Automatic Stock Portfolio Diversification Using β-VAE

Hair Albeiro Parra Barrera
HEC Montreal
11315672
hair-albeiro.parra-barrera@hec.ca

Olivier Makuch
HEC Montreal
11345395
olivier.makuch@hec.ca

Abstract

Portfolio diversification is a cornerstone of financial portfolio management, aiming to mitigate risk through the inclusion of diverse assets. In this work, we propose a novel recommendation system for optimizing portfolio diversification by leveraging embeddings generated through a β-Variational Autoencoder (β-VAE). Our method identifies and replaces highly similar stocks within a portfolio to enhance its diversification ratio, defined as the weighted average of individual asset volatilities relative to the total portfolio risk. Using historical financial data, the β-VAE learns compact and disentangled representations of stock features. Extensive experiments on SP 500 stocks and additional custom tickers demonstrate the efficacy of our approach, achieving significant improvements in diversification ratio compared to traditional Principal Component Analysis (PCA)-based embeddings. The β-VAE embeddings consistently outperformed PCA-based methods, achieving up to 85.43% improvement in diversification ratio for individual portfolios. Additionally, while Sharpe ratio improvements are not guaranteed, notable gains were observed in portfolios with initially poor configurations. This work offers a robust framework for integrating advanced representation learning into portfolio optimization, addressing critical challenges in sector concentration and risk management.

1 Introduction

Effective portfolio diversification is a fundamental principle in financial risk management, aiming to balance returns while minimizing exposure to individual asset volatility. Traditional methods for constructing diversified portfolios often rely on static heuristics and domain expertise, which may fail to adapt dynamically to market changes. Furthermore, the selection of assets to optimize diversification remains an open challenge, particularly when portfolios exhibit high sector concentration or inter-asset correlations.

Recent advancements in machine learning offer promising tools for addressing this challenge. Variational Autoencoders (VAEs), particularly the β-Variational Autoencoder (β-VAE), enable the learning of disentangled and compact latent representations from complex data. These latent embeddings can capture intricate relationships between stocks, including their sectoral and risk characteristics, providing a novel basis for diversification recommendations.

In this work, we develop a recommendation system that uses β-VAE embeddings to optimize portfolio diversification. By iteratively identifying and replacing highly similar stocks in a portfolio, our approach improves the diversification ratio, a key metric quantifying the spread of asset risk and allocation.

The contributions of this paper are as follows:

- We propose a β-VAE-based system for learning stock embeddings, enabling the representation of asset similarities in a low-dimensional space.

- We design an iterative algorithm that identifies suboptimal assets in a portfolio and recommends replacements from a dissimilar candidate pool, thereby improving diversification.

- We evaluate the proposed framework on SP 500 data and custom tickers, demonstrating improvements in diversification ratio compared to PCA-based embeddings, with notable gains in poorly diversified portfolios.

The remainder of this paper is organized as follows: Section 2 reviews related work in portfolio optimization and representation learning. Section 3 describes the β-VAE model, the proposed recommendation algorithm, and the experimental setup. Section 4 details the datasets used for training and evaluation. Section 5 presents experimental results, including diversification improvements and performance comparisons. Finally, Section 5.4 concludes with insights and future directions.

2 Related Work

Portfolio optimization has long been a foundational problem in financial management, with diversification being a key principle to reduce risk and enhance portfolio performance. Traditional methods, such as mean-variance optimization and factor models, often rely on historical returns and linear assumptions, which may fail to capture the complex relationships between assets in modern financial markets. To address these limitations, researchers have increasingly turned to advanced statistical and machine learning approaches.

Principal Component Analysis (PCA) is a widely used statistical technique in finance for dimensionality reduction and identifying underlying factors that drive asset returns. By transforming correlated variables into a set of uncorrelated components, PCA helps in understanding the primary sources of risk and return in a portfolio. For instance, Tan (2012) demonstrated that using PCA to estimate the covariance matrix in portfolio optimization can lead to improved portfolio efficiency and reduced transaction costs (1). Similarly, Dhingra et al. (2021) employed PCA to filter dominant financial ratios from each sector, enhancing the process of sectoral portfolio optimization (2).

However, while PCA is effective in capturing linear relationships, it may not fully encapsulate the non-linear dependencies present in financial data. To overcome this, machine learning techniques, particularly Variational Autoencoders (VAEs), have been explored for their ability to learn complex, non-linear representations. The β-Variational Autoencoder (β-VAE), introduced by Higgins et al. (2017), extends the traditional VAE by introducing a tunable β parameter that encourages disentangled and interpretable latent representations (3). This property makes β-VAE particularly suitable for representing assets in financial datasets, where relationships are often non-linear and high-dimensional.

In the domain of stock recommendations, Wijerathne et al. (2024) proposed a deep learning-based hybrid system combining collaborative filtering and content-based filtering. Utilizing BiVAE for user-item interaction modeling and VAE for content similarity, their system generates personalized stock recommendations tailored to user preferences and risk tolerance (4). While their focus is on personalization, the underlying use of VAEs for representation learning provides a foundation for applying similar techniques to portfolio-level optimization.

Our work builds upon these advancements by adapting β-VAE to the specific context of portfolio diversification. Unlike Wijerathne et al., who emphasize user-centric recommendations, we leverage β-VAE embeddings to represent stock characteristics and sectoral relationships for portfolio-level analysis. Additionally, we design an iterative diversification algorithm that uses these embeddings to replace highly similar stocks with dissimilar candidates, directly targeting the challenge of enhancing diversification ratios in financial portfolios.

By combining the strengths of PCA in capturing linear relationships and β-VAE's capability in learning disentangled non-linear representations, our approach addresses key limitations of traditional portfolio optimization methods. This integration offers a robust framework for dynamic, data-driven asset selection, aiming to improve diversification and, consequently, portfolio performance.

3 Methodology

3.1 Data Collection and Feature Engineering

The dataset construction process begins by scraping the list of S&P 500 companies, including their ticker symbols, names, and GICS sectors, from Wikipedia. Custom tickers can also be appended to the scraped list to extend the set of securities considered. Using the compiled list of tickers, historical stock data is fetched from the Yahoo Finance API for a one-year period at weekly intervals. Key financial metrics are computed for each stock, including mean open and close prices, high and low prices, last closing price, 52-Week high and low values, annualize volatility and last year's return rate.

This results in a comprehensive dataset $\mathcal{D} = \{\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,m})\}_{i=1}^{n}$, where n is the number of stocks and m is the number of features.

Returns for each stock are calculated as weekly simple returns:

$$r_t = \frac{p_t - p_{t-1}}{p_{t-1}},$$

where p_t is the price at time t. Stocks with inconsistent return lengths are discarded, and the dataset is pruned to retain only those with valid return data. Sector and industry classifications are encoded into numerical form using one-hot encoding, ensuring compatibility with machine learning models. The cleaned and structured dataset is then saved for use in training a β-Variational Autoencoder (β-VAE).

3.2 Latent Representation Learning with β-VAE

The β-Variational Autoencoder is employed to learn compact and interpretable embeddings of the stock features. The β-VAE model consists of two primary components: an encoder and a decoder. The encoder maps high-dimensional input features into a lower-dimensional latent space, producing a mean $\boldsymbol{\mu}$ and variance $\mathbf{log}\,\boldsymbol{\sigma}^{\mathbf{2}}$. The decoder reconstructs the original features from this latent representation.

Given an input vector $\mathbf{x}$, the encoder produces:

$$\boldsymbol{\mu}, \mathbf{log}\,\boldsymbol{\sigma}^{\mathbf{2}} = \text{Encoder}(\mathbf{x}),$$

and the latent embedding $\mathbf{z}$ is obtained via the reparameterization trick:

$$\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\epsilon} \cdot \exp\left(0.5 \cdot \mathbf{log}\,\boldsymbol{\sigma}^{\mathbf{2}}\right), \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

The decoder then reconstructs the input as:

$$\hat{\mathbf{x}} = \text{Decoder}(\mathbf{z}).$$

The loss function combines a reconstruction loss, measured as the mean squared error (MSE), and a Kullback-Leibler (KL) divergence term to regularize the latent space:

$$\mathcal{L} = \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 + \beta \cdot \text{KL}\left(\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2) \,\|\, \mathcal{N}(\mathbf{0}, \mathbf{I})\right),$$

where β controls the balance between reconstruction accuracy and latent space regularization.

The β-VAE is trained using the prepared dataset, and the learned embeddings are subsequently used to compute pairwise similarities between stocks in the portfolio recommendation algorithm.

3.3 Portfolio Diversification with β-VAE Embeddings

Portfolio diversification is a fundamental objective in financial management, aiming to minimize risk while maintaining returns. To quantify diversification, the diversification ratio (DR) is defined as:

$$\text{DR}(\mathcal{P}) = \frac{\sum_{i \in \mathcal{P}} w_i \sigma_i}{\sigma_{\mathcal{P}}},$$

where w_i is the weight of asset i in portfolio $\mathcal{P}$, σ_i is the volatility of asset i, and $\sigma_{\mathcal{P}}$ is the portfolio volatility.

The β-VAE embeddings are utilized to optimize portfolio diversification by iteratively identifying and replacing the most similar assets in the portfolio with candidates that are dissimilar in the learned latent space. The process follows these key steps:

1. **Identifying the Most Similar Pair:** Compute pairwise distances between assets in the portfolio based on their embeddings:
$$\text{Dist}(\mathbf{z}_i, \mathbf{z}_j) = \|\mathbf{z}_i - \mathbf{z}_j\|_{\text{dist}}, \quad \forall i, j \in \mathcal{P},\ i \neq j,$$
where $\|\cdot\|_{\text{dist}}$ represents the chosen distance metric (e.g., Euclidean or cosine).
2. **Selecting a Replacement Candidate:** From the set of all stocks outside the portfolio, identify N candidate replacements that are most dissimilar to the selected asset in the latent space:
$$\text{Dist}(\mathbf{z}_r, \mathbf{z}_k), \quad \mathbf{z}_k \in \mathcal{E}_{\text{all}} \setminus \mathcal{E}_{\mathcal{P}},$$
where $\mathbf{z}_r$ is the embedding of the asset selected for replacement.
3. **Evaluating the Diversification Ratio:** Temporarily create an updated portfolio $\mathcal{P}'$ by replacing the selected asset with a candidate, then calculate its diversification ratio:
$$\text{DR}(\mathcal{P}') = \frac{\sum_{i \in \mathcal{P}'} w_i \sigma_i}{\sigma_{\mathcal{P}'}}.$$
4. **Accepting or Rejecting the Swap:** If the diversification ratio of $\mathcal{P}'$ exceeds that of the current portfolio $\mathcal{P}$, accept the swap; otherwise, retain the original portfolio.

This iterative process continues until the diversification ratio converges, the maximum number of iterations is reached, or no further improvements are observed. The algorithm maintains a log of replacements and the history of diversification improvements. Full pseudocode for the β-VAE Recommendation Algorithm is provided in Appendix A.

3.4 Experimental Setup for Diversification

The portfolio diversification framework is evaluated using portfolios constructed from S&P 500 stocks and additional custom tickers. To ensure a robust comparison of methods, 20 random portfolios are generated, each initialized with 10 tickers. These portfolios serve as the baseline for applying diversification algorithms based on three embedding approaches: β-VAE embeddings, PCA embeddings with the same latent dimension as the β-VAE (PCA Latent Dim), and PCA embeddings preserving 90% of the data variance (PCA 90% Var).

The experimental process is as follows:

1. **Initialization:** Generate a random portfolio of 10 tickers and compute its initial diversification ratio ($\text{DR}_{\text{initial}}$).
2. **Optimization:** Apply the diversification algorithm for each embedding method:
   (a) Compute pairwise distances among portfolio assets based on the selected embedding method.
   (b) Replace the most similar asset pair with a dissimilar candidate from the pool of available stocks.
   (c) Recalculate the diversification ratio (DR) for the updated portfolio.
3. **Convergence:** Repeat the optimization steps until the diversification ratio converges or the maximum number of iterations is reached ($K = 250$).

Performance is assessed using the following metrics:

- **Diversification Ratio Improvement ($\textbf{DR}_{\textbf{final}} - \textbf{DR}_{\textbf{initial}}$):** Measures the increase in portfolio diversification for each method.
- **Sharpe Ratio Improvement ($\textbf{SR}_{\textbf{final}} - \textbf{SR}_{\textbf{initial}}$):** Evaluates risk-adjusted returns, where $\text{SR} = \frac{\mu_{\mathcal{P}} - r_f}{\sigma_{\mathcal{P}}}$. Here, $\mu_{\mathcal{P}}$ is the portfolio's mean return, $\sigma_{\mathcal{P}}$ is its volatility, and $r_f = 0\%$ is the assumed risk-free rate.
- **Comparison Across Methods:** Both average and individual portfolio performances are compared across β-VAE, PCA Latent Dim, and PCA 90% Var embeddings, highlighting their respective strengths and limitations in diversification optimization.

By evaluating portfolios across these three embedding methods and using a consistent set of 20 initial portfolios, we aim to comprehensively assess the efficacy of β-VAE embeddings relative to PCA-based alternatives. Experimental results, including quantitative improvements in diversification ratio and Sharpe ratio as well as qualitative changes in portfolio composition, are presented in Section 5.

4 Dataset

4.1 Dataset Description

The primary dataset for this project is constructed from three key sources, constituing a collection of 521 tickers and a number of features based on:

- The S&P 500 company list, scraped from Wikipedia, providing information on ticker symbols, company names, and GICS sectors.
- Historical stock data, fetched from Yahoo Finance, including financial metrics and one-year return vectors for each stock.
- Custom tickers, appended to the S&P 500 tickers to broaden the universe of securities.

The data pipeline generates multiple processed datasets on the same tickers:

- **S&P 500 Dataframe**: Contains basic information about S&P 500 companies.
- **Stock Dataframe**: Aggregates raw financial metrics and sector/industry classifications.
- **Processed Stock Data for VAE**: A fully cleaned and one-hot encoded version of the stock dataframe, ready for input into the β-VAE model.
- **Return Vectors**: A dictionary of weekly return vectors for all stocks, used for portfolio optimization.

4.1.1 Feature Descriptions

The primary datasets include detailed financial and categorical information about each stock, such as ticker symbols, company names, and sector/industry classifications based on the GICS framework. Key financial metrics include market capitalization, opening and closing prices, 52-week high and low values, annualized volatility (over one month and one year), yearly dividend rates, and last year's return rates. Additional derived features include one-hot encoded sector and industry classifications to enable compatibility with machine learning models. These features collectively provide a rich dataset for training the β-VAE and calculating portfolio diversification metrics.

4.2 Dataset Analysis

The datasets vary significantly in size and complexity:

- **S&P 500 Dataframe**: Contains 521 rows (one for each S&P 500 company and additional custom tickers) and 3 columns (Symbol, Security, GICS Sector). This serves as the starting point for data collection.
- **Stock Dataframe**: Expands the data to 18 columns by aggregating financial metrics and sector/industry classifications. This dataset is the foundation for feature engineering.
- **Processed Stock Data for VAE**: After encoding sector and industry classifications into one-hot format, the processed dataframe contains 145 columns. Each row represents a single stock, with financial features and one-hot encodings of categorical variables.
- **Return Vectors**: A dictionary with stock tickers as keys and return vectors as values. Each vector represents weekly returns for the past year.

To ensure consistency, stocks with incomplete or invalid data (e.g., inconsistent return lengths) are excluded during preprocessing. The resulting datasets are clean, structured, and suitable for downstream modeling and analysis.

4.3 Dataset Enhancement

Several enhancements are applied to the raw data:

- **Feature Engineering**: Key metrics such as volatility, 52-week high/low, and last year return rate are calculated from raw stock data to enrich the dataset.

- **One-Hot Encoding**: Sector and industry columns are one-hot encoded to create meaningful categorical features for the β-VAE model.
- **KNN Imputation**: Missing values in numerical columns (e.g., Market Cap) are imputed using KNN-based methods, ensuring no missing data remains.
- **Standardization**: Numerical features are normalized using a StandardScaler for compatibility with the β-VAE model.

The final processed dataset serves multiple purposes, including (1) Training the β-VAE to generate embeddings that capture stock similarities, (2) Providing return data for portfolio optimization, and (3) Enabling the calculation of portfolio diversification metrics.

5 Results

This section presents the findings of the experiments conducted using the β-VAE framework and portfolio diversification algorithms. The results are analyzed in terms of diversification ratio (DR) and Sharpe ratio (SR), highlighting the effectiveness of the proposed method.

5.1 Portfolio Diversification Results

Twenty random portfolios were generated, each initialized with 10 stocks. The diversification ratio (DR) and Sharpe ratio (SR) were calculated for each portfolio before and after applying the diversification algorithm using three methods: β-VAE embeddings, PCA embeddings with the same dimension as the β-VAE (Latent Dim), and PCA embeddings preserving 90% variance. Table 1 summarizes the average results across all portfolios.

Method	**Average Initial DR**	**Average Final DR**	**Average DR Improvement (%)**
β-VAE	2.34	**3.13**	**34.98**
PCA (Latent Dim)	2.34	2.73	17.19
PCA (90% Var)	2.34	2.64	13.38

Table 1: Average diversification ratio improvements across all portfolios for different embedding methods. The β-VAE consistently outperforms the PCA-based embeddings in improving DR.

To provide deeper insight, Tables 2, 3, and 4 display the top three portfolios for each method based on DR improvement, including SR improvements for comparison.

Portfolio	**Initial DR**	**Final DR**	**ΔDR (%)**	**Initial SR**	**Final SR**	**ΔSR (%)**
17	2.009	**3.725**	**85.43**	0.264	0.557	110.74
2	2.119	**3.826**	**80.57**	0.216	0.517	139.82
3	2.235	**3.663**	**63.94**	0.378	0.458	20.99

Table 2: Top 3 portfolios by DR improvement using β-VAE embeddings. The β-VAE method achieves significant gains in both DR and SR in these cases.

Portfolio	**Initial DR**	**Final DR**	**ΔDR (%)**	**Initial SR**	**Final SR**	**ΔSR (%)**
12	2.159	**2.928**	**35.61**	0.280	0.418	49.39
19	2.018	**2.706**	**34.08**	0.202	0.338	67.67
18	2.330	**3.052**	**31.01**	0.069	0.485	607.02

Table 3: Top 3 portfolios by DR improvement using PCA (Latent Dim) embeddings. While effective, this method shows smaller gains in DR compared to β-VAE. We observe an extreme improvement in SR, explained by the initial poor (random) choice of portfolio.

Portfolio	Initial DR	Final DR	ΔDR (%)	Initial SR	Final SR	ΔSR (%)
17	2.009	**2.717**	**35.24**	0.264	0.327	23.73
19	2.018	**2.501**	**23.94**	0.202	0.281	39.25
2	2.119	**2.612**	**23.27**	0.216	0.258	19.58

Table 4: Top 3 portfolios by DR improvement using PCA (90% Var) embeddings. The gains in DR are generally smaller than those achieved by β-VAE.

The β-VAE consistently achieves the largest DR improvements, as seen in both the average results and the top-performing portfolios.

5.2 Portfolio Distribution Analysis

To visualize the effect of the β-VAE diversification algorithm, portfolio distributions were analyzed before and after applying both the maximum diversification optimization and the β-VAE diversification. Figure 5 illustrates the changes in portfolio composition for a representative portfolio.

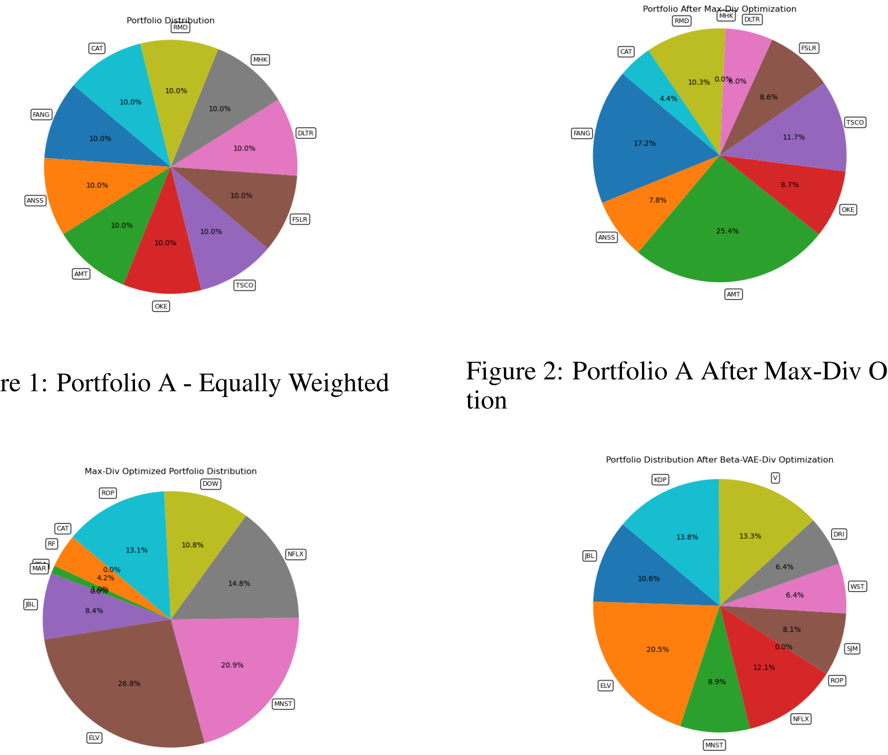

Figure 1: Portfolio A - Equally Weighted

Figure 2: Portfolio A After Max-Div Optimization

Figure 3: Portfolio B - Max-Div Optimized

Figure 4: Portfolio B After β-VAE Diversification

Figure 5: Portfolio distributions before and after applying the max-diversification and β-VAE diversification algorithms. Our algorithm not only updates the weight distribution but suggests swapping some of the portfolio securities themselves, with the goal of increasing teh diversification ratio beyond Max-Div optimization.

5.3 Model Optimization and Performance

The β-VAE model was optimized using random search with the Optuna framework, resulting in a latent dimension of 5, a disentanglement parameter (β) of 5.45, a learning rate of 0.000586, and a batch size of 32. These hyperparameters enabled the model to effectively capture latent stock features, leading to improved embeddings and enhanced portfolio diversification outcomes.

5.4 Summary of Findings

The experiments conducted demonstrate the effectiveness of β-VAE embeddings in enhancing portfolio diversification. The key findings include:

- Significant improvement in diversification ratio (up to **85.43%**).
- Noticeable but extremely variable gains in Sharpe ratio for select portfolios (up to **607.02%** for PCA Latent Dim, and up to **110.74%** for β-VAE).
- β-VAE consistently outperforms PCA-based embeddings in DR improvement, as shown in Tables 2, 3, and 4.

These results underscore the potential of β-VAE embeddings as a tool for constructing robust and diverse investment portfolios. However, SR improvements remain highly variable and are not guaranteed, depending on the portfolio's composition and market conditions.

In this work, we introduced a novel approach for optimizing portfolio diversification using embeddings generated by a β-Variational Autoencoder (β-VAE). The proposed system iteratively enhances the diversification ratio (DR) of portfolios by replacing highly similar stocks with dissimilar candidates identified through learned embeddings. Extensive experiments demonstrated the efficacy of this approach, achieving significant improvements in DR across all tested portfolios and consistent outperformance of PCA-based embedding methods. Notably, the β-VAE achieved an average DR improvement of **34.98%**, with individual portfolio improvements reaching up to **85.43%**. Additionally, while Sharpe ratio (SR) gains were highly variable, select portfolios with initially poor configurations exhibited notable improvements.

The key contributions of this research are as follows:

- A β-VAE-based embedding system that captures complex, non-linear relationships between stocks, enabling meaningful comparisons in a low-dimensional latent space.
- An iterative diversification algorithm that successfully identifies and replaces suboptimal stocks to maximize DR, demonstrating improvements surpassing those achieved by PCA-based methods.
- Validation of the approach through experiments on real-world datasets, including 20 random portfolios constructed from S&P 500 stocks and additional custom tickers, showcasing robust enhancements in diversification and risk-return metrics.

The results highlight the advantages of β-VAE embeddings over traditional PCA-based approaches, particularly in their ability to disentangle sectoral and risk-related features, offering a robust framework for improving portfolio diversification. This work underscores the potential of advanced representation learning techniques for addressing challenges in sector concentration, inter-asset correlations, and risk management.

5.5 Future Work

Future work can focus on incorporating real-time market data and dynamically adapting embeddings to enhance performance under volatile conditions. Expanding the system's objectives to include risk, return, and ESG factors, as well as broadening the dataset to encompass international markets, emerging economies, and alternative assets like cryptocurrencies, could further improve its applicability. Exploring advanced embedding techniques, such as transformers or diffusion models, and comparing them to β-VAE and PCA could provide deeper insights into performance and interpretability trade-offs. Integrating the approach with established frameworks like Black-Litterman and conducting robustness tests under extreme market conditions will ensure adaptability, resilience, and practical utility in diverse financial scenarios.

References

[1] J. Tan, “Principal component analysis and portfolio optimization,” *SSRN Electronic Journal*, 2012.

[2] V. Dhingra, A. Sharma, and S. K. Gupta, “Sectoral portfolio optimization by judicious selection of financial ratios via pca,” *arXiv preprint arXiv:2106.11484*, 2021.

[3] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, “β-vae: Learning basic visual concepts with a constrained variational framework,” in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[4] N. Wijerathne, J. Samarakoon, K. Rathnayake, S. Jayasinghe, S. Ahangama, I. Perera, V. Dhananjaya, and L. Sivanesaharajah, “Deep learning based personalized stock recommender system,” in *ICONIP 2023, CCIS 1966*, pp. 362–374, Springer, 2024.

A Full Algorithm

The full pseudocode for the β-VAE Recommendation Algorithm, which is detailed in Section 3, is presented below:

Algorithm 1 Portfolio Diversification with β-Variational Autoencoder Embeddings

```
Require: $\mathcal{P}$: Initial portfolio of tickers, $\mathcal{E}_{\mathcal{P}}$: Embeddings for portfolio tickers, $\mathcal{R}_{\text{all}}$: Return vectors
    for all tickers, $\mathcal{E}_{\text{all}}$: Embeddings for all stocks, $K$: Maximum number of iterations, $N$: Number
    of top dissimilar candidates for replacement, dist: Distance metric (e.g., Euclidean or cosine),
    opt_alg: Portfolio optimization algorithm (e.g., max diversification).
Ensure: Optimized portfolio $\mathcal{P}^*$, diversification ratio history, and replacement log.
Initialize $\mathcal{P}^* \leftarrow \mathcal{P}$ and compute initial diversification ratio:
        $\text{DR}(\mathcal{P}^*) = \frac{\text{Weighted Volatility of Assets}}{\text{Portfolio Volatility}}.$
Exclude $\mathcal{P}$ tickers from $\mathcal{E}_{\text{all}}$ and set diversification ratio history: $\mathcal{H} = [\text{DR}(\mathcal{P}^*)]$.
Initialize replacement log: $\mathcal{L} = \emptyset$.
for $k = 1, \ldots, K$ do
    Step 1: Find the Most Similar Pair of Stocks in $\mathcal{P}^*$.
    Compute pairwise distances between portfolio embeddings $\mathcal{E}_{\mathcal{P}}$:
        $\text{Dist}(\mathbf{z}_i, \mathbf{z}_j) = \|\mathbf{z}_i - \mathbf{z}_j\|_{\text{dist}}, \quad i \neq j.$
    Identify the most similar pair:
        $(\mathbf{y}_i, \mathbf{y}_j) = \arg\min_{i,j} \text{Dist}(\mathbf{z}_i, \mathbf{z}_j), \quad i \neq j.$
    Randomly select one of the two stocks to replace, $\mathbf{y}_r \in \{\mathbf{y}_i, \mathbf{y}_j\}$.
    Step 2: Find Replacement Candidates.
    Compute distances from $\mathbf{z}_r$ to embeddings in $\mathcal{E}_{\text{all}}$:
        $\text{Dist}(\mathbf{z}_r, \mathbf{z}_k) = \|\mathbf{z}_r - \mathbf{z}_k\|_{\text{dist}}.$
    Identify the $N$ most dissimilar stocks:
        $\mathcal{C} = \{\mathbf{z}_{k_1}, \ldots, \mathbf{z}_{k_N}\}, \quad \text{where } \text{Dist}(\mathbf{z}_r, \mathbf{z}_{k_1}) > \cdots > \text{Dist}(\mathbf{z}_r, \mathbf{z}_{k_N}).$
    Randomly select a replacement ticker $\mathbf{y}_{\text{new}} \in \mathcal{C}$.
    Step 3: Evaluate the Swap.
    Temporarily create a portfolio $\mathcal{P}'$ with $\mathbf{y}_r$ replaced by $\mathbf{y}_{\text{new}}$:
        $\mathcal{P}' = (\mathcal{P}^* \setminus \{\mathbf{y}_r\}) \cup \{\mathbf{y}_{\text{new}}\}.$
    Optimize $\mathcal{P}'$ using opt_alg and calculate its diversification ratio:
        $\text{DR}(\mathcal{P}') = \frac{\text{Weighted Volatility of Assets in } \mathcal{P}'}{\text{Portfolio Volatility of } \mathcal{P}'}.$
    Step 4: Accept or Reject the Swap.
    if $\text{DR}(\mathcal{P}') > \text{DR}(\mathcal{P}^*)$ then
        Accept the swap: $\mathcal{P}^* \leftarrow \mathcal{P}'$.
        Update $\mathcal{H}$: Append $\text{DR}(\mathcal{P}^*)$.
        Log replacement: $\mathcal{L}[\mathbf{y}_r] \leftarrow \mathbf{y}_{\text{new}}$.
    else
        Reject the swap.
    end if
    Step 5: Termination Check.
    if $\text{DR}(\mathcal{P}^*)$ has not improved for consecutive iterations or target metrics reached then
        Terminate the algorithm.
    end if
end forreturn $\mathcal{P}^*$, $\mathcal{H}$, $\mathcal{L}$.
```