KAN configurations, as well as experiments with replacing LoRA B instead of LoRA A, could present opportunities for continued improvement.

Overall, the promising results of our FolioLLM model demonstrate the potential of large language models in the financial domain, particularly for automating and personalizing the portfolio construction process.

8 Ethical Considerations

Machine learning methods are widely used in the financial industry, yet most professionals rely on their implementation, leaving decision-making to trusted experts. Despite this, there are numerous instances of erroneous investment decisions by experts, such as the collapse of Archegos due to inadequate assetallocation and the subsequent downfall of Credit Suisse, causing significant damage to investors globally (Credit Suisse Group Special Committe, 2021). AI-produced investments require even greater scrutiny to avoid similar catastrophic mistakes.

Moreover, AI investment advisers might disproportionately affect individuals with smaller portfolios and less investment expertise. These individuals rely heavily on the accuracy and suitability of AI-generated advice. Any material errors or failure to consider personal circumstances could result in financial losses. Solowey, 2023 highlights issues related to the application of agency law in LLMs and the liability for negative outcomes produced by advisory LLM models, stressing the importance of true alignment with human interests, a fiduciary duty for human advisors. Despite those challenges, there several ways to mitigate them. Solowey, 2023 has suggested creating a recovery funds for LLM-powered financial applications to compensate for potential losses or allow users in certain circumstances to revert decisions made by LLM (such as reverting trades executed by AI). In addition, there are ways to adhere strict quality controls and thus limit liability of developers and limiting usage of AI-powered investment tools to category of qualified finance professionals.

Additionally, the introduction of AI tools could significantly impact job prospects in financial industry. The widespread use of such tools could lead to cuts among portfolio managers, analysts, and personal financial advisers, fundamentally altering the industry.

References

Araci, D. (2019). Finbert: Financial sentiment analysis with pre-trained language models.

Cascio, M. (2023). How to create an investment portfolio strategy with chatgpt. https://medium.datadriveninvestor.com/how-to-create-an-investment-portfolio-strategy-with-chatgpt-4c55b249f894

Credit Suisse Group Special Committe. (2021, July). *Credit suisse group special committee of the board of directors report on archegos capital management* (tech. rep.). https://www.credit-suisse.com/media/assets/corporate/docs/about-us/investor-relations/financial-disclosures/results/csg-special-committee-bod-report-archegos.pdf

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Douze, M., Guzhva, A., Deng, C., Johnson, J., Szilvasy, G., Mazaré, P.-E., Lomeli, M., Hosseini, L., & Jégou, H. (2024). The faiss library.

Guo, Z., Jiang, G., Zhang, Z., Li, P., Wang, Z., & Wang, Y. (2023). Shai: A large language model for asset management.

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., de Las Casas, D., Hendricks, L. A., Welbl, J., Clark, A., Hennigan, T., Noland, E., Millican, K., van den Driessche, G., Damoc, B., Guy, A., Osindero, S., Simonyan, K., Elsen, E., … Sifre, L. (2022). Training compute-optimal large language models.

Hu, H., Singh, A., Najibi, M., Girshick, R., Malik, J., & Dollar, P. (2021). Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Johnson, J., Douze, M., & Jégou, H. (2019). Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, *7*(3), 535–547.

Liu, Z., Wang, Y., Vaidya, S., Ruehle, F., Halverson, J., Soljačić, M., Hou, T. Y., & Tegmark, M. (2024). Kan: Kolmogorov-arnold networks.

López de Prado, M. M. (2018). *Advances in financial machine learning*. Wiley. https://books.google.co.uk/books?id=c75IDwAAQBAJ

López de Prado, M. M. (2020). *Machine learning for asset managers*. Cambridge University Press. https://doi.org/10.1017/9781108883658

Markowitz, H. (1952). Portfolio Selection. *The Journal of Finance*, *7*(1), 77–91. https://doi.org/10.2307/2975974

Reimers, N., & Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Solowey, J. (2023, May). *Ai will revolutionise financial advice*. https://www.cato.org/commentary/ai-will-revolutionise-financial-advice

Van Cappelle, T. (2023). There is Alpha in Large Language Models. *CFA Society Netherlands*, *21*(155), 18–25.

Wu, S., Irsoy, O., Lu, S., Dabravolski, V., Dredze, M., Gehrmann, S., Kambadur, P., Rosenberg, D., & Mann, G. (2023). Bloomberggpt: A large language model for finance.

Yang, H., Liu, X.-Y., & Wang, C. D. (2023). Fingpt: Open-source financial large language models.

Zhao, H., Liu, Z., Wu, Z., Li, Y., Yang, T., Shu, P., Xu, S., Dai, H., Zhao, L., Mai, G., Liu, N., & Liu, T. (2024). Revolutionizing finance with llms: An overview of applications and insights.

A Appendix 1: Information Prompt Responses

Prompt: How has the DIVP ETF performed this year, and what are the fees associated with it? **Base Model Response:** *"The DIVP ETF has been in the news for some time now. Last year, the DIVP ETF was the best performing ETF in the world, with a market capitalization of $13 billion. The DIVP ETF has been up by more than 20% in the past year, and has outperformed the DIVP ETF by more than 20% in the past year. The DIVP ETF is also up over 15% in the past year."* **Issues:** Repetitive, inaccurate, and does not provide fees.

Fine-tuned Model Response: *"The DIVP ETF has a cap at 0% of total assets and at least 80% of total liabilities. It is designed to track the performance of the S&P 500 Index. The ETF invests in equity securities, which are exchangeable, that correspond to the total daily performance of the S&P 500 Index."* **Improvement:** Provides some structured information, though it still misses specific performance and fee details.

Prompt: Could you summarize the key characteristics of the DIVP ETF for me, please? **Base Model Response:** *"DIVP ETF: DIVP ETF is the first investment product that can be used to hedge the price of a cryptocurrency. DIVP ETF is designed to be a hedge against the cost of an asset in the cryptocurrency. DIVP ETF is not a speculative asset, as it is not backed by any assets or liabilities."* **Issues:** Incorrect and irrelevant information about cryptocurrency.

Fine-tuned Model Response: *"First, it is an ETF. In this sense, it is a direct investment fund. In the short term, the ETF seeks to provide investment returns that closely correspond to the return of the DIVP Index. In the long term, the ETF seeks to provide investment returns that closely correspond to the return of the DIVP Index."* **Improvement:** Provides relevant information about the investment strategy, though it could be more detailed.

Prompt: I'm interested in the DIVP ETF. What's its investment strategy and does it pay out income? **Base Model Response:** *"I think that if you're a DIVP ETF investor, you have a lot of options. You can buy a DIVP ETF, and if you get lucky, you can buy stocks and bonds, and that's not really a huge amount of money. But you need to have a lot of options. You need a lot of options, and you need a lot of options."* **Issues:** Rambling, incoherent, and does not address the question.

Fine-tuned Model Response: *"First of all, the DIVP ETF is a DIVP fund. In order to invest in DIVP ETF, the ETF must have at least 10% of total assets and at least 80% of non-cash assets. The ETF's investment objective is to achieve return that closely replicates the performance of the Russell 2000 Index. The ETF invests in a portfolio of securities that generally exhibit similar performance characteristics to the Russell 2000 Index."* **Improvement:** Provides structured information on investment strategy, but does not mention income payout.

From these comparisons, it is evident that the fine-tuned model generally provides more structured and contextually appropriate responses. Although specific details are sometimes still missing, the improvements in coherence and relevance make the fine-tuned model more suitable for generating ETF-related content. These results are promising and suggest that the fine-tuning and data augmentation techniques have effectively enhanced the model's ability to generate accurate and relevant financial advice.

B Appendix 2: Examples of Prompt-Response Training Templates

ETF Descriptive Dataset

```
{
        ETF Details:
        Ticker: SPY
        Bloomberg Ticker: SPY US Equity
        Name: SPDR S&P 500 ETF Trust
        Description: SPDR S&P 500 ETF Trust is an exchange-
            traded fund incorporated in the USA. The ETF
            tracks the S&P 500 Index. The Trust consists of a
            portfolio representing all 500 stocks in the S&P
            500 Index. It holds predominantly large-cap U.S.
            stocks. This ETF is structured as a Unit
            Investment Trust and pays dividends on a quarterly
             basis. The holdings are weighted by market
            capitalization.
        Type: ETF
        Manager: State Street Corp"
        ...
}
```

Prompt-Response Pairs

Descriptive ETF Training Sample Pair:

```
{
        "prompt": "What are some ETFs that provide exposure to
             emerging markets?",
        "expected_answer": "Some ETFs that provide exposure to
             emerging markets include VWO (Vanguard FTSE
            Emerging Markets ETF), EEM (iShares MSCI Emerging
            Markets ETF), and IEMG (iShares Core MSCI Emerging
             Markets ETF)."
}
```

Portfolio Construction Training Sample Pair:

```
{
        "prompt": "Create a portfolio of 7 ETFs with equal
            weights?",
        "response": "Here is the list of suitable ETFs and
            their weightings:\nAmundi UK Government Bond 0-5Y
            UCITS ETF: 14.29%, Chimera S&P KSA Shariah ETF:
            14.29%, BCC Global ETN: 14.29%, SBER Dynamic Smart
             Fund: 14.29%, E FundSI Hong Kong Securities
            Investment Theme ETF: 14.29%, Samsung KODEX Active
             Korea Total Bond Market AA- ETF: 14.29%, UBS ETC
            2x Short CMCI WTI Crude Oil Index - 3 Month:
            14.29%, "
}
```

C Appendix 3: Mean-Variance Optimization

Introduce in Markowitz, 1952 mean-variance portfolio optimization, is a mathematical framework for assembling a portfolio of assets such that the expected return is maximized for a given level of risk. This approach is based on the idea that investors are risk-averse and seek to achieve the best possible return for a given level of risk.

Consider a portfolio of n assets. Let $\mathbf{w} = (w_1, w_2, \ldots, w_n)^T$ be the vector of weights, where w_i represents the proportion of the total portfolio value invested in asset i.

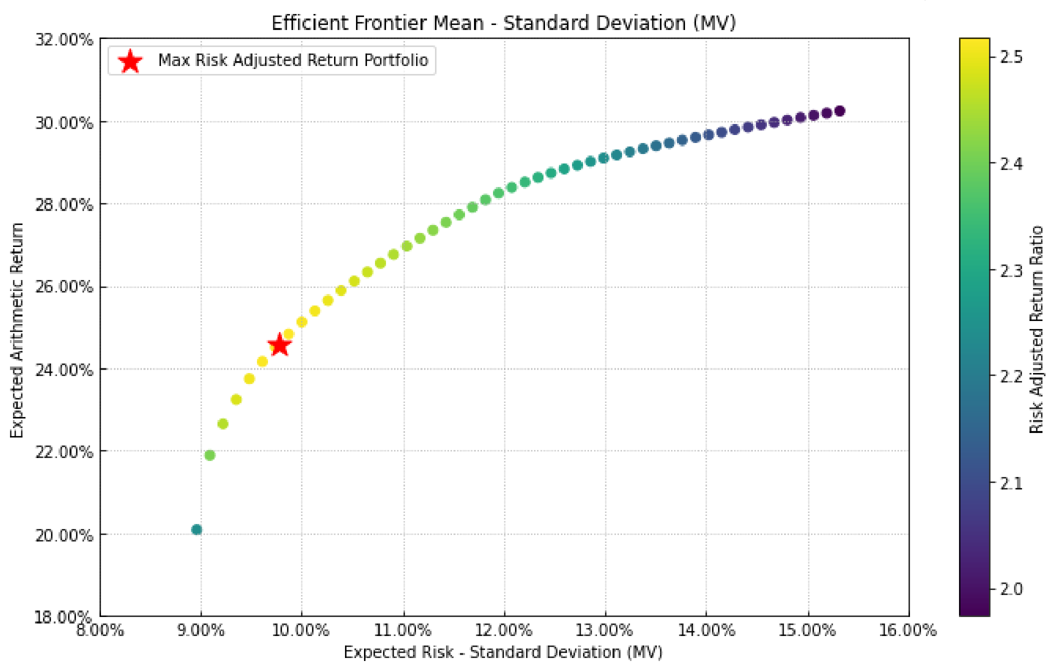
- **Expected Return:** The expected return of the portfolio, μ_p, is given by $\mu_p = \mathbf{w}^T \mu$, where $\mu = (\mu_1, \mu_2, \ldots, \mu_n)^T$ is the vector of expected returns for each asset.
- **Portfolio Variance:** The risk (variance) of the portfolio, σ_p^2, is given by $\sigma_p^2 = \mathbf{w}^T \mathbf{\Sigma} \mathbf{w}$, where $\mathbf{\Sigma}$ is the $n \times n$ covariance matrix of asset returns.

Different optimization criteria can be used based on the investor's objective. Here we consider three criteria: minimum variance, maximum return, and maximum Sharpe ratio.

- **Minimum Variance:** The objective is to minimize the portfolio variance: $\min_{\mathbf{w}} \mathbf{w}^T \mathbf{\Sigma} \mathbf{w} \quad \text{subject to} \quad \mathbf{1}^T \mathbf{w} = 1, \quad \mathbf{w} \geq 0$
- **Maximum Return:** The objective is to maximize the expected return of the portfolio: $\max_{\mathbf{w}} \mathbf{w}^T \mu \quad \text{subject to} \quad \mathbf{1}^T \mathbf{w} = 1, \quad \mathbf{w} \geq 0$
- **Maximum Sharpe Ratio:** The objective is to maximize the Sharpe ratio of the portfolio (implemented in FolioLLM). The Sharpe ratio is defined as the ratio of the portfolio's excess return over the risk-free rate to its standard deviation: $S = \frac{\mathbf{w}^T \mu - r_f}{\sqrt{\mathbf{w}^T \mathbf{\Sigma} \mathbf{w}}}$, where r_f is the risk-free rate. This can be formulated as: $\max_{\mathbf{w}} \frac{\mathbf{w}^T \mu - r_f}{\sqrt{\mathbf{w}^T \mathbf{\Sigma} \mathbf{w}}} \quad \text{subject to} \quad \mathbf{1}^T \mathbf{w} = 1, \quad \mathbf{w} \geq 0$

This is typically solved by converting to a quadratic programming problem. For FolioLLM we have utilized Sequential Least Squares Programming optimizer (SLSQP), suitable for the case where functions are continuously twice differentiable. The solution to the mean-variance optimization problem for various target returns μ_p gives us the efficient frontier, which represents the set of optimal portfolios that offer the highest expected return for a defined level of risk.

Below is the efficient frontier for the FolioLLM ETF universe of 12,224 funds.



D Appendix 4: Motivation for Integrating KAN into LoRA

The motivation behind incorporating the Kolmogorov-Arnold Network (KAN) into the Low-Rank Adaptation (LoRA) approach stems from the desire to better capture the nuanced relationships between the most important features in our ETF dataset. While the standard LoRA technique has demonstrated promising results in fine-tuning large language models, we hypothesized that a more expressive function approximate could further enhance the model's ability to extract relevant patterns from the complex, multi-dimensional ETF data.

The KAN layer, with its hierarchical structure of linear transformations and nonlinearities, is well-suited for modeling intricate feature interactions. By replacing one of the LoRA matrices (either A or B) with a KAN layer, we aimed to leverage the network's representational power to better adapt the pre-trained model to the specific requirements of the ETF and portfolio management domain.

Introducing a moderate-sized KAN layer into the LoRA framework does not incur a significant computational overhead. Our experiments have shown a roughly 25% increase in training times when replacing either the A or B matrix with a KAN layer, compared to the standard LoRA approach. However, replacing both the linear LoRA matrices A and B with KAN layers would make the training process more computationally demanding and potentially compromise one of the most attractive features of LoRA - its efficiency and speed.

Figure D.1 illustrates one of the options for incorporating the KAN layer into the LoRA structure. The specific tensor sizes shown are for demonstration purposes and may vary depending on the final model configuration.

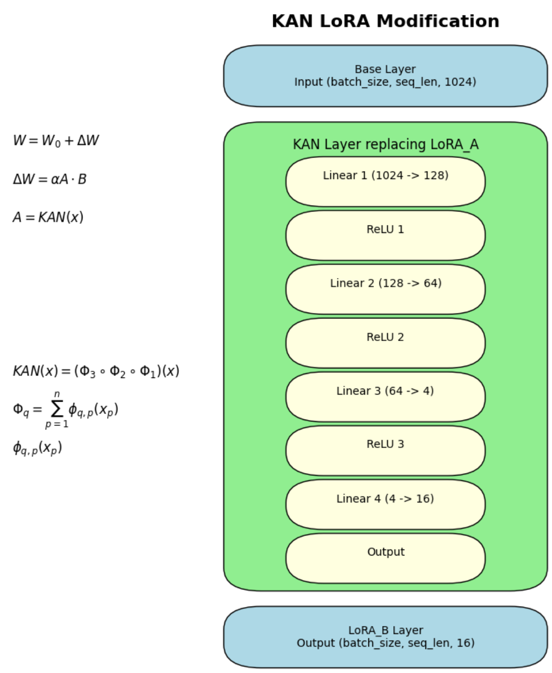


Figure 6: Incorporating a KAN layer into the LoRA modification.

By striking a balance between the expressive power of KAN and the computational efficiency of LoRA, we aimed to develop a fine-tuning approach that can better capture the nuanced relationships in ETF data without sacrificing the scalability and deployability benefits of the LoRA technique.

E Appendix 5: FolioLLM Responses

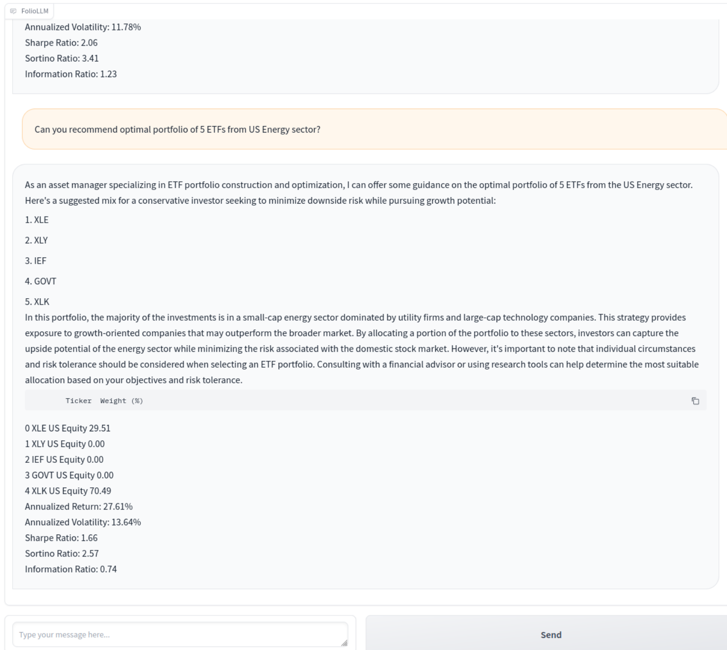


Figure 7: Optimal Portfolio Response - US Energy.

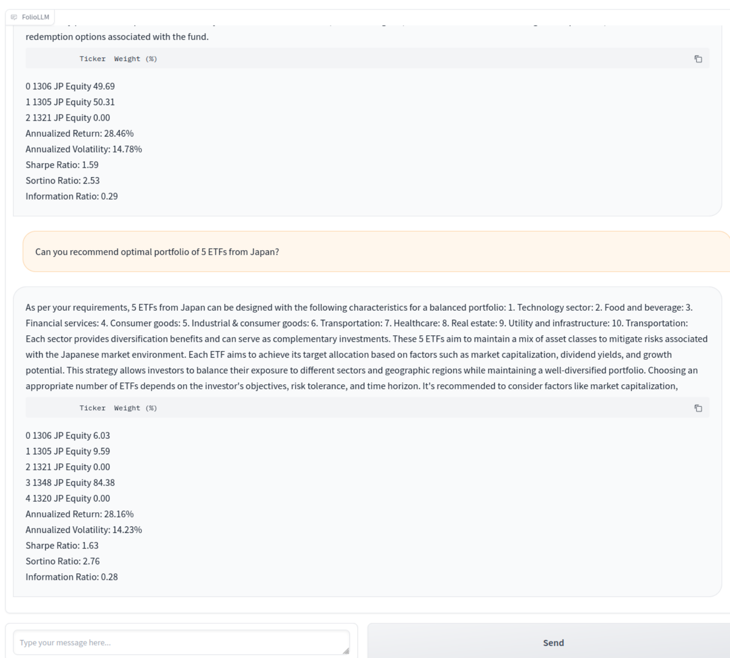


Figure 8: Optimal Portfolio Response - Japan.