Asset Embeddings

Xavier Gabaix Ralph S.J. Koijen Robert J. Richmond Motohiro Yogo*

August 22, 2024

Abstract

Firm characteristics are ubiquitously used in economics. These characteristics are often based on readily-available information such as accounting data, but those only reflect part of investors' information set. We show that useful information about firm characteristics is embedded in investors' holdings data and, via market clearing, in prices, returns, and trading data. Leveraging recent advancements in artificial intelligence (AI) and machine learning (ML), particularly in representing unstructured data (e.g., text or speech) as continuous vectors in high-dimensional spaces, we develop a methodology to learn asset embeddings directly from holdings data. Indeed, just as documents arrange words that can be used to uncover word structures via word embeddings, investors organize assets in portfolios that can be used to uncover firm characteristics that investors deem important via asset embeddings. This theme provides a natural bridge to connect recent advances in the fields of AI and ML to finance and economics. We study various modeling architectures, including recommender systems, shallow neural network models, and transformer-based BERT models. We demonstrate the value added of asset embeddings by introducing three new benchmarks that can be evaluated using a single quarter of data. The models also generate investor embeddings, which measure investor similarity, and can be used for performance measurement and identifying crowded trades. We show how we can use large language models (LLMs), combined with firm- or investor-level text data, to interpret why firms or investors cluster together. We discuss extensions of our framework to generative portfolios, risk management, and the design of stress tests.

*xgabaix@fas.harvard.edu, ralph.koijen@chicagobooth.edu, rrichmon@stern.nyu.edu, myogo@princeton.edu. For comments and suggestions, we thank Tania Babina, Bernard Bergeron, Leland Bybee, John Campbell, Serhiy Kozak, Bryan Kelly, Tengyuan Liang, Fernando Perez-Cruz, Tarun Ramadorai, Hyun Shin, and seminar participants at seminars and conferences. We thank Manav Chaudhary, San Singh, and Huanyu Zhang for excellent research assistance. For financial support, Gabaix thanks the Ferrante Fund, and Koijen thanks the Center for Research in Security Prices at the University of Chicago and the Fama Research Fund at the University of Chicago Booth School of Business.

1 Introduction

The success of embedding models to analyze text, image, and audio, has revolutionized the understanding of complex unstructured data by representing data (e.g., words or speech) as continuous vectors in a potentially high-dimensional space. Despite the success of embedding techniques in these fields, their application to the realm of finance and economics remains largely unexplored. Our main contribution is to introduce "asset embeddings" and show, both empirically and theoretically, that portfolio holdings (and transformations thereof) are particularly well suited to estimate asset embeddings. We illustrate how those asset embeddings allow economists to go beyond the common practice of using observable characteristics based on, for instance, accounting data.

A text embedding is a numerical representation of text that captures the meaning of words or phrases in a vector space, and is typically extracted from text data via natural language processing (NLP). Similarly, we define an asset embedding as a numerical representation of an asset's or firm's characteristics in a vector space. Asset embeddings can be extracted from holdings data and trading data, as we do in this paper, or other non-traditional sources of data such as text. This contrasts with the more conventional measures of firm characteristics based on readily-available data, such as accounting data.

To estimate asset embeddings, we rely on embedding data sets. Embedding data sets have two dimensions, for instance, daily returns across firms within a quarter or text data across 10-K filings of firms. Our framework is flexible and can accommodate data from a variety of sources, such as prices, returns, volume or even text data. Given this flexibility, our first contribution is to develop a simple equilibrium asset pricing model in Section 2 to argue that holdings data are particularly well suited to estimate asset embeddings. Intuitively, investors choose portfolios based on characteristics that capture a firm's risk and expected returns, and non-pecuniary benefits and costs such as a firm's sustainability. Put simply, documents structure words in ways that allow us to estimate word embeddings and, analogously, investors (institutional and retail investors alike) organize securities in ways that allow us to estimate asset embeddings. Our simple microfoundation makes clear that, in a large class of models, portfolio holdings embed all information that is relevant to investors. The key question is then which method to use to "invert" the asset demand system to uncover the information that is relevant to investors.

In Section 3, we explore three methods inspired by the modern NLP literature: recommender systems, which are closely related to principal components analysis (PCA), simple neural network models (Word2Vec), and the recent transformer architectures that are used in large language models (LLMs) such as BERT and GPT.

For the Word2Vec and transformer models, an important question is how this modeling architecture can be used with *numerical* values instead of discrete *words*. That is, how to transform numerical information on holdings into Word2Vec- or BERT-readable language, which uses words.

Our second contribution is to propose such a bridge, and we estimate a model following the BERT architecture that we label "AssetBERT." In this case, we estimate the model based on ranked portfolio holdings; indeed, we replace the ranking of words in a sentence by the ranking of stocks in a portfolio. In case of BERT, we use a masked language task: we predict the identity of e.g. the 10th largest holdings of a given investor, given data on the other holdings (see Figure 1 below). This is analogous to the completion tasks on which BERT was trained, where BERT has to guess the missing word, in a sentence like "The Fed decided to _____ rates to fight inflation". This broad theme, which can be applied to other modeling approaches in the audio, vision, and text literature, provides a natural bridge to connect recent advances in the fields of AI and ML to finance and economics.

Our third contribution is introduce three benchmarks to evaluate and compare models of asset embeddings. Such benchmarks play a central role in the modern AI literature (e.g., ImageNet in the context of computer vision models) to level the playing field and to objectively measure progress. A key requirement for each of our benchmarks is that we can discriminate between models using just a single quarter of data. In this way, when new models are introduced, we can easily compare them and measure progress using new out-of-sample data. The benchmarks are related to predicting relative valuations of firms, predicting missing firm characteristics, and measuring substitution patterns in investors' portfolios.

We implement the different models and compare them to observed characteristics using data on mutual funds, exchange-traded funds (ETFs), closed-end funds, variable annuity (VA) funds, and hedge funds from 2005.Q1 to 2022.Q4, which is our fourth contribution. Three main insights emerge from this analysis. First, embedding models perform well compared to observed characteristics, and this is particularly true for high-dimensional embedding models. Second, the recommender system models perform well on two of the three benchmarks (i.e., explaining relative valuations and predicting missing characteristics), but fall short in explaining substitution patterns. For this third benchmark, the Word2Vec model and in particular AssetBERT perform particularly well. This highlights the richness of the holdings data and the benefit of exploring different modeling architectures. Third, in addition to observed characteristics, we also explore text-based embeddings from Cohere and OpenAI, which are leading AI companies. Their embedding models not only focus on the name of a firm (e.g., Apple as a fruit), but supposedly also capture the nature of firms (e.g., Apple as a technology firm). We find that text-based embeddings do not perform well in our benchmarks, and discuss potential reasons for why they fail. Exploring ways in which text-based embeddings can be improved, for instance via fine-tuning, is an interesting direction for future research.

Even though our focus in this paper is on asset embeddings, we show how the same models can be used to estimate investor embeddings, which are vector representations of investors and their trading styles. As observed investor characteristics are primarily limited to institutional type (e.g., hedge

funds, mutual funds or insurance companies), size, and activeness, investor embeddings provide a richer representation of investors. We discuss how investor embeddings can be used for performance measurement and to identify crowded trades. We estimate a version of the transformer model for this purpose, which we label the "InvestorBERT," and illustrate its performance in identifying similar investors.

In Section 7, we discuss several extensions. For instance, we discuss how embedding models can be used to generate replicating portfolios without return data. For instance, at the onset of the COVID pandemic, we can identify several stocks that are severely exposed to the pandemic. We then use the AssetBERT model to generate another 10 stocks based on the 10 initial, salient stocks, similar to how transformer models are used to generate text. Second, building on this logic, we can generate changes in asset and investor embeddings that in turn lead to changes in market prices once combined with an asset demand system (Koijen and Yogo, 2019). This provides a new way to generate stress scenarios that can be valuable in risk management applications, while accounting for realistic substitution patterns across assets and investors.

As our final contribution, we discuss in Section 8 how we can using text data, combined with modern LLMs, to interpret asset and investor embeddings. This multi-model approach allows us to (i) use holdings data to learn asset and investor embeddings and (ii) to provide narratives why certain assets and investors cluster in financial markets.

Related literature Our paper relates to various strands in the literature. First, a large literature studies the information contained in observable firm-level characteristics, for instance, to predict returns, model the stochastic discount factor or to predict returns, see Kelly et al. (2019), Kozak et al. (2020), DeMiguel et al. (2020), Feng et al. (2020), Freyberger et al. (2020), Lettau and Pelger (2020), Cong et al. (2022); see Kelly and Xiu (2023) for a recent review. A common concern is that the characteristics used are a, potentially small, subset of the information available to investors (Hansen and Richard, 1987). By inverting the asset demand system and extracting asset embeddings from portfolio holdings, we may be able to get closer to the information set used by investors.

Second, a growing literature explores embeddings in the context of text, image, and audio data. Important contributions that are related to our modeling approach are Ducharme et al. (2003), Pennington et al. (2014), Mikolov et al. (2013a), and, for transformer models, Vaswani et al. (2017) and Devlin et al. (2019). in this context of financial market data, Dolphin et al. (2022) estimate a neural network model for stocks using daily returns to capture the covariance between stock returns. Concretely, they define a target stock (say, Apple) and context stocks as those stocks that have a return that is close (in absolute value) to the target stock. With the target and context in hand, they estimate a neural network model analogous to Mikolov et al. (2013a) and explore whether the asset embeddings are useful for hedging purposes.

Third, we are part of a growing literature using machine learning techniques in economics.

Early work in asset pricing includes Hassan et al. (2019); Gu et al. (2020); Nagel (2021); Bybee et al. (2023); Chen et al. (2023); Hommel et al. (2023). Related, several recent papers point out the challenges in empirical asset pricing research when characteristics are missing (Bryzgalova et al., 2022; Chen and JackMcCoy, 2024; Freyberger et al., 2022; Beckmeyer and Wiedemann, 2023). Asset embeddings that are learned from holdings data are ideally positioned to fill this gap.

Fourth, our paper relates to a recent literature on demand system asset pricing, where the main objective is to develop theoretical and empirical frameworks to jointly understand prices, portfolio holdings, and macroeconomic and firm-level variables. Koijen and Yogo (2019) develop a characteristics-based equity demand system that links portfolio holdings to prices, observable characteristics, and residual demand (labeled latent demand). By solving for equilibrium prices, and substituting those in the demand equations, we show that we can "invert" the demand system to learn the characteristics that determine investors' demand.

Lastly, our paper is also related to a recent literature that estimates factor models based on portfolio holdings for various applications, see Madhavan et al. (2021), Gabaix and Koijen (2022), Betermier et al. (2022), and Balasubramaniam et al. (2023). Madhavan et al. (2021) estimate principal components based on ETF and mutual fund holdings to measure trends in crowdedness over time. Gabaix and Koijen (2022) focus on 13F data and sector-level holdings from the flow of funds to isolate idiosyncratic demand shocks, after removing common factors, to estimate demand elasticities using granular instrumental variables (Gabaix and Koijen (2024)). Betermier et al. (2022) develop a factor model of returns that features two new factors beyond the market return. The first factor is sorted by age and the second by wealth. These additional factors explain both holdings and returns. Balasubramaniam et al. (2023) focus on a matrix of stock coholdings. Their main focus is then on uncovering how households tilt their portfolios along the dimension of observable characteristics and clusters of characteristics using factor models. They compare the explained variation of this model to the fraction explained by a latent factor model. Relative to these papers, our main focus is on learning the representation of assets via asset embeddings, and using state-of-the-art AI and ML methods to do so, and to demonstrate their value added in a series of applications. As a by-product, we also learn a representation of investors via investor embeddings.

Outline Section 2 provides a micro foundation of embeddings data sets with a particular focus on holdings data. Section 3 discusses various methods to estimate asset and investor embeddings building on the insights from the natural language processing literature. Section 5 describes the data. In Section 4, we introduce several benchmarks to compare various embedding models. Section 6 presents the main empirical results. Section 7 discusses extensions, including a programmatic discussion of how our asset and investor embeddings will be useful to address a host of issues in finance. Section 8 provides a method to interpret asset and investor embeddings. Section 9 concludes.

Notations We use the notation $i \in \{1, \ldots, I\}$ for investors, $a \in \{1, \ldots, A\}$ for assets, and q for quarters. As most calculations happen within a quarter, we often omit subscripts q for simplicity. We normalize a firm's shares outstanding to one. A stock's market capitalization (or price) is denoted by P_a, the number shares held by investor i by Q_{ia}, and dollar holdings by $H_{ia} = Q_{ia}P_a$. Lowercase letters denote logarithms, e.g. $h_{ia} = \ln H_{ia}$. $\mathcal{A}_i$ is the set of stocks held by investor i and $|\mathcal{A}_i|$ is the number of stocks held by investor i. For a matrix $M \in \mathbb{R}^{I \times A}$ with elements M_{ia}, $M_i \in \mathbb{R}^A$ denotes the transpose of i-th row; $M_a \in \mathbb{R}^I$ denotes the a-th column; $\|M\|_F = \left(\sum_{i,a} M_{ia}^2\right)^{\frac{1}{2}}$ denotes the Frobenius norm of a matrix M. For two matrices B and C of the same size, $B \odot C$ is element-wise product matrix, $(B \odot C)_{jk} = B_{jk}C_{jk}$. We use the notation β^e for the $e$stimated value of a parameter β.

2 Portfolio holdings as embedding data

A key feature of embedding data is that there are two dimensions in a given period, say, a quarter. For instance, we can use daily returns across stocks, daily volume data or word distributions of firms' 10-K filings. We argue in this section that holdings data, and transformations thereof, are particularly well suited as embedding data. We start with a simple equilibrium model of asset demand to micro-found the use of holdings data as embedding data. Appendix A provides additional derivations.

We model log dollar demand of investor i for asset a as

$$h_{ia} = c_i^h + (1 - \zeta_i)p_a + \nu_{ia}, \tag{1}$$

where ζ_i is the investor's demand elasticity, ν_{ia} the demand shifter, and c_n^h captures the investor's size.[1] We assume that the demand shifter has a factor structure,

$$\nu_{ia} = x_a' \lambda_i^\nu + u_{ia}, \tag{2}$$

where x_a is the *asset embedding* that captures how much stock a is affected by each embedding dimension. The *investor embedding*, λ_i^ν, captures how an investor tilts its portfolio to the various embedding dimensions. In the simple model of this section, they enter linearly, but they can enter nonlinearly in more general models (e.g. with bounds that prevent an investor from over- or under-weighting a stock by too much). The residuals, u_{ia}, are idiosyncratic bets of investor i on stock a. There are different micro foundations that give rise to this model of demand (Koijen and Yogo, 2019; Koijen et al., 2022), for instance by assuming that investors have mean-variance preferences and

[1] In the language of Gabaix and Koijen (2022) this ζ_i is the micro elasticity $\zeta_i^\perp$ (price elasticity of demand when choosing between different stocks), rather than the macro elasticity (price elasticity of demand when choosing between the aggregate stock market vs the aggregate bond market), but for notational simplicity we omit the $\perp$.

that returns follow a factor model. The asset embeddings, x_a, then capture variation in expected returns (or alphas) and heterogeneity in factor loadings across assets.

In Appendix A, we show that when we impose market clearing, solve for asset prices, and substitute the equilibrium prices in (1), we obtain the equilibrium holdings

$$h_{ia} = \phi_i^h + \phi_a^h + \lambda_i' x_a + \epsilon_{ia},$$

where λ_i is an linear transformation of λ_i^ν and ζ_i. Only when all investors are identical in terms of $\lambda_i^\nu = \lambda^\nu$ and $\zeta_i = \zeta$, will $\lambda_i = 0$, in which case we cannot use equilibrium holdings to recover asset embeddings. However, this is an empirically irrelevant special case given the heterogeneity in investment strategies and trading behavior observed empirically.

We also show in Appendix A that when asset embeddings do not change too much over time, $\Delta x_{aq} \simeq 0$, then we have for rebalancing

$$\Delta h_{iaq} = \Delta\phi_{iq}^h + \Delta\phi_{aq}^h + \Delta\lambda_{iq}' x_{a,q-1} + \Delta\epsilon_{iaq},$$

which relates directly to returns.

Our broad economic insight is that, in a large class of models (including many models with private information), holdings data embed all information that is relevant to investors. Indeed, the often-cited notion that econometricians know less than investors holds true primarily when holdings data is not accessible - such as when the econometrician can only use data on fundamentals and prices.

3 Applying modern AI methods in economics and finance

3.1 The main idea: From words and sentences to assets and investors

Building on the insight that holdings data encapsulate all relevant investor information, the next step is to determine the best methods to extract this information from portfolio holdings or rebalancing. Recent advances in artificial intelligence (AI) and machine learning (ML), especially in areas like audio, image, and natural language processing (NLP), offer valuable tools for this purpose. NLP models, in particular, provide a compelling analogy as they represent data (e.g., words or speech) as continuous vectors in high-dimensional spaces, known as embeddings. These embeddings capture similarities between words, sentiments or documents and can be used to do "math with words." A classic example is to use the vector representations of "king", "woman", and "man" to compute king - man + woman, which yields a vector close to the word "queen".

In economics, we are often interested in modeling the similarity between firms and assets, with the common practice being to use observed characteristics. The AI literature offers a different

approach: learning embeddings directly from data. Given that holdings data encapsulate all relevant information for investors, it provides a natural foundation for estimating asset embeddings using AI techniques.

Our central insight is a simple, yet broadly applicable one. Just as documents organize words in NLP, images organize pixels in computer vision, songs organize notes in audio, investors in financial markets organize assets in finance and economics. We can therefore adapt the existing modeling frameworks such that we can use holdings data to estimate asset embeddings.

Using this connection, we now discuss at a high, conceptual level how to estimate asset embeddings with holdings data by using NLP models that have been developed during the last three decades . While this overview is obviously not exhaustive, our main goal is to explain the analogy between problems and modeling approaches in the NLP literature and their application to estimating asset embeddings. For textbook treatments of these methods, we refer to Jurafsky and Martin (2023) and Prince (2023).

The traditional approach is Latent Semantic Analysis (LSA) (Dumais et al., 1988), which applies principal components analysis on the document-word matrix. We start by exploring such factor models in Section 3.2 based on the matrix of portfolio holdings across investors for different assets.

The recent AI and ML literature went well beyond those methods. A first important innovation is to use local, non-linear methods, such as Word2Vec (Mikolov et al., 2013b,a) and GloVe (Pennington et al., 2014). In case of Word2Vec, embeddings are used in a shallow neural network to predict a target word based on the surrounding words in the target word's context window.[2] To map this model to holdings data, we first rank stocks based on the positions taken by an investor.[3] We then model the probability that an investor holds a position in a given stock, say, Google based on similarly-sized positions in, say, Apple, Microsoft, and Meta using the same neural network. Indeed, just as sentences order words, we exploit the fact that investors order stocks in their portfolios. We discuss this approach in more detail in Section 3.3.[4]

There are two important limitations of the Word2Vec approach. First, the ordering of words within the context window does not matter. Second, the embedding is context-invariant and there is a one-to-one mapping from words to the embedding vectors. These limitations are relevant also in economics: E.g., Tesla can be viewed as an electric-vehicle (EV) company, a battery company or, more recently, an AI company, depending on the other firms to which Tesla is being compared.

[2]Alternatively, we can predict the words in the context of the target word based on the target word itself. The two methods are labeled the continuous bag of words and the continuous skip-gram approach.

[3]Alternatively, we can rank stocks by the deviation of an investors portfolio from the market portfolio (i.e., active holdings) or we can use investors' portfolio rebalancing instead of using the level of holdings..

[4]GloVe is a related method, which takes the co-occurrence matrix of words across documents as its input, where co-occurring words are measured in a context window around a target word. Next, embeddings are estimated via a weighted principal components analysis, while putting more weight is on words that frequently co-occur. It is straightforward to apply this method in the context of investors' portfolios by forming a matrix of stocks that frequently co-occur in a certain window.

Those limitations are addressed in the third generation of NLP models: transformer models. Recent large language models (LLM) such as BERT and GPT follow this architecture. The key idea is that we start from a context-invariant embedding, and then adjust the embedding by averaging the initial embedding with the embeddings of surrounding words in a sentence or, in our case, surrounding stocks in an investor's portfolio. Hence, instead of a small window of context stocks, we use the entire portfolio,with multiple layers in a neural network used to average the initial embeddings to provide context. We provide further details in Section 3.4.

3.2 Recommender systems

The simplest, yet powerful, class of models that we consider are recommender systems, which are closely connected to LSA models (Dumais et al., 1988). Motivated by the asset demand system in 2, with parameters $\theta = (\delta_i, \delta_a, x_a, \lambda_i)$, we solve

$$\min_{\theta} \sum_{i,a} (h_{ia} - \delta_i - \delta_a - x_a' \lambda_i)^2 + \frac{\xi}{AK} \sum_a x_a' x_a + \frac{\xi}{IK} \sum_i \lambda_i' \lambda_i, \tag{3}$$

where we sum over non-missing values of the log holdings, h_{ia}. The first term is simply the mean-squared error loss for a factor model on log holdings, while the second and third terms are used to regularize the estimation of the asset embeddings, x_a, and investor embeddings, λ_i. We determine the regularization parameter ξ using 10-fold cross-validation. We provide further details in Appendix C.

Recommender systems were very successful in the famous Netflix challenge, which asked researcher teams to predict which movies a viewers i would like, given their past ratings h_{ia} (available only for a small subset of movies). Then x_a summarizes the deep characteristics of the movie, and λ_i the utility weight of the user on the movie.

It is useful to consider why embeddings are useful, potentially much more than "observable characteristics." Suppose that some viewers like teenage vampire movies. It would be very cumbersome for researchers to try to hand-code those characteristics x_{ak}, and similar niche characteristics. But a recommender system, estimating x_a without hand-coding, will automatically generate characteristics k that are akin to "teenage" and "vampire", and will give movies of that genre a high x_{ak}; and their aficionados will have a high loading λ_{ik} on those characteristics.

Likewise, suppose that some investors like stocks with "a large number of GPUs for AI computing." This information is not in plain accounting data, and would be hard to encode. However, a recommender system (and more generally, ML techniques) will uncover it, create a characteristic x_{ak} akin to a firm's "GPU infrastructure" for firm a, and the investors paying attention to it will have a high λ_{ik}. Given the measurement challenges associated with certain firm characteristics related to broad economic questions, such as a firm's intangible capital or the ability to work from

home, learned asset embeddings can provide meaningful representations of firms.

We study numerous variants of the recommender system specified in 3 that are designed to explore what can be learned from different dimensions of the holdings data. As investors' portfolios are sparse, we first explore what can be learned from variation in the extensive margin alone by estimating the model in (3) using only binary data: we set $h_{ia} = 1$ if $H_{ia} > 0$ and $h_{ia} = 0$ otherwise.[5] We do not impose any regularization ($\xi = 0$) and simply estimate the model using PCA. We refer to this method as "RS-Binary."

Next, we use the ranking of assets in an investor's portfolio but not the actual portfolio shares. We set h_{ia} to the percentile rank (with $h_{ia} = 1$ for the largest position and $h_{ia} = \frac{1}{|\mathcal{A}_i|}$ for the smallest position) and replace missing values by zeros. As before, we estimate this model using PCA. We refer to this method as "RS-Ranks."

We then consider three variants that use the actual amounts held, but that make different economic assumption on how to handle the missing values. In the first two variants, we remove the investor and stock fixed effects (δ_i and δ_a) and define $\check{h}_{ia} = h_{ia} - \delta_i - \delta_a$. We then replace the missing values by either 0, a method we label "RS-Level-0", or by the smallest value of $\check{h}_{ia}$ for a given investor, which we refer to as "RS-Level-Min." In the former method, we consider the missing values to be uninformative and driven by, for instance, investor mandates. We therefore set them to the average value of 0. In the second method, we think of the missing values as being informative and zero holdings are best interpreted as investors expressing a negative view on a security and binding short-sales constraints. Both of these methods are also estimated using PCA.

Lastly, we implement the recommender system as in (3) and only sum over the non-missing values. In this case, we only use intensive-margin variation in investors' portfolios. We refer to this model simply as "RS-Level".

It is a priori unclear on theoretical grounds which method performs best, and we therefore provide abroad comparison using the benchmarks that we introduce in Section 4.

Extension 1: Historical data We can estimate (3) using a single quarter or using multiple quarters. One natural way to connect multiple quarters is to add another regularization terms of the form $\|X_t - X_{t-1}\|_F$, which allows for limited time variation in asset embeddings. Intuitively, the asset embeddings from the last period serve as a prior to estimating this period's asset embeddings. This logic can naturally be extended to more advanced time-series models for the parameters.

Extension 2: Supervised embeddings As a second extension, we can estimate supervised asset embeddings, inspired by Bryzgalova et al. (2023). With parameters $\theta = (x_a, \lambda_{iq}, \delta_{iq}, \delta_{aq}, \delta_t, \beta_t)$,

[5] In this case, we set $\delta_i = 0$.

we solve:

$$\min_{\theta} \frac{1-\kappa}{N_h \sigma_h^2} \sum_{i,a} (h_{ia} - \delta_i - \delta_a - x_a' \lambda_i)^2 + \frac{\kappa}{N_y \sigma_y^2} \sum_a (y_a - \delta - \beta' x_a)^2 + \frac{\xi}{AK} \sum_a x_a' x_a + \frac{\xi}{IK} \sum_i \lambda_i' \lambda_i,$$

with N_h and N_y the number of non-missing observations of h_{ia} and y_a, σ_h^2 and σ_y^2 their variances, and $\kappa \in [0, 1]$ controls the relative importance of the holdings data and the model data, y_a. By using the model data, we can extract asset embeddings that account for the task for which the embeddings are eventually used. As with ξ, we can determine κ using 10-fold cross-validation.

In our exploration, we found that high-dimensional unsupervised embeddings perform well in the benchmarks when we use regularization in the benchmarks themselves, so we do not explore the supervised embeddings in detail in this paper. That said, depending on the application, supervised asset embeddings may be a powerful extension of the unsupervised methods.

3.3 Simple neural network models: Word2Vec

The next important step in the NLP literature is Word2Vec (Mikolov et al. (2013a,b)). While PCA considers all the words and tries to fit a global model, Word2Vec pays attention to words in a narrow window around a target word.

To estimate the model, we take advantage of the fact that investors order assets just as sentences order words. To apply the Word2Vec model, we therefore compute the rank, ρ_{ia}, of asset a in investor i's portfolio. For a given investor i, we then predict the name ranked ρ, given that we are given the names of the neighboring stocks.[6] Call $\mathcal{N}_\rho = \{b : 0 < |\rho_i - \rho_{ib}| \leq \delta\}$ the known assets in neighborhood of rank ρ, i.e., the stocks whose rank differs by at most δ or rank ρ; and define $\bar{x}_{i\rho} \coloneqq \frac{\sum_{b \in \mathcal{N}_{i\rho}} x_b}{|\mathcal{N}_{i\rho}|}$ to the average value of the embedding around rank ρ for investor i. Intuitively, the missing stock a at rank ρ can be expected to have similar characteristics to those of neighboring stocks, so similar to $\bar{x}_{i\rho}$. Word2Vec uses this idea, and uses $\bar{x}_{i\rho}$ to predict the asset a at rank ρ, with in the following manner:

$$\mathbb{P}(\rho_{ai} = \rho \mid \bar{x}_{i\rho}) = \frac{\exp(x_a' \bar{x}_{i\rho})}{\sum_c \exp(x_c' \bar{x}_{i\rho})} \tag{4}$$

i.e., x_a should have high covariance with $\bar{x}_\rho$. The embeddings are estimated to maximize that likelihood.

The methodology discussed so far is designed to provide asset embeddings. We can follow the

[6]Here we describe the "continuous bag of words" version of Word2Vec. There is also a "continuous skip-gram" version, which predicts the whole neighborhood with just one word.

same logic to construct investor embeddings. Instead of ranking stocks for a given investor, we rank investors for a given stock. We can then predict the probability that investor i is the n-th largest owner of a stock based on the other investors holding the stock. We implement this approach for the transformer model, discussed next,to illustrate its ability to identify similar investors.

3.4 Transformer models

In the models discussed so far, there is a one-to-one mapping from a stock to an embedding vector, x_a. The recent explosion of work in the NLP literature uses transformer models introduced by Vaswani et al. (2017) that are central to recent large language models, including BERT[7] models (Devlin et al., 2019) and generative pre-trained transformer (GPT) models (Radford et al., 2018). Transformer models are deep neural network models that add an attention mechanism that results in contextualized embeddings. Transformer models can feature an encoder and/or a decoder part. In case of BERT, which is the modeling architecture we adopt in the paper, the model features only the encoder layers of the transformer.[8]

A key question is how this modeling architecture can be used with *numerical* values rather than discrete *words*. That is, how to transform numerical information on holdings h_{ia} into BERT-readable structure, which uses words. In this section, we propose a solution to this problem and refer to the resulting BERT model estimated using numerical holdings data as AssetBERT. We first explain the idea behind the attention mechanism and then show how to solve the problem of using numerical data in a BERT model.

3.4.1 Main intuition and a simple example of attention

The basic idea behind attention can be understood with a simple example. There are numerous articles and blog posts explaining this intuition, including textbook treatments in Jurafsky and Martin (2023) and Prince (2023), and we specialize it to our context. Suppose that the set of stocks held by investor i is $\mathcal{N}_i$, and we are interested in the embedding of asset $a \in \mathcal{N}_i$, which here is a stock. We start from a stock's initial embedding x_a (the "query"), and compute the similarity score with all stocks in the investor's portfolio, x_b (the "keys")

$$\sigma_{ab} = x_a' x_b.$$

[7]BERT stands for Bidirectional Encoder Representations from Transformers.

[8]The GPT architecture uses constrained self-attention using only the context to the left so it can be used for text generation. The BERT architecture, as we discuss below in more detail, is bidirectional and uses information from the left and the right of a word. In the context of portfolio holdings, economic intuition suggests that using information from the entire portfolio is most meaningful to obtain contextualized asset embeddings. However, as we will discuss in Section 7.2, the model can be used to generate portfolios and it may be the case that GPT-style architectures are particularly well suited for that.

We then compute the contextualized embedding, x_a^i, by computing the weighted average of the embeddings in the investor's portfolio

$$x_a^i = \sum_{b \in \mathcal{N}_i} \frac{e^{\sigma_{ab}}}{\sum_{c \in \mathcal{N}_i} e^{\sigma_{ac}}} x_b.$$

The embedding used in the weighted average, x_b, is referred to as the "value." Intuitively, if x_a is an embedding vector for an asset capturing information about a firm's industry, reliance on external finance, and supply-chain risk. Then, depending on the other stocks in an investor's portfolio, we obtain a different embedding vector for the same firm, perhaps better reflecting the firm's industry or external financing conditions.

In computing the contextualized embedding, we used the original embedding, x_a, in three places, namely for the query, keys, and values. The transformer model then generalizes this logic and uses $q_a = W^Q x_a$ as query, $k_a = W^K x_a$ as keys, and $v_a = W^V x_a$ as values, where the W are matrices to be estimated. The contextualized embedding is then computed as

$$x_a^i = \sum_{b \in \mathcal{N}_i} \frac{e^{\sigma_{ab}}}{\sum_{c \in \mathcal{N}_i} e^{\sigma_{ac}}} v_b, \qquad \sigma_{ab} = q_a' k_b.$$

As is clear from this structure, the attention mechanism takes into account an investor's entire portfolio. The transformer encoder stacks multiple of such layers and adds a feed-forward layer after each attention layer. Moreover, instead of a single attention layer, there are multiple attention heads that help to speed up the calculations and may improve the model's performance as the multiple attention heads can focus on different aspects of an investor's portfolio. Michel et al. (2019) provides an in-depth discussion on the benefits of multi-headed attention mechanisms.

What is still missing from the model so far is the location of the words or, in our case, the position of the stock in an investor's portfolio. Without such information, the model treats the inputs as a bag of words (or bag of stocks). To incorporate this important information, we rank all stocks in an investor's portfolio based on the holdings (or, alternatively, active weights or rebalancing) to determine a stock's position. We then add position embeddings to the initial asset embeddings x_a and use this instead as inputs into the model. The position embeddings are estimated in modern BERT models along with the initial asset embeddings and the parameters of the attention layers. [9]

3.4.2 AssetBERT: Architecture choices and inference

Before explaining our AssetBERT's architecture and how we estimate it, we summarize the classic BERT model and how it has been estimated. The original BERT model has been estimated based

[9] We also explored versions of the AssetBERT model with sinusoidal position embeddings as in the original Devlin et al. (2019) model. However, the model with trained position embeddings performs better empirically, and we therefore use this model throughout.

on two tasks: masked language modeling and next-sentence prediction. As next sentence prediction is not directly relevant in our setting, we focus on masked language modeling only.[10] In case of masked language modeling, one (or multiple) of the input words (referred to as tokens) is masked in a sentence. The model then generates a prediction for the masked word based on the final embedding for that word (which depends on the other words in the sentence via the attention mechanism). Indeed, the model generates a distribution over words for the masked word based on the embedding after the final transformer block. In practice, 15% of the words are masked.[11] The model is then estimated to maximize the probability of picking the right word on a training sample and evaluated on a test sample to make sure that the model does not overfit.

To estimate AssetBERT, our baseline solution is the following. For each investor i, we order the assets' names $a^i(1), a^i(2), \ldots, a^i(A)$, by decreasing holdings sizes h_{ia}, so that $a^i(k)$ is the name of the k-th largest position of investor i. This is a "sentence," which would look like, for a hypothetical investor 1:[12]

$$\text{Apple, IBM, Tesla,...., Walmart} \tag{5}$$

We start with a sentence or investor 1, and then have a second sentence for investor 2, which might be IBM, Cisco, Apple,..., Ford. The tokens are just the stock names, without further tokenization (except for e.g. an end-of-sentence separator token, like [CLS] and [SEP]).[13] So, there are A tokens, the number of assets, plus a few extra BERT-specific tokens.

To estimate the model, we mask one or more of the stocks. See for instance the Ark exchange traded fund (ETF) in July 2023 in Figure 1 (the ARK Innovation ETF was then a popular technology fund). In this example, the model sees the holdings of all stocks in the ARK Innovation ETF, except the one at holdings rank 4, which turns out to be Zoom (in practice, we mask more than one stocks in the training procedure). The task for the model is to predict the identity of that asset at rank 4. The parameters are then chosen the maximize the likelihood that the model predicts the correct stock, for this task and similar masked assets at other institutions. Formally, the metric of performance is cross-entropy.

The BERT models that we estimate have in all cases four attention layers, two attention heads per attention layer, and a context window of 64 stocks. If investors hold more than 64 stocks, we split the portfolio in equal chunks before feeding the data into the model.

[10] One idea would be to split portfolios into two parts, for instance, the first 30 holdings and the next 30 holdings. The model can then be used to predict which portfolios belong together. We thank Fernando Perez-Cruz for this suggestion.

[11] Of the masked tokens, 80% receive the token [MASK], 10% are the actual token, and 10% are replaced by a random token, see Appendix A of Devlin et al. (2019). We follow the same masking scheme for AssetBERT.

[12] Because a sentence can have a maximal length, we actually break it in smaller chunks, but this is conceptually not important.

[13] There are five BERT special tokens, [CLS] for the beginning of a sentence, [SEP] for the end of a sentence, [UNK] for unknown tokens (which does not happen in holdings data), [PAD] (to complete the length of the sentence to a standard batch size of, in our case, 64), and [MASK] (for the masked tokens).

Figure 1: **Example of prediction task for masked holdings** The model sees the holdings of all stocks in the ARK Innovation ETF, except the one with holdings rank 4. The task is to guess (i.e., form a probability distribution) the identity of this masked stock, which turns out to be Zoom. This is analogous to the situation in language models, where a typical task is to predict a missing word in a sentence like "Please pass me the _____ and pepper".

Holdings Data - ARKK
As of 07/07/2023

ARK INVEST

ARKK
ARK Innovation ETF

	Company	Ticker	CUSIP	Shares	Market Value ($)	Weight (%)
1	TESLA INC	TSLA	88160R101	3,496,872	$967,024,982.88	12.43%
2	COINBASE GLOBAL INC -CLASS A	COIN	19260Q107	7,945,138	$620,515,277.80	7.98%
3	ROKU INC	ROKU	77543R102	8,865,426	$546,110,241.60	7.02%
4	~~ZOOM VIDEO COMMUNICATIONS-A~~	~~ZM~~	~~98980L101~~	~~8,258,591~~	~~$534,248,251.79~~	~~6.87%~~
5	UIPATH INC - CLASS A	PATH	90364P105	28,152,366	$463,106,420.70	5.95%
6	BLOCK INC	SQ	852234103	7,069,493	$456,759,942.73	5.87%
7	EXACT SCIENCES CORP	EXAS	30063P105	4,031,264	$368,739,718.08	4.74%
8	UNITY SOFTWARE INC	U	91332U101	8,350,868	$338,627,697.40	4.35%
9	SHOPIFY INC - CLASS A	SHOP	82509L107	5,430,238	$335,751,615.54	4.32%
10	DRAFTKINGS INC-CL A	DKNG UW	26142V105	12,035,607	$303,658,364.61	3.90%

3.4.3 InvestorBERT

As discussed in the previous section, the W2V and BERT models can also be used to estimate investor embeddings. Instead of feeding the model a portfolio of stocks for a given investor, we feed it the holdings of different investors for a given stock. We then mask some of the investors and the model's task is to predict the missing investor (as opposed to the missing stock in case of AssetBERT). We refer to this model as InvestorBERT and illustrate the performance of this model in Section 6.

3.4.4 AssetBERT and InvestorBERT: Extensions

We discuss several extensions of the basic BERT models that we explore in this paper.

Time-varying asset embeddings In our current implementation of these models, we focus on a single quarter, or several quarters, to estimate the models. These rolling estimates are simple to implement, but they are likely inefficient. Also, this problem is more unique to our setting compared to language models. While some aspects of a firm or asset are highly persistent over time, others are more likely to change over time (e.g., profitability and riskiness). While holdings reflect this new reality of firms, we may still want to use data from older quarters as well. We discuss two

approaches to model time variation in the context of transformer models.[14]

First, the model is estimated by masking 15% of the stocks in the sample. Instead of applying the masking probability equally across quarters, we can lower the masking probability in earlier quarters in the sample. This implicitly puts less weight on older data.

Second, we can estimate the model in two steps. In this case, we use a longer sample (e.g., 5 or 10 years of data) to estimate the model, and we then fine-tune the model on the current quarter or the last couple of quarters. In this case, we use a long history to estimate the longer-run similarities between firms and the more recent data to update those estimates.

We explored the second approach by pre-training the AssetBERT model up to a period of five years and fine-tuning the model over a period ranging from the current quarter to four quarters. While the additional information helps the model's performance, the improvements are too limited to warrant the additional complexity for the current paper.[15]

An integrated AssetBERT and InvestorBERT model We now discuss how the logic of the AssetBERT and InvestorBERT models can be combined in a single model. Furthermore, in estimating the combined model, we can incorporate information on actual portfolio holdings as opposed to only their ranks.

We outline the architecture of this combined model. First, we start from an AssetBERT model, yielding x_a^i as the contextualized asset embedding for stock a held by investor i. Second, we use an InvestorBERT model to obtain λ_i^a, the contextualized investor embedding for investor i holding stock a. We then train the model using an objective function that averages (potentially with different weights) the cross-entropy of the AssetBERT model, the cross-entropy of the InvestorBERT model, and a least-squares objective to capture the information in actual portfolio holdings,

$$\sum_{i,a}(h_{ia} - \delta_i - \delta_a - x_a^{i'}\lambda_i^a)^2.$$

As we can flexibly choose the weights on each of the three terms of the objective function, the model nests the AssetBERT and InvestorBERT models as special cases.

4 Benchmarking asset embeddings

Benchmarks play a crucial role in the advancement of AI by providing a standardized means to evaluate and compare the performance of different models. In AI, benchmark competitions, such as those based on ImageNet, have been instrumental in identifying the best-performing models and

[14] Alternatively, one can explore which parameters vary significantly over time and only allow for time variation in those parameters. This intuition is similar to the LoRA (Low Rank Approximation) fine-tuning approach of LLMs.

[15] Concretely, the R^2 increases by up to 5% when exploring the ASMP benchmark.

establishing clear metrics for success. These benchmarks drive innovation by leveling the playing field and enabling researchers to gauge the progress made in various domains, such as computer vision.

Inspired by the success of AI benchmarks, we introduce a similar approach in the field of finance. In the remainder of this section, we propose three benchmarks. We designed the benchmarks to be able to discriminate between models in a single quarter. Given this property, one can imagine quarterly competitions where researchers submit their predictive models as black boxes, and we can assess their performance on new, out-of-sample data released each quarter. This methodology closely mirrors current practices, like matching macro-finance moments or pricing Fama-French portfolios, but with a critical difference. The cross-sectional nature of our benchmarks ensures that even a single quarter's data provides a robust test of model accuracy, facilitating a more precise and reliable evaluation of asset embedding models.

4.1 Relative valuations benchmark (RV)

An important task of financial analysts is to determine the relative valuations of firms. A typical approach is to identify comparable firms and to use information about their valuation ratios (such as the price-earnings ratio or book-to-market ratio) to determine a firm's valuation (Hommel et al., 2023). In our first benchmark, we apply this logic using asset embeddings. As a point of reference, we also use observed characteristics.

We regress log market equity, m_{at}, on log book equity, b_{at},

$$m_{at} = \alpha_{0t} + \alpha_{1t} b_{at} + m_{at}^{\perp}, \tag{6}$$

and collect the valuation residual, $m_{at}^{\perp}$. The model is estimated using the cross section of stocks in a given quarter. We then explain the valuation residual using an embedding vector,

$$m_{at}^{\perp} = \gamma_{0t} + \gamma_{1t}' x_{at} + \epsilon_{at}. \tag{7}$$

We estimate $(\gamma_{0t}, \gamma_{1t})$ on 80% of the sample in a quarter. We standardize the embeddings cross-sectionally and regularize the coefficients using a ridge penalty. The penalty parameter is determined using cross-validation (within the 80% sample).

We then evaluate the embeddings on the remaining 20% of the sample. Our measure of performance in this relative valuation (RV) benchmark is the out-of-sample R^2, averaged across quarters,

$$RV = 1 - \frac{1}{T} \sum_{t} \frac{V(m_{at}^{\perp} - \hat{\gamma}_{0t} - \hat{\gamma}_{1t}' x_{at})}{Var(m_{at}^{\perp})}.$$

The recommender system provides a natural reference point to interpret the results. Suppose

that (i) the bilinear model of dimension K is correctly specified and (ii) the λ_i span the K dimensions (precisely $\sum_i \lambda_i \lambda_i'$ has full rank K). Then, as the number of investors $I \to \infty$, we should be able to fully recover the x_a.[16] If, in addition, (iii) the estimation sample is very large ($A \to \infty$), then γ_{1t}^e should be estimated without error, and the $RV \to 1$.

4.2 Missing characteristics benchmark (MC)

A recent literature explores how to handle missing characteristics, which is ubiquitous in finance research (Bryzgalova et al., 2022; Chen and JackMcCoy, 2024; Freyberger et al., 2022; Beckmeyer and Wiedemann, 2023), but is particularly prevalent in cross-sectional asset pricing applications. Broadly speaking, these approaches use cross-sectional or time-series correlations in non-missing characteristics to impute the missing values.

In developing our benchmark, we focus on cross-sectional as we are interested in measuring firms' similarity. [17]

We propose to use asset embeddings to impute the missing values. To test this approach, we mask 20% of the characteristics (and pretend them to be missing) and fill the missing values using a simple cross-sectional regression approach. We estimate the following model on non-missing values,

$$c_{at} = \alpha_{0t} + \alpha_{1t}' x_{at} + u_{at},$$

where c_{at} is the characteristic containing missing values. As before, we regularize the coefficients using a ridge penalty, where the penalty is determined using cross-validation. We then impute the missing values using $\hat{\alpha}_{0t} + \hat{\alpha}_{1t}' x_{at}$. The performance measure is then the out-of-sample R^2, MC, averaged across quarters,

$$MC = 1 - \frac{1}{T} \sum_t \frac{V(c_{at} - \hat{\alpha}_{0t} - \hat{\alpha}_{1t}' x_{at})}{Var(c_{at})}.$$

4.3 Asset similarity in managed portfolios benchmark ($ASMP$)

In the ASMP benchmark, we assess the quality of embeddings to measure similarity of assets using managed portfolios of mutual funds, ETFs, and hedge funds. Specifically, consider an investor i with portfolio θ_i and $\mathcal{N}_i$ the set of stocks held by the investor. The goal is to predict the k-th largest stock based on the positions in all other stocks and the asset embeddings. Concretely, if an investor holds 100 stocks, we for instance mask the second–largest stock ($k = 2$) and predict it using the investor's positions in the other 99 stocks. We focus on small values of k, i.e. the prediction of large

[16] These statements come from the basic results from PCA.

[17] Bryzgalova et al. (2022) show that, due to the persistence in many characteristics, lagged values are often useful in predicting missing values. This information can easily be added in practical applications.

positions.

To evaluate the embeddings, we need a probability model over the missing position, $p_{ib}^k = \mathbb{P}(k_i = b \mid \mathcal{N}_i^{-k})$, where k_i is the k-th largest position of investor i and $\mathcal{N}_i^{-k}$ is the set of stocks held excluding the k-th largest position. The W2V and the AssetBERT models directly provide estimates of p_{ib}^k, and we use those in evaluating those models in this benchmark.

For the recommender system, we rely on an additional assumption. We start from the model

$$h_{ia} = \delta_i + \delta_a + \lambda_i' x_a + u_{ia},$$

where we make the additional assumption that u_{ia} follows a logistic distribution. We estimate δ_i and λ_i using the stocks in $\mathcal{N}_i^{-k}$ only. Intuitively, the missing stock, which is the k-th largest investor's portfolio, corresponds to the largest value of $h_{ib}, b \notin \mathcal{N}_i^{-k}$ if the investor would take a position in all stocks.

This assumption implies that

$$p_{ib}^k = \frac{\exp(\delta_b + \lambda_i' x_b)}{\sum_{c \notin \mathcal{N}_i^{-k}} \exp(\delta_c + \lambda_i' x_c)}.$$

The final benchmark is to average the log likelihood contributions across investors, in excess of the probability of randomly picking a stock,

$$\frac{\frac{1}{I}\sum_i \ln p_{ik_i}^k + \frac{1}{I}\sum_i \ln |\mathcal{N}_i^{-k,c}|}{\frac{1}{I}\sum_i \ln |\mathcal{N}_i^{-k,c}|} \leq 1,$$

where $|\mathcal{N}_i^{-k,c}|$ is the number of stocks in the complement of $\mathcal{N}_i^{-k}$. While not necessary for our application as the number of investors in our sample is large, we can increase the number of observations by repeating this exercise for $k = 2, 3, ..$, that is, leave out one position at the time but vary which positions is omitted.

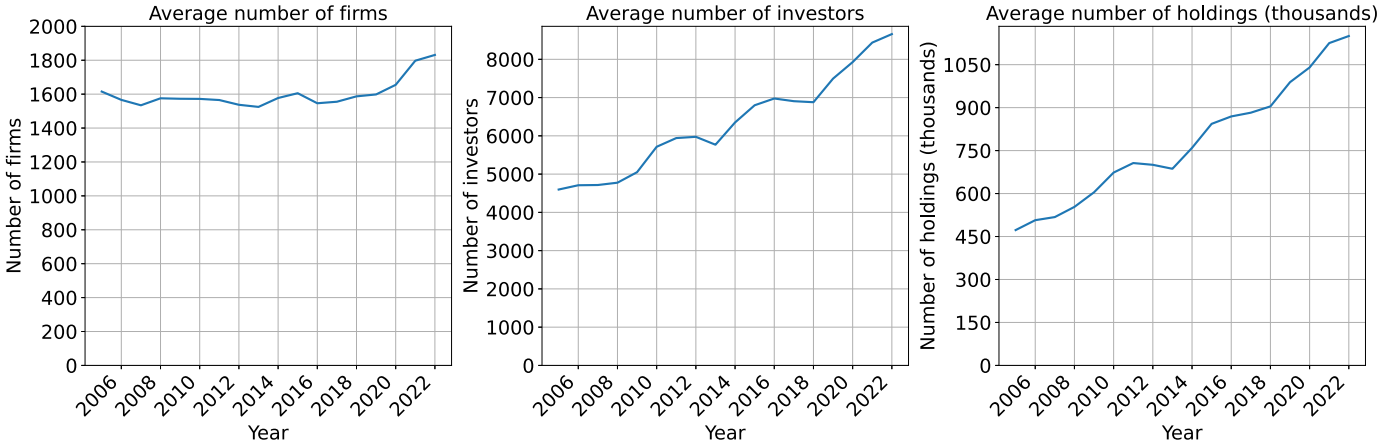
5 Data and competing measures of asset similarity

5.1 Data sources and sample selection

We combine data on equity prices from CRSP, accounting data from Compustat, and holdings data from FactSet. The data construction follows Koijen et al. (2022) and we merge our data with the data from Jensen et al. (2023) for characteristics. We aggregate the data at the company level. To interpret the asset embeddings in Section 8, we also use data on earnings calls sourced from FactSet. Full details on the data construction can be found in D.

We use holdings data on mutual funds, ETFs, closed-end funds, variable annuity funds, and

Figure 2: Average number of firms, investors, and holdings per year. The sample is from 2005.Q1 to 2022.Q4.



hedge funds. The holdings data from hedge funds are sourced from 13F filings. While it is possible to directly estimate asset and investor embeddings using 13F data, the level of aggregation would be much higher (e.g., Vanguard) as opposed to fund-level data (e.g., the Vanguard small-cap value fund). As our goal is to uncover the asset substitution patterns of investors, the more disaggregated fund-level data is more appropriate.[18] By this logic, future research may be able to combine these data with other sources of disaggregated holdings data (see, e.g., Gabaix et al., 2022), for instance from households and other institutions. The ideal data for this study would be from custodians, who observe portfolios from a large number of investors at a high frequency. The sample period is from 2005.Q1 to 2022.Q4.

In selecting our sample, we remove micro caps as measured by Jensen et al. (2023). Also, we keep stocks that are held by at least 20 investors and investors who hold at least 20 stocks. When using historical data on holdings beyond the current quarter, we restrict attention to the list of stocks in the current quarter as the "vocabulary."

In Figure 2, we plot the average number of firms (left panel), investors (middle panel), and holdings (right panel) per year. Due to the growing importance of institutional investors in financial markets, and the associated growth in the number of funds, the number of holdings per firm sharply increases over time.

In the context of LLMs, it is well understood that increasing the number of tokens (the equivalent of portfolio holdings) enables one to estimate larger transformer models for a fixed vocabulary (Kaplan et al., 2020).[19] The trends in the summary statistics are therefore encouraging for asset embedding models, and the model sizes that can be explored.

[18] In an earlier version of the paper, we found that asset embeddings estimated from 13F data work quite well, but the performance increases using the more disaggregated fund-level data, as is to be expected.

[19] Deriving the appropriate scaling laws for holdings-based transformer models is an interesting direction for future research.

5.2 Competing models of asset similarity

Having established established the benchmarks in Section 4, we now introduce the competing models of asset similarity.

Our main focus is on the models of asset embeddings introduced in Section 3. In all cases, we estimate a small models where the dimension of the model matches the dimension of the observed characteristics that we use. We also estimate a larger, 10-dimensional model. In the context of AI, this is still a very small model, and we will explore much larger models for the best performing recommender system and AssetBERT models, up to dimension 128.

As a point of reference, we use observed characteristics. We limit our set of characteristics to the (Fama and French, 2015) characteristics, which include the market beta, size, book-to-market ratio, asset growth (investment), and profitability. We also include momentum as a characteristic, depending on the application. We only include companies for which these characteristics are available.

An unavoidable challenge with selecting characteristics is that there is some discretion in the selection. For instance, in evaluating missing characteristics, if we try to predict missing values of market-to-book, including sales-to-price and sales-to-book would allow the model to perfectly reconstruct market-to-book. We therefore limit ourselves to characteristics that play a central role in the asset pricing literature and that are truly distinct and that have no mechanical correlation. By the same logic, when evaluating the relative valuation benchmark, we only use accounting characteristics and no other price variables (e.g., momentum). So while these lists of characteristics are by no means exhaustive, they provide a reasonable point of reference to benchmark the estimated asset embeddings. Of course, in practical applications, it may be beneficial to expand the set of observable characteristics as in Bryzgalova et al. (2022) or Jensen et al. (2023). We winsorize the characteristics at the 1% and 99% in each quarter.

Lastly, we also use text-based embeddings from Cohere and OpenAI. We use the `embed-english-v3.0` model from Cohere and the `text-embedding-3-large` model from OpenAI. Importantly, the text-based embeddings are not just representing the work (e.g., Apple as the fruit) but supposedly also contain information about the company (Apple Inc. is a technology company). We download the embeddings based on the CRSP company name. Certainly, when prompting, for instance, OpenAI's ChatGPT model about companies, it has detailed knowledge about firms' activities and business models. The Cohere text embeddings are 1024-dimensional and we reduce them to the desired dimension using UMAP (McInnes et al., 2020). The OpenAI API allows us to download the embeddings at any desired dimension.

One important shortcoming of text-based embeddings is that they are static in nature. We therefore use them in the final sample of our quarter, but it is more challenging to use them historically for back-testing (Sarkar and Vafa, 2024). Lastly, we include the text-based embeddings

as a point of reference, as we do with observed characteristics.[20]

6 Main results

We implement the three benchmarks that we outlined in Section 4: relative valuations (Section 6.1), missing characteristics (6.2), and asset similarity in managed portfolios (6.3). We also present results on the performance of the InvestorBERT model in Section 6.4.

6.1 Relative valuations benchmark

In Figure 3, we report the results for the relative valuations (RV) benchmark discussed in Section 6.1. In all evaluations, we include random embeddings to provide a point of reference ("Placebo"). We consider two models that use observed characteristics. The first model only uses the market beta ("Beta") while the second model adds asset growth, profitability, and dividends-to-assets ("Chars") to explain the valuation residual. We then consider five recommender system models: using only information on the extensive margin ("RS-Binary"), using percentile ranks ("RS-Ranks"), using the log levels of holdings, after removing investor and asset fixed effects, and replacing the missing values by zeros ("RS-L-0"), alternatively, replacing the missing values by the smallest value for a given investor ("RS-L-Min"), and only using intensive margin variation ("RS-L"). Lastly, we include the Word2Vec model ("W2V"). Section 3 provides the details on the models. For all embedding models, we consider a "base" variant, which in this case is a 4-dimensional to match the observed characteristics and a "large" variant of dimension 10. We estimate the benchmarks for each quarter in our sample and average the statistics over time.

As expected, the placebo model (of the same size as the "Base" models) cannot explain the valuation residual and $RV \simeq 0$, which is similar to using the market beta. When using a broader set of observed characteristics, we can explain about 15% of the variation, but it remains limited. As discussed before, a broader set of of characteristics could be helpful subject to the caveats discussed in Section 5. The embedding models perform significantly better, even for the same dimension as the characteristics (the "Base" models). Among these models, the recommender system models that replace the missing values by the minimum for a given investor ("RS-L-Min") or the model that only uses intensive margin variation ("RS-L") perform best.

When we increase the depth of the embeddings from $K = 4$ to $K = 10$, we see a significant improvement in all of the embedding models. This suggests that high-dimensional models can learn about as much information relevant for this benchmark from the extensive margin ("RS-Binary") as the intensive margin ("RS-L"). The "RS-L-Min" model performs well across both dimensions.

[20]It is plausible that these embeddings can be improved by fine-tuning them on the task on hand. The comment applies to asset embeddings, and we leave such extensions for future research.

Figure 3: Relative valuation benchmark. Results for the RV benchmark. The models are: random embeddings ("Placebo"), using the market beta ("Beta"), the market beta, asset growth, profitability, and dividends-to-assets ("Chars"), a recommender system using only information on the extensive margin ("RS-Binary"), using percentile ranks ("RS-Ranks"), using the log levels of holdings, after removing investor and asset fixed effects, and replacing the missing values by zeros ("RS-L-0"), alternatively, replacing the missing values by the smallest value for a given investor ("RS-L-Min"), using intensive margin variation ("RS-L"), and the Word2Vec model ("W2V"). The base model corresponds to $K = 4$ and the large model to $K = 10$. The models are estimated and evaluated from 2005.Q1 to 2022.Q4.

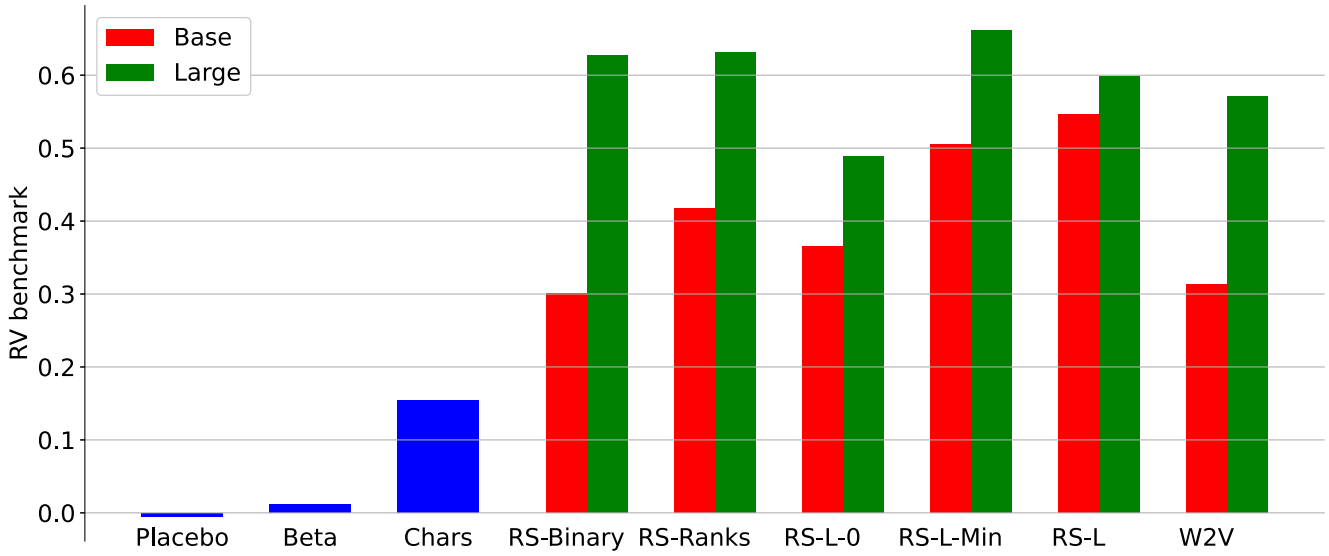


Figure 4: Relative valuation benchmark: Combining embeddings and characteristics. We combine characteristics with the large embedding models in Figure 3.

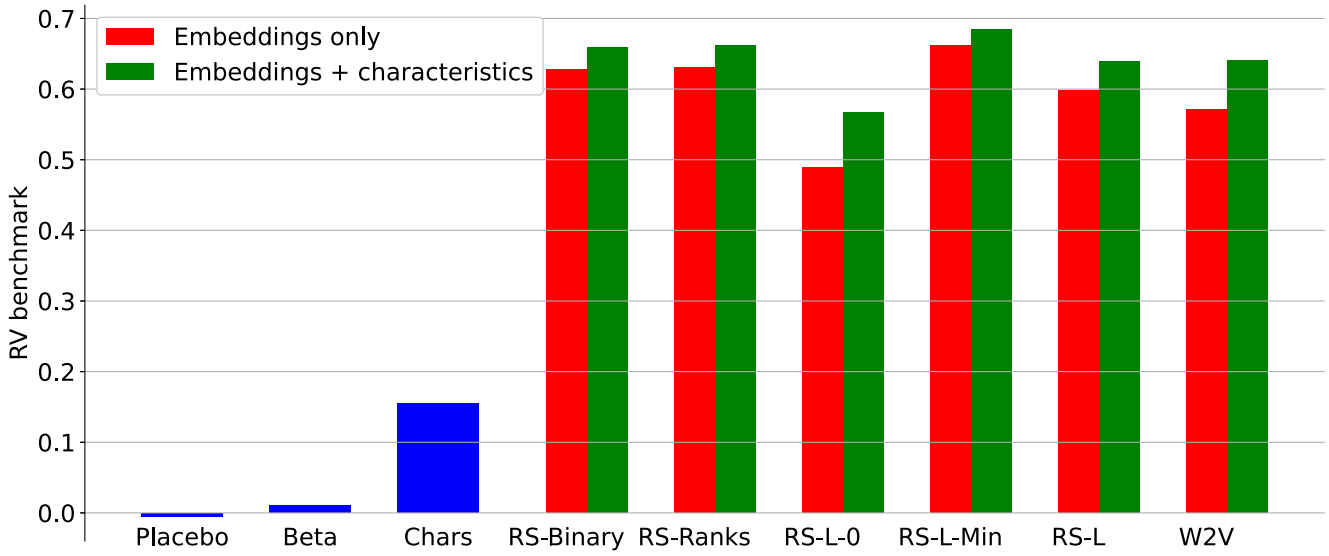
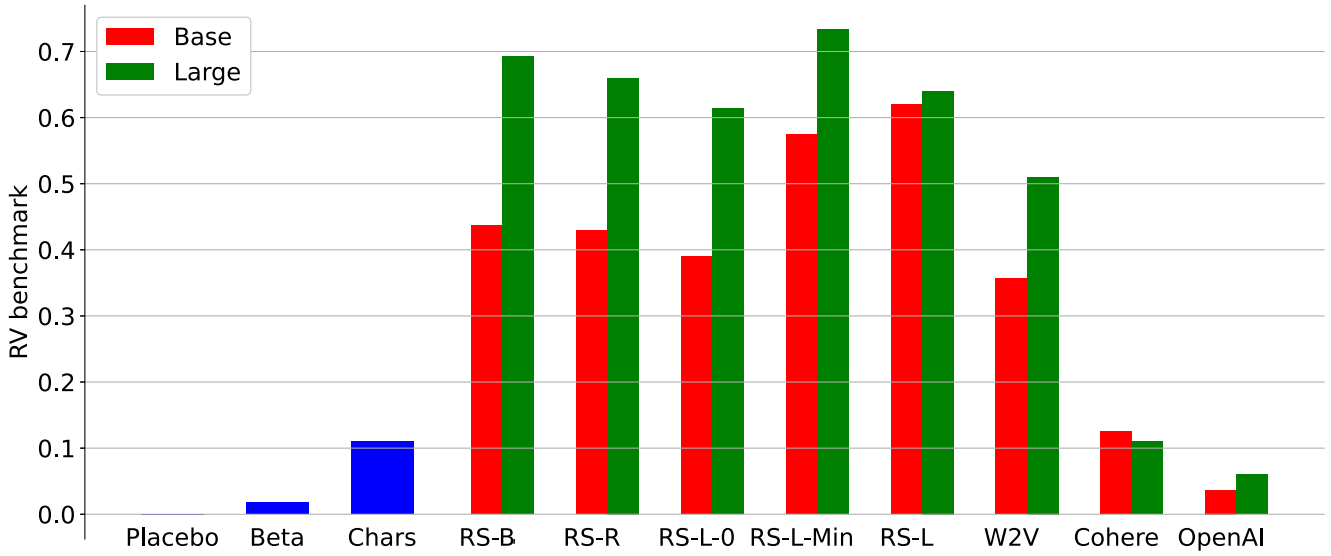


Figure 5: Relative valuation benchmark: Text-based asset embeddings. We use the same embedding models as in Figure 3. In addition, we add text-based embeddings from Cohere and OpenAI. As the text-based embeddings do not have a time dimension, we focus on the final quarter of our sample, 2022.Q4.



In Figure 4, we combine the large embedding models with the four characteristics. In all cases, the improvement is quite minimal, suggesting that most of the relevant information can be extracted via asset embeddings.

In Figure 5, we explore whether text-based embeddings can be used for this benchmark. As we discuss in Section 5, we use embeddings from Cohere and OpenAI and we use the same dimension as the "Base" and "Large" models (which are both very small compared to the depth of the text embeddings). Also, as the text embeddings do not have a time dimension, we focus on the final quarter of our sample (2022.Q4) for this comparison. Figure 5 shows that the text-based embeddings cannot explain the valuation residual well.

To further understand this result, we provide the 10 closest companies for Apple Inc, Citigroup Inc, and Walmart Inc using the embeddings from OpenAI in Table 1. While these embeddings identify firms that are related beyond their names, names still play an important role. For instance, in case of Citigroup Inc, the 10 closest firms are often in the financial industry, yet many firms also start with a "C" or even "Ci." A similar observation can be made for Apple, where the closest firm is Appian, and Walmart, which is closest firm to Walgreens.

As the text-based embeddings are trained on vast amounts of text data and can be used in a broad set of applications, it is plausible that the performance of text-based embeddings can be improved by fine-tuning. We leave such explorations for future research.

To conclude the RV benchmark, we compare the performance of the "RS-Level-min" model to the AssetBERT model. Like the Word2Vec model, it is trained on a different objective and the

Table 1: Text-based embeddings from OpenAI

Input company	Apple Inc	Citigroup Inc	Walmart Inc
Rank			
1	Appian Corp	Citizens Financial Group Inc	Walgreens Boots Alliance Inc
2	Adobe Inc	Goldman Sachs Group Inc	Home Depot Inc
3	Interdigital Inc	American International Group Inc	Murphy Usa Inc
4	Microsoft Corp	Comerica Inc	Amazon Com Inc
5	Gopro Inc	Cigna Corp New	Qurate Retail Inc
6	Netapp Inc	Capital One Financial Corp	Big Lots Inc
7	Intel Corp	Caci International Inc	Burlington Stores Inc
8	Alphabet Inc	Capital City Bank Group	Dollar Tree Inc
9	Autodesk Inc	C N O Financial Group Inc	Nordstrom Inc
10	Appfolio Inc	JP morgan Chase & Co	Kohls Corp

model is more non-linear compared to the bilinear recommender system models. In Figure 6, we plot the RV benchmark for the final sample of our quarter for $K = 4, 8, 16, 32, 64, 128$.

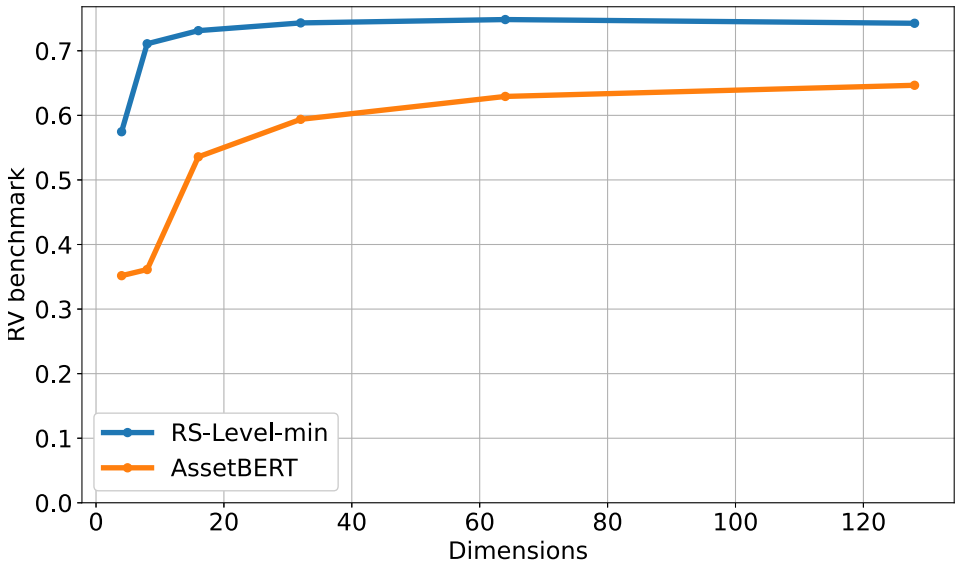
For the AssetBERT model in this benchmark we use the model's input embeddings. Alternatively, we could compute the contextualized embeddings, x_a^i, for each investors and compute the size-weighted embeddings, $\sum_i S_i^a x_a^i$ with S_i^a the fraction of stock a held by investor i.

The figure provides two important insights. First, the AssetBERT model is quite capable of explaining valuation residuals, although the recommender system model performs better for all dimensions. Second, the AssetBERT model improves more significantly as K increases. The recommender system model's performance flattens out at $K = 16$, but the AssetBERT model benefits from increasing K further.

Several refinements of the AssetBERT model can be explored in future research. First, as already discussed, we can extract embeddings from different layers of the model. We currently use the input embeddings, but it may be the case that an aggregation of the contextualized embeddings performs better. Second, and similar to the text-based embeddings, the model can be fine-tuned on each specific task. Indeed, in case of language models, the masked language modeling is a pre-training step and the model is then fine-tuned on a specific task. The same could be done with the AssetBERT model.

Using asset embeddings to improve the practice of firm valuation We think that asset embeddings could be useful for valuation, building on the analysis of Hommel et al. (2023). To value a firm (given its projected cash-flows), the basic textbook method is to use the cost of capital given by the CAPM, using one characteristics, the firm's beta. This is what MBA students learn, but the reality is that many of their teachers now are unconvinced that using the CAPM for the

Figure 6: Relative valuation benchmark: High-dimensional models. We compare the performance of "RS-Level-min" and AssetBERT of various dimensions on the RV benchmark for 2022.Q4. We consider $K = 4, 8, 16, 32, 64, 128$.



cost of capital is right – either in practice (as it gives poor results to explain existing valuations) or in theory (as the CAPM is a very special and antiquated model).[21]

But what to do instead? Instead, people sometimes mention a multi-factor model (as in the Fama-French benchmark we used above), although they also tend to perform poorly for valuation purposes (Hommel et al., 2023). More frequently, they use "comparable" firms. But the choice of comparables is quite challenging. E.g., is Apple a tech company, a consumer good company etc? It has many rough comparables, and the comparability weight to give to each comparable firm is a priori unclear. Assets embeddings offer a systematic, data-driven way to find "comparables" — see the revealed tastes by investors for characteristics; and it also has a theoretical microfoundation (see Section 2).

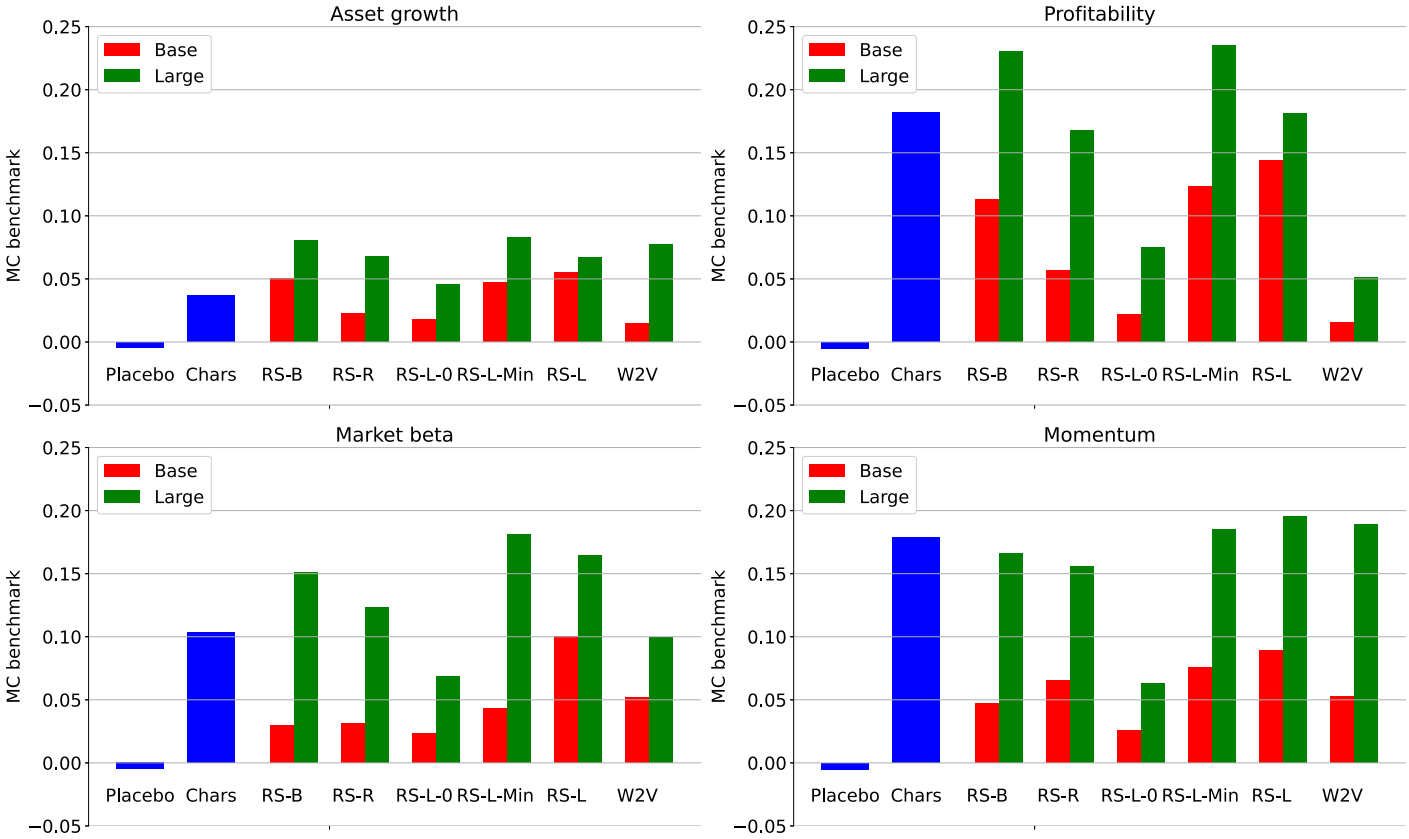
To get the cost of capital of a firm, given its embeddings x_a, there are two methods. First, given projected cashflows and its embedding-predicted valuation, we can infer a cost of capital. Second, we can see how embeddings predict returns (so that the risk premium at time t is $\pi_{at} = \pi_x x_{at}$ for some vector π_x that can be estimated. This yields a cost of capital.

6.2 Missing characteristics benchmark

We now explore the missing characteristics benchmark. The structure of Figure 7 is the same as Figure 3, except that we explore six characteristics to explain the missing characteristic: market beta, asset growth, profitability, log book-to-market ratio, log market capitalization, and momentum. De-

[21] See the Chicago Booth panel (April 18 2023), `https://www.kentclarkcenter.org/surveys/discount-rates/`

Figure 7: Missing characteristics benchmark. The structure of Figure 7 is the same as Figure 3, except that we explore six characteristics to explain the missing characteristic: market beta, asset growth, profitability, log book-to-market ratio, log market capitalization, and momentum. Depending on which characteristic is assumed to contain missing values, the other five characteristics are used to estimate the missing values. The "Base" embedding models are then also of dimension five, while the "Large" models are still of dimension 10. The missing characteristics are asset growth, profitability, the market beta, and momentum.



pending on which characteristic is assumed to contain missing values, the other five characteristics are used to estimate the missing values. The "Base" embedding models are then also of dimension five, while the "Large" models are still of dimension 10.

As before, the placebo embeddings result in an MC benchmark value of about zero, as we would expect. If we use observed characteristics, explaining missing values of profitability and momentum is easier with MC values between 15-25% than market beta and, in particular, asset growth with MC values around 10% or below. The base embedding models perform similarly (in case of asset growth) or underperform compared to the observed characteristics. However, when we increase the depth of the embeddings, the recommender system models tend to outperform the observed characteristics.

Given the benefit of increasing the dimension of the recommender system models, we again explore higher-dimensional models in Figure 8 (analogous to Figure 6). We also use this as an

Figure 8: Missing characteristics benchmark: High-dimensional models. The structure is the same as Figure 6.

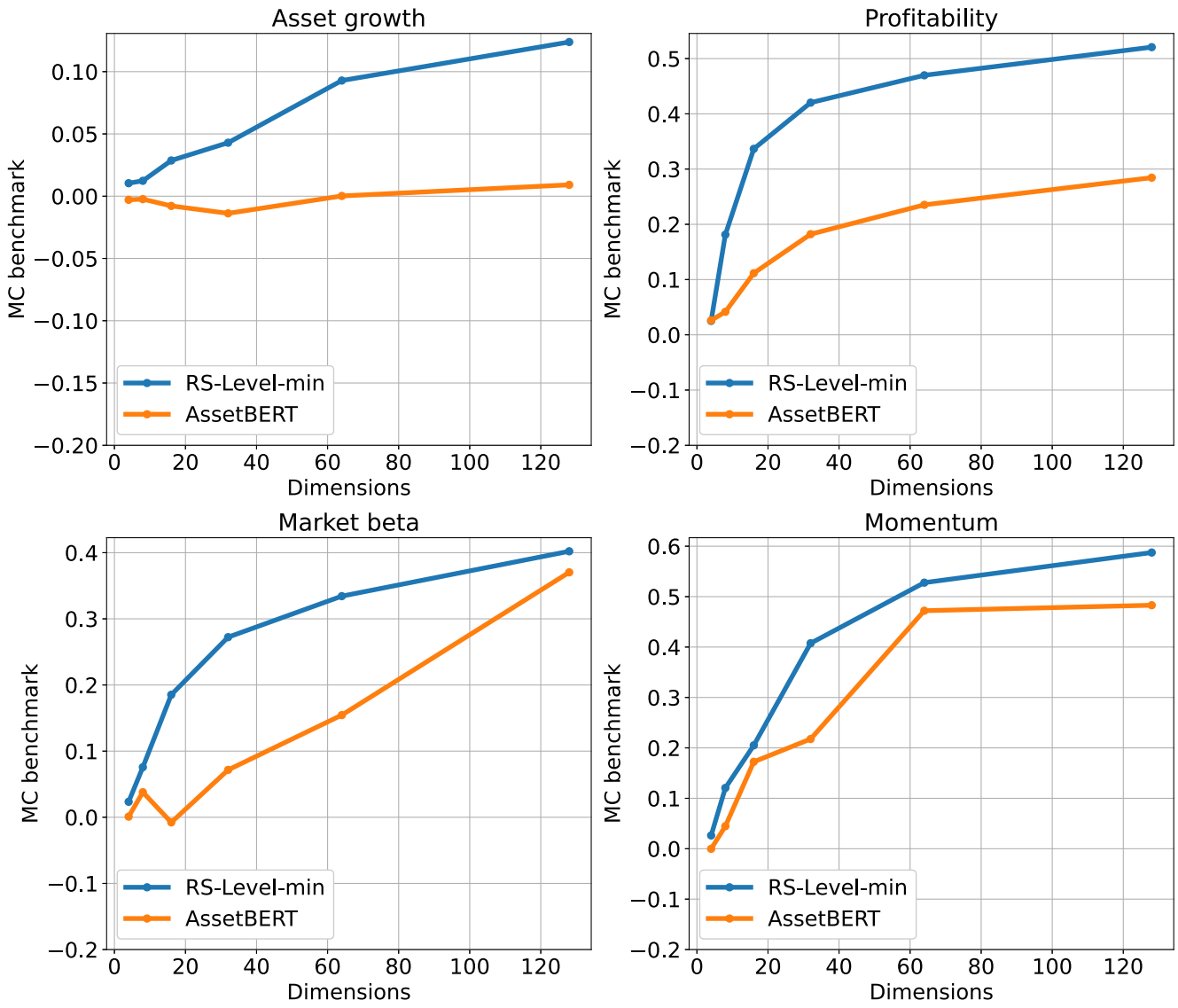
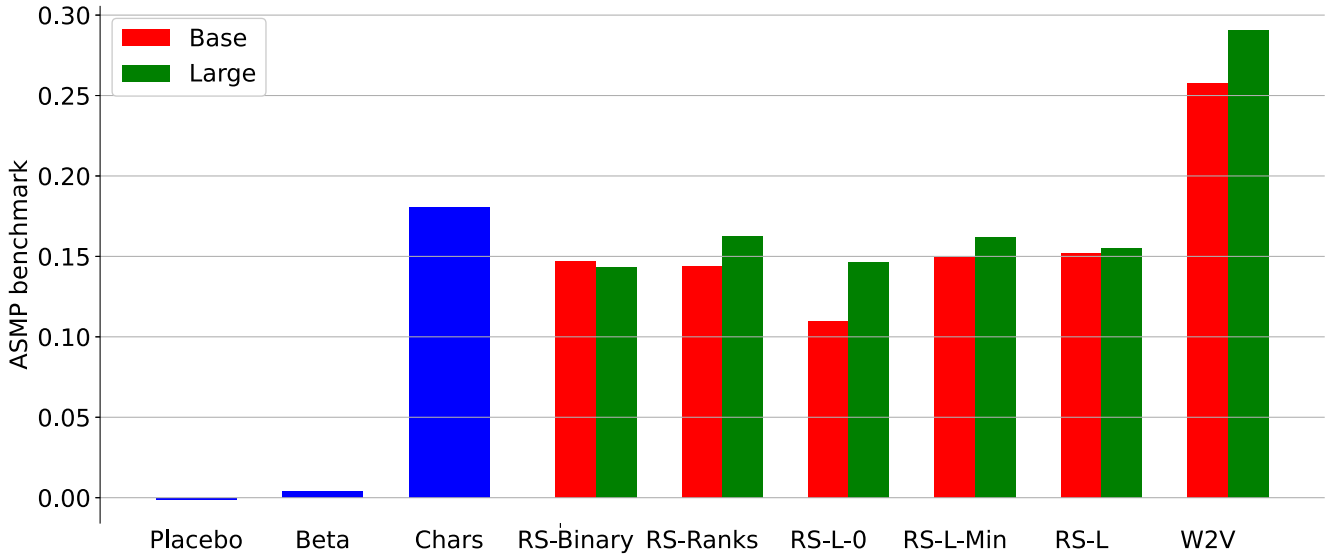


Figure 9: Asset similarity benchmark. The structure of Figure 7 is the same as Figure 3, except that we explore six characteristics to predict the missing stock: market beta, asset growth, profitability, log book-to-market ratio, log market capitalization, and momentum.



opportunity to assess the performance of the AssetBERT model.

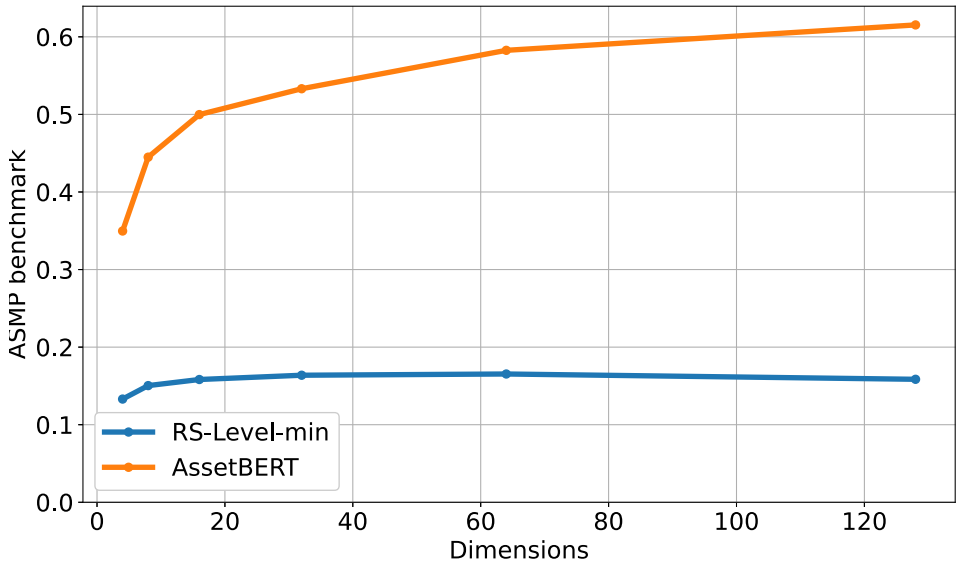
The main takeaway from Figure 8 is that high-dimensional embeddings are a powerful tool to explain missing characteristics. This may be a useful insight for the asset pricing literature exploring different methods to fill missing characteristics, where we can potentially combine characteristics and embeddings. The second observation is that the AssetBERT model also performs quite well, in particular in high dimensions. As before, we use the model's input embeddings in evaluating this benchmark, which implies that some of the refinements we discussed before (aggregating contextualized embeddings and fine-tuning the AssetBERT model on this task) can potentially further improve its performance.

6.3 Asset similarity in managed portfolios benchmark

We now provide the results for the ASMP benchmark that we introduced in Section 4.3, focusing on predicting the second stock in an investor's portfolio. We present the results for observed characteristics, recommender systems, and Word2Vec in Figure 9. The structure of the figure is the same as is the same as Figure 3, except that we explore six characteristics to predict the missing stock: market beta, asset growth, profitability, log book-to-market ratio, log market capitalization, and momentum.

Compared to the previous two benchmarks, the Word2Vec model now significantly outperforms the models with observed characteristics and recommender systems. The ASMP benchmark, which can be interpreted as an R^2 value, is 26% for the base model and 29% for the large model. All other

Figure 10: Asset similarity benchmark: High-dimensional models . The structure is the same as Figure 6.



models are below 20%.

In Figure 10, we explore the performance of AssetBERT and the RS-Level-min for larger models, as before. There are two main takeaways from the figure. First, the AssetBERT model performs better at any scale than any of the other models, including Word2Vec. Even for $K = 4$, the ASMP benchmark equals 35% already, increasing steadily to over 60%. Second, the high-dimensional recommender system cannot get close to AssetBERT in this benchmark.

The large performance gap between the models can have various explanations. First, the recommender system model may be too restrictive and the more expressive transformer model can use its flexibility to capture non-linearities better. Second, there is an extra modeling step required to map embeddings to logits, and it may be possible to generalize this model and improve the model's performance. Third, the AssetBERT model is trained specifically on this task by optimizing the model's cross-entropy, while the objective function of the recommender system is a least-squares objective. Exploring these extensions are useful directions for future research to explore.

Illustrating the AssetBERT model The AssetBERT model performs well on all benchmarks, and in particular on the ASMP benchmark. To provide some additional insight into the model's performance, we revisit the ARKK ETF discussed in Section 3.4 for the final period of our sample.

In Table 2, we report the largest 10 positions of ARKK's portfolio and we mask the third position, which is Tesla Inc. The AssetBERT model is then used to predict the identity of the missing stock. In the second column, we report the predictions from the 16-dimensional AssetBERT model. While the precise position varies across AssetBERT model configurations, Tesla Inc is consistently highly ranked for this ETF.

Table 2: Illustrating AssetBERT in case of ARKK The first column reports the largest 10 holdings of the ARKK ETF in 2022.Q4. We use the 16-dimensional AssetBERT model to predict the missing stock in position 3, which is Tesla Inc. In the second column, we report the top predictions for the missing stock.

Rank	Actual holding	Predicted holding for position 3
1	Zoom Video Communications Inc	Alphabet Inc
2	Exact Sciences Corp	Amazon Com Inc
3	[MASK]	Apple Inc
4	Roku Inc	Servicenow Inc
5	Block Inc	Adobe Inc
6	UiPath Inc	Microsoft Corp
7	Teladoc Health Inc	Advanced Micro Devices Inc
8	Twilio Inc	**Tesla Inc**
9	Beam Therapeutics Inc	Visa Inc
10	Unity Software Inc	Netflix Inc

6.4 Performance of the InvestorBERT model

We conclude this section by illustrating the performance of the InvestorBERT model for 2022.Q4. Identifying similar investors is a challenging task beyond institutional type (e.g., hedge fund, mutual fund or insurance company), size, type of benchmark, and measures of activeness (e.g., turnover and active share). Investor embeddings can be used to identify similar investors.

The benchmark is analogous to the ASMP benchmark but instead of predicting the second stock for a given investor, we predict the second investor for a given stock.

Figure 11 presents the results for this benchmark for $K = 4, 8, 16, 32, 64, 128$. As is clear from the figure, the same pattern emerges as for the AssetBERT model for the ASMP benchmark: the InvestorBERT model tends to perform well on this benchmark and high-dimensional models fare particularly well.

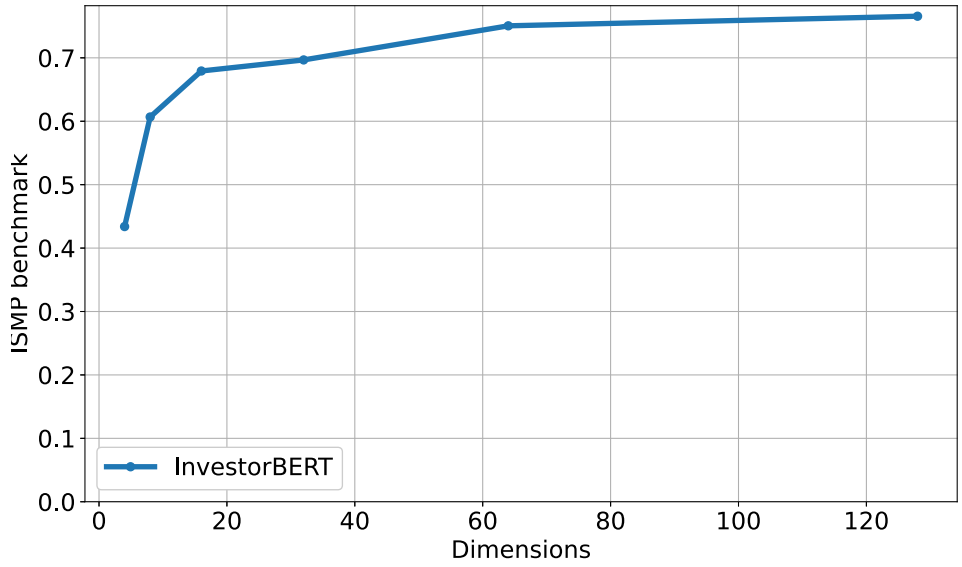
7 Extensions

In this section, we discuss a series of extensions of the core ideas developed in this paper.

7.1 Alternative embedding data sets

We explored text-based embeddings from Cohere and OpenAI, but those embeddings can potentially be improved via fine tuning using text data from, for instance, newspapers, firms' filings or earnings calls (see Hassan et al. (2019), Chen and Sarkar (2020), Bybee et al. (2023), and van Binsbergen et

Figure 11: Investor similarity benchmark for 2022.Q4.



al. (2023), among many others, for applications of such data).

More broadly, a key feature of embedding data is that it has two dimensions. In case of holdings data, the portfolio weights vary across investors and across stocks. Other examples include daily returns or volume data within a quarter. The results in Section 2 provide conditions under which these data are informative as well.

7.2 Generative portfolios: Forming factor mimicking portfolios without return data

Asset embedding models can be used to generate replicating portfolios without relying on historical data for estimation. For instance, suppose our goal is to design a "COVID asset pricing factor" that performs poorly when COVID is strong. One starting point is to be long United Airlines and short Zoom, as they are exemplar stocks for sensitivity to COVID. But this portfolio formed of only two stocks has a lot of idiosyncratic risk. So instead, one can use asset embeddings to find stocks "similar" to Zoom and United Airlines, and add them to the portfolio in order to diversify the idiosyncratic risk and obtain a "COVID asset pricing factor."

We use the 16-dimensional AssetBERT model to illustrate this application. We start from 10 firms in our sample that experienced the most negative return during 2020.Q1.[22] We report this list in the first column of Table 3. The list includes companies in industries that were heavily impacted, including airlines, energy and oil, and insurance and financial services.

Taking this list as given, we split the portfolio and mask a position in the middle (i.e., position 6

[22]To ensure that the firms are also recognizable, we require the firms have a market cap greater than $10bn at the end of 2020.Q1.

Table 3: Illustration of generative portfolios using AssetBERT. We use the 16-dimensional AssetBERT model during 2020.Q1 to identify first that are exposed to the COVID-19 pandemic. The first set columns report the 10 firms with the most negative return during this quarter. The second set of columns uses the AssetBERT model to identify the most similar firms.

	Worst Performing Stocks in 2021 Q1			Similar stocks predicted by AssetBERT		
	Company	Return	Market cap	Company	Return	Market cap
1	Occidental Petroleum Corp	-0.70	10,367	Metlife Inc	-0.39	27,997
2	Marathon Petroleum Corp	-0.60	15,353	Cummins Inc	-0.24	20,044
3	Discover Financial Services	-0.58	10,924	Wells Fargo & Co New	-0.46	117,366
4	Eog Resources Inc	-0.57	20,907	International Paper Co	-0.31	12,207
5	Boeing Co	-0.54	84,149	Prudential Financial Inc	-0.44	20,650
6	Pioneer Natural Resources Co	-0.53	11,625	General Motors Co	-0.42	29,695
7	Conocophillips	-0.52	33,167	Public Service Enterprise Gp Inc	-0.23	22,685
8	American International Group Inc	-0.52	21,180	Capital One Financial Corp	-0.51	23,063
9	Phillips 66	-0.51	23,490	Sysco Corp	-0.46	23,203
10	Valero Energy Corp New	-0.51	18,568	T Rowe Price Group Inc	-0.19	22,263

out of a portfolio of then 11 stocks). We then ask the AssetBERT model, estimated using data from 2020.Q1, to complete the portfolio. The second column reports the top predictions by the model. Most of these companies were also heavily affected by the pandemic and are in related industries.

We emphasize that this is just an illustration and that if the goal is to generate portfolios, it may be optimal to train the model using causal language modeling (in which the next stock is masked) as opposed to masked language modeling (in which a stock in the middle of the portfolio is masked). Put differently, building on the analogy to large language models, we would like to estimate a decoder model instead of an encoder model (i.e., a GPT-style model as opposed to a BERT-style model).

7.3 Applications of investor embeddings

Our main focus in this paper is on asset embeddings, x_a, and we only mention investor embeddings, λ_i, in passing. We briefly discuss three applications of investor embeddings, and leave the actual exploration of those ideas for future research.

First, investor embeddings identify investors who hold similar (active) portfolios. It is generally challenging to group investors beyond their size, how active they are, and institutional type. Investor embeddings provide a systematic approach to cluster investor types.

Second, investor embeddings can be used for performance measurement. Building on the insights of Daniel et al. (1997), who develop a characteristics-based benchmark, our investor embeddings can be used to do this in higher dimensions. Concretely, an investor i with investor embedding λ_i

Table 4: Identifying similar investors using investor embeddings. For each of the three investors, "Dimensional US Large Cap Value ETF," "iShares Exponential Technologies Index ETF", and "Virtus LifeSci Biotech Products ETF", we select the five most similar investors using investor embeddings. We use the RS-Level-min embeddings of dimension 64 in 2022.Q4.

Fund	Dimensional US Large Cap Value ETF
Rank 1	SA US Value Fund
Rank 2	Dimensional Funds ICVC - International Value Fund
Rank 3	PGIM Quant Solutions Large-Cap Value Fund
Rank 4	UBS (Irl) ETF Plc - Factor MSCI USA Prime Value ESG UCITS ETF
Rank 5	Columbia Multi Manager Value Strategies Fund
Fund	iShares Exponential Technologies Index ETF
Rank 1	Multi Units LU - Lyxor MSCI Disruptive Tech. ESG Filtered
Rank 2	LUX IM - AI & DATA
Rank 3	AtonR Fund (The)
Rank 4	Dux Umbrella FI - Trimming USA Technology
Rank 5	HANetf ICAV - HAN-GINS Innovative Technologies UCITS ET
Fund	Virtus LifeSci Biotech Products ETF
Rank 1	Global X Genomics & Biotechnology ETF
Rank 2	BNY Mellon Global Fds. Plc - Smart Cures Innovation Fund
Rank 3	WisdomTree BioRevolution Fund
Rank 4	JPMorgan Funds - Thematics - Genetic Therapies
Rank 5	JSS Investmentfonds II - Sustainable Eq. - Future Health

can be compared to investors with embedding vectors λ_j such that $\|\lambda_i - \lambda_j\|_F \leq \delta$,[23] and we can compute $\alpha_{it} = r_{it} - \frac{1}{N_{it}} \sum_{j:\|\lambda_i - \lambda_j\|_F \leq \delta} r_{jt}$ with N_{it} the number of investors that are sufficiently close to investor i.

Third, building on these insights, investor embeddings can be used to identify trading directions that are common across investors, which provides a natural measure of crowded trades. Given the market dislocations that can occur due to crowded trades (e.g., during August 2007 with quantitative equity strategies or in August 2024 related to the Japanese carry trade), developing analytical tools along these lines can be of value to investors and market regulators.

To illustrate that the investor embedding models are able to identify similar investors, we identify similar investors to several investors with different trading strategies in Table 4 for the final quarter of our sample. For three funds, "Dimensional US Large Cap Value ETF," "iShares Exponential Technologies Index ETF", and "Virtus LifeSci Biotech Products ETF." We use the RS-Level-min embeddings of dimension 64 in 2022.Q4. In all cases, the funds identified via investor embeddings

[23] We discuss how to scale and rotate embeddings for the Frobenius norm to be an economically meaningful distance measure in Appendix B.

appear to follow similar investment strategies as the selected fund.

7.4 Risk management and stress scenarios

Central banks, regulators, and financial institutions could find asset and investor embeddings useful for risk management, including stress scenarios. Indeed, Generative AI generates new pictures, texts, etc. from a initial prompts—e.g. via stable diffusion (Rombach et al. (2022)). Likewise, one could generate changes in asset and investor embeddings, that in turn lead to changes in market prices once combined with an asset demand system (Koijen and Yogo, 2019). This will generate useful stress scenarios, that include episodes never seen in real life, but are still plausible generalizations from past scenarios.

8 Using text data to interpret embeddings

To conclude the paper, we provide an approach to interpret asset embeddings (Section 8.1) and investor embeddings (Section 8.2). Our simple approach is to use LLMs to summarize information on groups of firms or investors that are similar, in order to provide and interpretation of why they cluster together. This is of course only a first step towards a deeper understanding of why firms and investors are clustered together, and we provide it as a proof-of-concept.

8.1 Interpreting asset embeddings

To understand why firms cluster together, we use data from earnings calls. We use the firms in the left column of Table 3 for which it is clear why these firms are grouped together during the pandemic, but it is perhaps less obvious why these firms would cluster together otherwise.

We start from cleaned earnings call transcripts. We then collect the earnings calls for the 10 firms in a given quarter and use them as context in an LLM, for which we use OpenAI's `gpt-4o-2024-08-06` model. We then use the following prompt (alongside the earnings calls):

> You are a sophisticated financial analyst. The above context includes transcripts of earnings calls for the following companies:
>
> {company_list}
>
> that took place during the following year quarters: {year_quarter_list}. Carefully analyze these transcripts and then identify up to three of the common, most important risks shared by these companies. Please provide specific examples and details for each risk and discuss all companies in this context.

By passing the earnings calls of a particular quarter, we can summarize the actual risks mentioned in earnings calls and limit the risk of hallucinations. In Figure 12, we display the results for the earnings calls that took place during 2019.Q4 (top section) and 2020.Q2 (bottom section). While the first set of summaries discuss rather generic risks, the second set of summaries identify the key risks to these firms during this period when COVID associated risks were especially relevant for these firms.

This basic logic can be extended in at least three ways. First, while risk considerations is one reason why investors hold stocks together in a portfolio, there are many others, e.g., similar growth opportunities, similar ESG characteristics, et cetera. We can adjust the prompt accordingly to not only list the common risk, but also growth opportunities, et cetera. Second, to identify the more unique risks among the companies of interest, we can ask the model to summarize the 10 earnings calls and ask what these 10 companies have in common that is different from another set of, say, 10 to 20 companies across various industries. This risks for the second group of firms serves as a baseline to compare the companies of interest to, and filters out the common risks that are not particularly unique to the 10 firms of interest. From a technical perspective, one may need to use models with larger context windows (e.g., Google's Gemini) or use retrieval-augmented generation (RAG) methods to handle the larger amount of data.[24] Third, our initial exploration only uses data on earnings calls. We can naturally extend the text data used to include news articles, analyst reports, et cetera.

8.2 Interpreting investor embeddings

This same logic can also be applied to investor embeddings. In this case, we can obtain descriptions of investment strategies from prospectuses Abis et al. (2022), analyst reports (e.g., from Morningstar), investor letters, et cetera.

9 Conclusion

We introduce the concept of asset and investor embeddings and argue, theoretically and empirically, that portfolio holdings are a natural source of embedding data. Just as documents are useful to uncover word structures, investors' holdings reveal asset structures or embeddings. Embeddings can be used to complement the standard set of firm characteristics that are ubiquitously used in finance and economics.

[24]The context window corresponds to the amount of text that the model can process in a single query. For OpenAI's GPT4 model, the context window is limited to 128k tokens. For Anthropic's Claude 3.5 Sonnet model, the context window is 200k tokens, and for Google's Gemini 1.5 Pro model, it is 2m tokens. A rough estimate is that one token generally corresponds to about 0.75 word in English.

Figure 12: Interpreting asset embeddings. The first set of results is for earnings calls that took place in 2019.Q4. The second set of results is for earnings calls that occurred in 2020.Q2.

Based on the analysis of the earnings call transcripts for the companies listed, three common, significant risks shared by these companies are:

1. **Commodity Price Volatility:**
    - **Occidental Petroleum Corp., Marathon Petroleum Corp., EOG Resources, Inc., Pioneer Natural Resources Co., ConocoPhillips, Phillips 66, and Valero Energy Corp.**: These companies are heavily exposed to fluctuations in oil and gas prices, which can significantly impact their revenues and profitability. For instance, Occidental Petroleum and ConocoPhillips discussed the impact of oil price volatility on their financial performance and strategic decisions. Similarly, Valero Energy and Phillips 66 highlighted how refining margins and crude differentials, which are influenced by global oil prices, affect their earnings. The uncertainty in commodity prices can lead to unpredictable cash flows and necessitate adjustments in capital expenditure and operational strategies.
2. **Regulatory and Environmental Risks:**
    - **The Boeing Co., American International Group, Inc., and Valero Energy Corp.**: These companies face significant regulatory and environmental risks that can affect their operations and financial performance. Boeing, for example, is dealing with regulatory scrutiny and safety concerns related to the 737 MAX, which has led to production halts and reputational damage. American International Group (AIG) is subject to regulatory changes in the insurance industry, which can impact its product offerings and profitability. Valero Energy and other energy companies are also navigating environmental regulations, such as those related to emissions and renewable fuels, which can lead to increased compliance costs and operational adjustments.
3. **Market and Economic Uncertainty:**
    - **Discover Financial Services, The Boeing Co., and American International Group, Inc.**: These companies are exposed to broader market and economic uncertainties that can affect consumer demand, investment decisions, and financial stability. Discover Financial Services, for instance, is sensitive to changes in consumer credit behavior and economic conditions that influence loan growth and credit quality. Boeing's performance is tied to global air travel demand, which can be impacted by economic downturns and geopolitical tensions. AIG's insurance business is also influenced by economic cycles, which affect underwriting results and investment income. These uncertainties require companies to be agile in their strategic planning and risk management practices.

Overall, these risks highlight the importance of strategic flexibility, robust risk management, and proactive regulatory compliance for these companies to navigate the challenges in their respective industries.

The earnings call transcripts for the companies listed above reveal several common risks that these companies faced during the second quarter of 2020. Here are three of the most important shared risks:

1. **Impact of COVID-19 on Demand and Operations:**
    - **Occidental Petroleum Corp., Marathon Petroleum Corp., and Valero Energy Corp.:** These companies highlighted the significant decline in demand for oil and refined products due to the COVID-19 pandemic. Occidental Petroleum noted the unprecedented drop in oil prices and demand, which led to operational adjustments and financial measures to preserve liquidity. Marathon Petroleum and Valero Energy also discussed the impact of reduced travel and economic activity on gasoline and jet fuel demand, leading to lower refinery utilization rates.
    - **The Boeing Co.:** Boeing faced a sharp decline in demand for commercial aircraft as airlines reduced operations due to travel restrictions and decreased passenger traffic. The company had to adjust production rates and manage supply chain disruptions.
    - **Discover Financial Services and American International Group, Inc.:** These financial companies experienced challenges related to the economic downturn caused by the pandemic. Discover Financial Services noted increased credit risk and the need for higher loan loss provisions, while AIG faced potential claims related to business interruption and other insurance lines.
2. **Financial Liquidity and Capital Management:**
    - **ConocoPhillips and EOG Resources, Inc.:** Both companies emphasized the importance of maintaining strong liquidity and financial flexibility in response to the volatile oil market. ConocoPhillips highlighted its strong balance sheet and liquidity position, while EOG Resources focused on cost reductions and capital discipline to preserve cash flow.
    - **Phillips 66 and Pioneer Natural Resources Co.:** These companies took measures to conserve cash and maintain liquidity, such as reducing capital expenditures and suspending share repurchases. Phillips 66 secured additional credit facilities to enhance its financial position, while Pioneer Natural Resources emphasized its strong balance sheet and cost-cutting efforts.
    - **American International Group, Inc.:** AIG discussed its focus on liquidity and capital strength, highlighting its actions to manage financial resources prudently during the crisis.
3. **Supply Chain and Operational Disruptions:**
    - **The Boeing Co.:** Boeing faced significant supply chain disruptions due to the pandemic, affecting its production schedules and delivery timelines. The company had to work closely with suppliers to manage these challenges and ensure business continuity.
    - **Valero Energy Corp. and Marathon Petroleum Corp.:** These companies experienced operational disruptions as they adjusted refinery operations to match reduced demand. Valero Energy discussed the need to balance supply with demand to avoid inventory build-up, while Marathon Petroleum highlighted the impact of lower utilization rates on its operations.
    - **Occidental Petroleum Corp. and ConocoPhillips:** Both companies had to navigate supply chain challenges related to oilfield services and equipment availability, as well as manage production curtailments in response to market conditions.

Overall, these companies faced significant risks related to the COVID-19 pandemic's impact on demand, financial liquidity, and supply chain disruptions. Each company took specific actions to mitigate these risks and adapt to the rapidly changing environment.

Our current applications are in equity markets, but the same ideas can be applied to other asset classes such as fixed income markets, commodities, and currency markets as holdings data are becoming widely available across asset classes and geographies, including for US households (Gabaix et al. (2022)).

Our conceptual insight that holdings data contain key information about asset characteristics or embeddings also provides a natural connection between finance (and economics more broadly) and recent methodological advances in the fields of machine learning and artificial intelligence that we are exploring in ongoing work.

References

Abis, Simona, Andrea M Buffa, Apoorva Javadekar, and Anton Lines, "Learning from Prospectuses," 2022. Working paper.

Balasubramaniam, Vimal, John Y. Campbell, Tarun Ramadorai, and Benjamin Ranish, "Who Owns What? A Factor Model for Direct Stockholding," *Journal of Finance*, 2023, *78* (3), 1545–1591.

Beckmeyer, Heiner and Timo Wiedemann, "Recovering Missing Firm Characteristics with Attention-based Machine Learning," 2023. Working paper.

Betermier, Sebastien, Laurent E. Calvet, Samuli Knüpfer, and Jens Soerlie Kvaerner, "What Do the Portfolios of Individual Investors Reveal About the Cross-Section of Equity Returns?," 2022. Working paper.

Brandt, Michael W., Pedro Santa-Clara, and Rossen Valkanov, "Parametric Portfolio Policies: Exploiting Characteristics in the Cross-Section of Equity Returns," *Review of Financial Studies*, 2009, *22* (9), 3411–3447.

Bryzgalova, Svetlana, Sven Lerner, Martin Lettau, and Markus Pelger, "Missing Financial Data," 2022. Working paper.

Bryzgalova, Svetlana, Victor DeMiguel, Sicong Li, and Markus Pelger, "Asset-Pricing Factors with Economic Targets," *Working paper*, 2023.

Bybee, Leland, Bryan Kelly, and Yinan Su, "Narrative Asset Pricing: Interpretable Systematic Risk Factors from News Text," *The Review of Financial Studies*, 2023.

Campbell, John Y. and Tuomo Vuolteenaho, "Bad Beta, Good Beta," *American Economic Review*, 2004, *94* (5), 1249–1275.

Chen, Andrew Y. and JackMcCoy, "Missing values handling for machine learning portfolios," *Journal of Financial Economics*, 2024, *155.*

Chen, Jiafeng and Suproteem K. Sarkar, "A Semantic Approach to Financial Fundamentals," 2020. Working paper.

Chen, Luyang, Markus Pelger, and Jason Zhu, "Deep learning in asset pricing," *Management Science*, 2023.

Cong, Lin William, Ke Tang, Jingyuan Wang, and Yang Zhang, "AlphaPortfolio: Direct Construction Through Deep Reinforcement Learning and Interpretable AI," 2022. Working paper.

Daniel, Kent, Mark Grinblatt, Sheridan Titman, and Russ Wermers, "Measuring Mutual Fund Performance with Characteristic-Based Benchmarks," *Journal of Finance*, 1997, *52* (3), 1035–1058.

Deerwater, Scott, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman, "Indexing by Latent Semantic Analysis," *Journal of the American Society for Information Science*, 1990, *41* (6), 391–407.

DeMiguel, Victor, Alberto Martin-Utrera, Francisco J. Nogales, and Raman Uppal, "A Transaction-Cost Perspective on the Multitude of Firm Characteristics," *Review of Financial Studies*, 2020, *33*, 2180–2222.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *Proceedings of NAACL-HLT 2019*, 2019.

Dolphin, Rian, Barry Smyth, and Ruihai Dong, "Stock Embeddings: Learning Distributed Representations for Financial Assets," *Working paper*, 2022.

Dubinsky, Andrew, Michael Johannes, Andreas Kaeck, and Norman J Seeger, "Option Pricing of Earnings Announcement Risks," *Review of Financial Studies*, 2019, *32* (2), 646–687.

Ducharme, Yoshua Bengio Rejean, Pascal Vincent, and Christian Jauvin, "A Neural Probabilistic Language Model," *Journal of Machine Learning Research*, 2003, *3*, 1137–1155.

Dumais, Susan T, George W Furnas, Thomas K Landauer, Scott Deerwester, and Richard Harshman, "Using latent semantic analysis to improve access to textual information," in "Proceedings of the SIGCHI conference on Human factors in computing systems" 1988, pp. 281–285.

Fama, Eugene F. and Kenneth R. French, "A Five-Factor Asset Pricing Model," *Journal of Financial Economics*, 2015, *116* (1), 1–22.

Feng, Guanhao, Stefano Giglio, and Dacheng Xiu, "Taming the Factor Zoo: A Test of New Factors," *Journal of Finance*, 2020, *75*, 1327–1370.

Freyberger, Joachim, Andreas Neuhierl, and Michael Weber, "Dissecting Characteristics Nonparametrically," *Review of Financial Studies*, 2020, *33*, 2326–2377.

Freyberger, Joachim, Bjorn Hoppner, Andreas Neuhierl, and Michael Weber, "Missing Data in Asset Pricing Panels," 2022. Working paper.

Gabaix, Xavier and Ralph SJ Koijen, "In search of the origins of financial fluctuations: The inelastic markets hypothesis," *NBER Working Paper No. 28967*, 2022.

Gabaix, Xavier and Ralph SJ Koijen, "Granular instrumental variables," Technical Report 7 2024.

Gabaix, Xavier, Ralph S.J. Koijen, Federico Mainardi, Sangmin Oh, and Motohiro Yogo, "Asset Demand of U.S. Households," *Working paper*, 2022.

Gu, Shihao, Bryan Kelly, and Dacheng Xiu, "Empirical asset pricing via machine learning," *The Review of Financial Studies*, 2020, *33* (5), 2223–2273.

Hansen, Lars Peter and Scott F Richard, "The role of conditioning information in deducing testable restrictions implied by dynamic asset pricing models," *Econometrica: Journal of the Econometric Society*, 1987, pp. 587–613.

Hassan, Tarek A, Stephan Hollander, Laurence Van Lent, and Ahmed Tahoun, "Firm-level political risk: Measurement and effects," *The Quarterly Journal of Economics*, 2019, *134* (4), 2135–2202.

Hoberg, Gerard and Gordon Phillips, "Text-Based Network Industries and Endogenous Product Differentiation," *Journal of Political Economy*, 2016, *124* (5), 1423–1465.

Hommel, Nicolas, Augustin Landier, and David Thesmar, "Corporate Valuation: An Empirical Comparison of Discounting Methods," 2023. Working paper.

Jensen, Theis Ingerslev, Bryan Kelly, and Lasse Heje Pedersen, "Is There a Replication Crisis in Finance?," *The Journal of Finance*, 2023, *78* (5), 246–2518.

Jurafsky, Dan and James H. Martin, *Speech and Language Processing (3rd ed. draft)* 2023.

Kaplan, Jared, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei, "Scaling Laws for Neural Language Models," 2020.

Kelly, Bryan and Dacheng Xiu, "Financial Machine Learning," 2023. Working paper.

Kelly, Bryan T., Seth Pruitt, and Yinan Su, "Characteristics are covariances: A unified model of risk and return," *Journal of Financial Economics*, 2019, *134*, 501–524.

Koijen, Ralph SJ and Motohiro Yogo, "A demand system approach to asset pricing," *Journal of Political Economy*, 2019, *127* (4), 1475–1515.

Koijen, Ralph SJ, Robert J Richmond, and Motohiro Yogo, "Which Investors Matter for Equity Valuations and Expected Returns?," *Working Paper*, 2022.

Kozak, Serhiy, Stefan Nagel, and Shrihari Santosh, "Interpreting Factor Models," *Journal of Finance*, 2018, *73* (3), 1183–1223.

Kozak, Serhiy, Stefan Nagel, and Shrihari Santosh, "Shrinking the cross-section," *Journal of Financial Economics*, 2020, *135*, 271–292.

Lettau, Martin and Markus Pelger, "Factors That Fit the Time Series and Cross-Section of Stock Returns," *Review of Financial Studies*, 2020, *33*, 2274–2325.

Madhavan, Ananth, Aleksander Sobczyk, and Andrew Ang, "What Happens With More Funds Than Stocks?," *Journal of Investment Management*, 2021, *19* (2), 4–28.

McInnes, Leland, John Healy, and James Melville, "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction," 2018. Working paper.

McInnes, Leland, John Healy, and James Melville, "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction," 2020.

Michel, Paul, Omer Levy, and Graham Neubig, "Are Sixteen Heads Really Better than One?," *33rd Conference on Neural Information Processing Systems*, 2019.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean, "Distributed Representations of Words and Phrases and their Compositionality," *NIPS*, 2013, pp. 3111–3119.

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean, "Efficient Estimation of Word Representations in Vector Space," *ICLR Workshop Papers*, 2013.

Nagel, Stefan, *Machine learning in asset pricing*, Vol. 8, Princeton University Press, 2021.

Pennington, Jeffrey, Richard Socher, and Christopher D. Manning, "GloVe: Global Vectors for Word Representation," *Conference on Empirical Methods on Natural Language Processing*, 2014.

Prince, Simon J.D., *Understanding Deep Learning*, MIT Press, 2023.

Radford, Alec, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever, "Improving Language Understanding by Generative Pre-Training," *OpenAI*, 2018.

Rombach, Robin, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer, "High-resolution image synthesis with latent diffusion models," in "Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition" 2022, pp. 10684–10695.

Ross, Stephen A., "The Arbitrage Theory of Capital Asset Pricing," *Journal of Economic Theory*, 1976, *13* (3), 341–360.

Sarkar, Suproteem and Keyon Vafa, "Lookahead Bias in Pretrained Language Models," 2024. Working paper.

van Binsbergen, Jules H., Svetlana Bryzgalova, Mayukh Mukhopadhyay, and Varun Sharma, "(Almost) 200 Years of News-Based Economic Sentiment," 2023. Working paper.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention Is All You Need," *31st Conference on Neural Information Processing Systems*, 2017.

Vuolteenaho, Tuomo, "What Drives Firm-Level Stock Returns?," *Journal of Finance*, 2002, *57* (1), 233–264.

A Portfolio holdings as embedding data

A.1 Equilibrium asset prices

Starting from the model in (1) and (2), we solve for asset prices by imposing market clearing. As we normalized the number of shares of asset a to 1:

$$\sum_i H_{ia} = P_a.$$

To obtain closed-form solutions, we linearize the market clearing condition around a long-run equilibrium of demand and valuations, $(\bar{h}_{ia}, \bar{p}_a)$. We denote the long-run portfolio share by $\bar{w}_a$, $\sum_a \bar{w}_a = 1$, and log dollar value of assets by $\bar{a}_i$, implying $\bar{h}_{ia} = \bar{a}_i + \ln \bar{w}_a$. This implies for long-run valuations $\sum_i \bar{H}_{ia} = \bar{P}_a$, or, equivalently, $\bar{p}_a = \ln \bar{w}_a + \ln \sum_i \exp(\bar{a}_i)$. If we linearize the first-order condition and solve for prices, we obtain[25]

$$p_a = c^p + \frac{\nu_{S^a a}}{\zeta_{S^a}} = c^p + \frac{1}{\zeta_{S^a}} \lambda'_{S^a} x_a + u_{S^a a}, \tag{8}$$

where $X_S \coloneqq \sum_i S_i X_i$ is the average of a variable X_i, weighing by importance S_i of investor i; $S_i^a = \frac{\exp(\bar{h}_{ia})}{\sum_j \exp(\bar{h}_{ja})}$ is a specific measure that of importance, namely the fraction of asset a held by investor i, and c^p is an unimportant constant. In what follow, to simplify the notations, and replace S_i^a by S_i.

This shows that we can recover embeddings from valuations and returns. However, to do so, we need a longer time period and assume that the loadings, $\frac{1}{\zeta_S}\lambda_S^\nu$, and the embeddings, x_a, do not change over time. Holdings data do not require this restriction, as we will show next.

A.2 Equilibrium portfolio holdings

We substitute (8) into (1) and obtain

$$h_{ia} = \phi_i^h + \phi_a^h + \lambda_i' x_a + \epsilon_{ia}, \tag{9}$$

[25] We start from a linear approximation of the market clearing equation

$$\sum_i \exp(\bar{h}_{ia}) + \sum_i \exp(\bar{h}_{ia})(h_{ia} - \bar{h}_{ia}) = \exp(\bar{p}_a) + \exp(\bar{p}_a)(p_a - \bar{p}_a),$$

implying $p_a = \bar{p}_a + h_{Sa} - \bar{h}_{Sa} = \ln \sum_i \exp(\bar{a}_i) - \bar{a}_S + c_S^h + (1 - \zeta_S) p_a + \nu_{Sa}$ and thus

$$p_a = \frac{\ln \sum_i \exp(\bar{a}_i) - \bar{a}_S + c_S^h}{\zeta_S} + \frac{\nu_{Sa}}{\zeta_S}.$$

where $\phi_i^h = (1-\zeta_i)\, c^p$, $\phi_a^h = \frac{1}{\zeta_S}\left(u_{Sa} + \lambda_S^{\nu\prime} x_a\right)$, $\epsilon_{ia} = u_{ia} - \frac{\zeta_i}{\zeta_S} u_{Sa}$, and $\lambda_i = \lambda_i^\nu - \frac{\zeta_i}{\zeta_S}\lambda_S^\nu$.

Hence, the recoverable investor embedding from holdings data is not λ_i^ν but its close cousin $\lambda_i = \lambda_i^\nu - \frac{\zeta_i}{\zeta_S}\lambda_S^\nu$: it captures the differential sensitivity of investor i to the asset characteristics x_a compared to the average sensitivity of the other investors, which is the λ_S^ν term. This includes the differential price elasticity, which is captured by $\frac{\zeta_i}{\zeta_S}$. If all investors were identical (up to size) we would have $\lambda_i = 0$.

When demand elasticities are the same across investors, $\zeta_i = \zeta$, then then term $\frac{\zeta_i}{\zeta_S} u_{Sa}$ in ϵ_{ia} only varies across assets and is part of ϕ_a^h instead.[26] We then obtain a pure factor model. Otherwise, there is some small contamination from investors trading against idiosyncratic demand shocks of other investors.

The main insight is that as long as investors disagree on how asset embeddings affect risk and expected return, or use asset embeddings in building portfolios for non-pecuniary reasons, portfolio holdings across investors are perfectly suited to extract embeddings. Investors who take bold bets and disagree more are particularly well suited to extract asset embeddings.

A.3 Returns, volume, and portfolio rebalancing

In addition to portfolio holdings, rebalancing is informative as well. This is particularly true if some holdings are slow to adjust and rebalancing provides more accurate information. We first study quarterly returns, $r_{aq} = \Delta p_{aq}$ — we omit dividends for simplicity:

$$r_{aq} = \frac{1}{\zeta_S}\Delta\left(\lambda_{Sq}^{\nu\prime} x_{aq}\right) + \Delta u_{Saq}.$$

If embeddings are relatively stable over time, $\Delta x_{aq} \simeq 0$, then $\Delta\left(\lambda_{Sq}^{\nu\prime} x_{aq}\right) = \left(\Delta\lambda_{Sq}^{\nu}\right)' x_{a,q-1}$ and

$$r_{aq} = \frac{1}{\zeta_S}\left(\Delta\lambda_{Sq}^{\nu}\right)' x_{a,q-1} + \Delta u_{Saq}. \tag{10}$$

This implies that factor models of returns can also be used to uncover embeddings that are fairly stable over time. Instead of using quarterly returns, we can use daily returns (or, indeed, volume data) as embedding data, having provided a micro foundation here.

Following the same logic, we can derive an expression for investors' portfolio rebalancing

$$\Delta h_{iaq} = \Delta\phi_{iq}^h + \Delta\phi_{aq}^h + \Delta\lambda_{iq}' x_{a,q-1} + \Delta\epsilon_{iaq}. \tag{11}$$

As is clear from (10) and (11), we can also estimate asset embeddings based on portfolio rebalancing, $\Delta q_{ia} = \Delta h_{ia} - r_{aq}$.

[26] Also, the term $-\frac{\lambda_S^\nu}{\zeta_S}\zeta_i = -\lambda_S^\nu$ in λ_i no longer varies across investors or assets, and the term $-\lambda_S^{\nu\prime} x_a$ therefore also folds into ϕ_a^h.

B Distances between embeddings, and canonical rotations of embeddings

We want to answer the question: what's a "natural" distance between embeddings? We start with the case of asset embeddings, and then move on to investor embeddings. This leads to a proposal for a "canonical rotation" of embeddings, which is unique (after choosing some weighting scheme).

Notations. We sometimes use $\langle \cdot \rangle$ for an average, e.g. $\langle x \rangle = \frac{1}{A} \sum_a x_a$.

B.1 Distance between assets

In general, we want to weight the components of X_a by an "importance" measure, and ensure that they are invariant by arbitrary dilation. Indeed, an unweighted measure like $\|X_a\|^2 = \sum_{k=1}^K X_{ak}^2$ is typically economically meaningless: if one rescaled the first component X_{a1} of all $X_a \in \mathbb{R}^K$ bya factor 10, for instance, ("dilates" it by 10), then $\|X_a\|^2$ changes, and indeed changes the ordering it induces.[27] But the economic quantities we consider should not change — "dilation invariance" should be respected. We present a norm that preserves dilation invariance, because it answers a well-posed economic question.

Suppose that we have a model for $\theta_{ia}(x_a)$ – how θ_{ia} depends on x_a (and the rest of the market). Then, if an asset's embedding changes by a small δx, i.e. moves from x_a to $x_b = x_a + \delta x$, the loading moves by $\delta \ln \theta_{ia} = \lambda_i \delta x$, where

$$\lambda_i = \frac{\partial \ln \theta_{ia}}{\partial x_a} \tag{12}$$

a K-dimensional vector. We now assume that λ_i is independent of a, and revisit that complication soon afterwards. Hence, log portfolio share moves by $\delta \ln \theta_{ia} = \lambda_i' \delta x$, so squaring, by $(\delta \ln \theta_{ia})^2 = \delta x' \lambda_i \lambda_i' \delta x$. We average over institutions, and get

$$\frac{1}{I} \sum_i (\delta \ln \theta_{ia})^2 = \delta x' M \delta x,$$

with the matrix

$$M_x = \frac{1}{I} \sum_i \lambda_i \lambda_i'. \tag{13}$$

Then, the distance between two assets can be defined to be:

$$d_{ab} = d(x_a, x_b) = \left((x_a - x_b)' M (x_a - x_b) \right)^{1/2}. \tag{14}$$

[27] For instance, the answer to the question "which of the two assets, a, b has the smallest norm?" changes after dilation, as we can have $X_{a1}^2 + X_{a2}^2 < X_{b1}^2 + X_{b2}^2$ but $100X_{a1}^2 + X_{a2}^2 > 100X_{b1}^2 + X_{b2}^2$.

Defining[28]

$$\tilde{x}_a = M^{1/2} x_a, \tag{15}$$

this can also be expressed

$$d_{ab} = \|\tilde{x}_a - \tilde{x}_b\| = \left((\tilde{x}_a - \tilde{x}_b)' (\tilde{x}_a - \tilde{x}_b) \right)^{1/2}. \tag{16}$$

The "intrinsic" interpretation is that d_{ab} is the average relative distance between the log holdings of a and b, due to the characteristics x_a and x_b, averaged across investors (and using a quadratic mean) — when the characteristics are "close:"

$$d_{ab} = \|\tilde{x}_a - \tilde{x}_b\| \simeq \left\langle (\ln \theta_{ia} - \ln \theta_{ib})^2 \right\rangle^{1/2} \text{ when } \|\tilde{x}_a - \tilde{x}_b\| \to 0. \tag{17}$$

This suggest to use a distance "controlling for size," where the "active" portfolio shares would matter, rather than the simple portfolio shares.

In the recommender system, this is automatically taken care of by including investor and stock fixed effects. For the non-linear models (e.g., Word2Vec and AssetBERT), we filter the data first and estimate $h_{ia} = \delta_i + \delta_a + h^*_{ia}$, and estimate the model on (ranks of) h^*_{ia}. To obtain λ_i for those models, we simply regress h_{ia} (or the percentile ranks) on x_a (perhaps with a ridge penalty). Or, we can calculate the derivative (given a nonlinear system) of $\lambda_i = \left\langle \frac{\partial h^*_{ia}}{\partial x_a} \right\rangle_{|i}$, averaging across assets.

Similarity between two assets The similarity of two assets could be defined as:

$$s_{ab} = \cos(\tilde{x}_a, \tilde{x}_b) = \frac{\tilde{x}'_a \tilde{x}_b}{\|\tilde{x}_a\| \, \|\tilde{x}_b\|}. \tag{18}$$

Defining $\bar{\tilde{x}} = \frac{1}{A} \sum_a \tilde{x}_a$, it could also be defined as:

$$s_{ab} = \cos(\tilde{x}_a - \bar{\tilde{x}}, \tilde{x}_b - \bar{\tilde{x}}). \tag{19}$$

This way, the measure is invariant by arbitrary shifts in all x_a.

Normalizing the average squared distance to 1 Now, we can rescale the distance, so that the average squared distance between assets is 1. To do so, we construct, $\bar{x} = \frac{1}{A} \sum_a x_a$, and the average squared distance to the center, σ^2. Then replace M by

$$\tilde{M} = \frac{1}{2\sigma^2} M, \qquad \sigma^2 = \frac{1}{A} \sum_a d(x_a, \bar{x})^2 \tag{20}$$

[28] We construct $M^{1/2}$ as follows: $M^{1/2} = UD^{1/2}U'$, with $M = UDU'$ the diagonalization of symmetric matrix M, with D diagonal and U an orthogonal matrix. Note that $M^{1/2}$ is symmetric.

Then, the average squared between two assets is 1:[29]

$$\left\langle \|\tilde{x}_a - \tilde{x}_b\|^2 \right\rangle = 1$$

Normalizing the components to have variance 1, and weighted distance Another variant is to ensure that, for each component k, the average square value of variance of $\tilde{X}_{ak}$ is 1. The construction is simply to calculate $\sigma_k^2 = var\left(\tilde{x}_{ak}\right)$, set $\tilde{X}_{ak} \coloneqq \frac{1}{\sigma_k}\tilde{x}_{ak}$. Then, the distance is

$$d_{ab} = \sum_k \left(\tilde{X}_{ak} - \tilde{X}_{bk}\right)^2 \sigma_k^2 \tag{21}$$

This basis, where for each $var\left(\tilde{X}_{ak}\right) = 1$, may be useful. There are robust variants, e.g. ensuring that the IQR of the $\tilde{X}_{ak}$ is 1.

Variants on the matrix We can have variants, e.g. weighting by institution size, as in

$$M_x = \sum_i S_i \lambda_i \lambda_i' \tag{22}$$

If the derivative $\lambda_{ia} = \frac{\partial \ln \theta_{ia}}{\partial x_a}$ does depend on a, we can also take the average over asset:

$$M_x = \langle \lambda_{ia} \lambda_{ia}' \rangle_{i,a:\theta_{ia}\neq 0} = \frac{\sum_{i,a:\theta_{ia}\neq 0} \lambda_{ia}\lambda_{ia}'}{\sum_{i,a:\theta_{ia}\neq 0}} \tag{23}$$

and of course weighted variants, e.g. weighing by S_i or $S_i S_a$.

One could also use the extensive margin, rather than the intensive margin. Let us model probability that institution i owns asset a is: $p_a^i = k_i e^{x_a' \beta_i}$, where in (39), $\beta_i = W_t \bar{x}_i \in \mathbb{R}^K$. For a general model, not necessarily of the above form, we can set $\beta_i = \frac{\partial \ln p_a^i}{\partial x_a}$ evaluated at the average embedding for fund i, $\bar{x}_i$. Then, one can do the same construction, but replacing the intensive margin sensitivity λ_i by the extensive margin sensitivity β_i.

B.2 Distance between investor embeddings

The reasoning for the distance between investors is completely the same as for assets. If an investor changes policy λ_i to $\lambda_j = \lambda_i + \delta\lambda$, his position changes to $\delta \ln \theta_{ia} = \delta\lambda' x_a$, where

$$x_a = \frac{\partial \ln \theta_{ia}}{\partial \lambda_i}. \tag{24}$$

[29] Indeed, $\left\langle \|\tilde{x}_a - \tilde{x}_b\|^2 \right\rangle = \left\langle \|\tilde{x}_a - \bar{\tilde{x}}\|^2 + \|\tilde{x}_b - \bar{\tilde{x}}\|^2 \right\rangle = 2\left\langle \|\tilde{x}_b - \bar{\tilde{x}}\|^2 \right\rangle = 2 \times \frac{1}{2} = 1$

So, the mean change across positions is

$$\frac{1}{A}\sum_a \left(\delta \ln \theta_{ia}\right)^2 = \frac{1}{A}\sum_a \delta\lambda' \left(x_a x_a'\right) \delta\lambda.$$

Hence, we take the distance

$$d\left(\lambda_i, \lambda_j\right) = \left(\left(\lambda_i - \lambda_j\right)' M_\lambda \left(\lambda_i - \lambda_j\right)\right)^{1/2}, \tag{25}$$

with

$$M_\lambda = \frac{1}{A}\sum_a \left(x_a x_a'\right)^2. \tag{26}$$

As for the distance between assets, we can also have variants, e.g. we use the weighting matrix:

$$M_\lambda = \frac{\sum_{i,a:\theta_{ia}\neq 0} \left(\frac{\partial \ln \theta_{ia}}{\partial \lambda_i}\left(\frac{\partial \ln \theta_{ia}}{\partial \lambda_i}\right)'\right)^2}{\sum_{i,a:\theta_{ia}\neq 0} 1}.$$

We can also normalize so that the average distance between two investors is normalized to 1 (in squared norm), exactly like we did for the asset embeddings in (20).

B.3 Evaluating the performance of distance measures

Suppose a distance d_{ab}. We could evaluate its performance (relative to other distances), via running the following regressions, and calculating the R^2 (a high R^2 is good).

The first set of regressions captures the idea that "a low distance should lead to similar holdings":

$$\ln\left\langle \left(\ln \theta_{ia}^{active} - \ln \theta_{ib}^{active}\right)^2 \right\rangle = \alpha + \beta \ln d_{ab} \tag{27}$$

where $\theta_{ia}^{active} = \frac{\theta_{ia}}{\theta_{ia}^{\text{Market Weights}}}$. In our baseline model, we expect $\beta = 2$. This could also be done on the extensive margin:

$$\ln\left\langle \left(1_{\theta_{ia}\neq 0} - 1_{\theta_{ib}\neq 0}\right)^2 \right\rangle = \alpha + \beta \ln d_{ab} \tag{28}$$

The second set of regressions capture the idea that "a low distance leads to similar returns": we hope for a high R^2 in:

$$\ln \frac{\mathbb{E}\left[\left(r_{at} - r_{bt}\right)^2\right]}{\sigma_{r_a}^2 + \sigma_{r_b}^2} = \alpha + \beta \ln d_{ab} \tag{29}$$

We might do variants, e.g. restricting the samples only to the lowest quartiles of value on the left-hand side.

Note that the same could be transposed for investor embeddings, inverting the roles of investor

and asset embeddings. One could also mix everything:

$$\ln \left\langle \left(\ln \theta_{ia}^{active} - \ln \theta_{jb}^{active} \right)^2 \right\rangle = \alpha + \beta \ln d_{ab} + \gamma \ln d_{ij}$$

and in our baseline models, we expect $\beta = \gamma = 2$.

C Recommender system: ALS algorithm

We summarize the alternating least-squares algorithm that we use to estimate the recommender system.[30] The goal is to solve the following optimization problem over $\theta = (\delta_{iq}, \delta_a, x_a, \lambda_{iq}, \delta_t, \beta_t)$,

$$\min_\theta \kappa_1 \sum_{i,a,q} (h_{iaq} - \delta_{iq}^\lambda - \delta_a^x - x_a' \lambda_{iq})^2 + \kappa_2 \sum_{t,a} (y_{at} - \delta_t^y - \beta_t' x_a)^2 + \xi_x \sum_a x_a' x_a + \xi_\lambda \sum_{i,q} \lambda_{iq}' \lambda_{iq} + \xi_y \sum_t \beta_t' \beta_t,$$

with $\kappa_1 = \frac{1-\kappa}{N_h \sigma_h^2}$ and $\kappa_2 = \frac{\kappa}{N_y \sigma_y^2}$, where N_h and N_y are the number of non-missing values in h and y. ξ_x and ξ_λ are regularization parameters for asset and investor embeddings, respectively, which form the model's hyper-parameters together with κ_1 and κ_2.[31] We choose ξ_x and ξ_λ proportional to $\frac{1}{AK}$ and $\frac{1}{IK}$, respectively. The summations only use non-missing observations, and we avoid imputing missing values with zeros.

We define $x_a^\delta = (\delta_a^x, x_a')'$, $\hat{x}_a^\delta = (1, \delta_a^x, x_a')'$, $\hat{x}_a = (1, x_a')'$, $\lambda_{iq}^\delta = (\delta_{iq}^\lambda, \lambda_{iq}')'$, $\hat{\lambda}_{iq}^\delta = (\delta_{iq}^\lambda, 1, \lambda_{iq}')'$, $\beta_{0t} = (0, \beta_t')'$, and $\hat{\beta}_t = (\delta_t^h, \beta_t')'$. The objective function is then given by

$$\min_\theta \kappa_1 \sum_{i,a,q} m_{iaq}^h (h_{iaq} - \hat{x}_a^{\delta'} \hat{\lambda}_{iq}^\delta)^2 + \kappa_2 \sum_{t,a} m_{at}^y (y_{at} - \hat{\beta}_t' \hat{x}_a) + \xi_x \sum_a x_a' x_a + \xi_\lambda \sum_{i,q} \lambda_{iq}' \lambda_{iq} + \xi_y \sum_t \beta_t' \beta_t,$$

where $m_{iaq}^h = 1$ when h_{iaq} is observed and $m_{iaq}^h = 0$ otherwise. Analogously, $m_{at}^y = 1$ when y_{at} is observed and $m_{at}^y = 0$ otherwise. We then set the missing values in h_{iaq} and y_{at} to zero.

We then implement the following steps until convergence

1. λ_{iq}^δ: The FOC is given by

$$-\kappa_1 \sum_a m_{iaq}^h (h_{iaq} - \delta_a^x - \hat{x}_a' \lambda_{iq}^\delta) \hat{x}_a + D_0(\xi_\lambda) \lambda_{iq}^\delta = 0,$$

   where $D_0(\xi_\lambda)$ is a diagonal elements with ξ_λ as its elements, except the (1,1)-element that equals zero. This is solved for

$$\lambda_{iq}^\delta = (\kappa_1 \hat{X}^{m\prime} \hat{X}^m + D_0(\xi_\lambda))^{-1} \kappa_1 \hat{X}^{m\prime} (h_{iq} - \delta^x),$$

[30] The model can also be estimated using the ADAM algorithm using, for instance, JAX.

[31] The model can be extended by also regularizing the bias parameters, $(\delta_{iq}, \delta_a, \delta_t)$.

where omitted subscripts indicate column vectors (e.g., $h_{iq}, \delta^x \in \mathbb{R}^{A_q}$). $\hat{X}^m$ is a matrix where the rows correspond to $m^h_{iaq}\hat{x}'_a$.

2. $\hat{\beta}_t$: This is solved for
$$\hat{\beta}_t = (\hat{X}^{m'}\hat{X}^m + D_0(\xi_y))^{-1}\hat{X}^{m'}y_t.$$

3. x^δ_a: The FOC is given by
$$-\kappa_1 \sum_{i,a,q} m^h_{iaq}(h_{iaq} - \delta^\lambda_{iq} - x^{\delta'}_a\hat{\lambda}_{iq})\hat{\lambda}_{iq} - \kappa_2 \sum_{t,a} m^y_{at}(y_{at} - \delta^y_t - \beta'_{0t}x^\delta_a)\beta_{0t} + D_0(\xi_x)x^\delta_a = 0,$$
which is solved for
$$x^\delta_a = (\kappa_1\hat{\Lambda}^{m'}\hat{\Lambda}^m + \kappa_2 B_0^{m'}B_0^m + D_0(\xi_x))^{-1}(\kappa_1\hat{\Lambda}^{m'}(h_{aq} - \delta^\lambda) + \kappa_2 B_0^{m'}(y_{at} - \delta^y_t)),$$
where the rows of $\hat{\Lambda}^m$ are $m^h_{iaq}\hat{\lambda}'_{iq}$ and the rows of B_0^m are β'_{0t}.

4. (Optional) Normalization: As the embeddings are only identified up to scaling and rotation, we normalize $X = (x_{ak})_{a=1\ldots A}$ to
$$\frac{1}{A}X'X = I_K. \tag{30}$$
so that for each component k, the average value of x^2_{ak} is 1, averaging across assets. In this way, the scaling is robust to the number of assets in a given period. To do so, using the candidate estimates X^c, we decompose $X^{c'}X^c = R'R$ using the Cholesky decomposition and construct $X = A^{\frac{1}{2}}X^cR^{-1}$.

Initialization To initialize the algorithm, we compute the mean of h_{iaq} across assets for an investor to initialize δ_{iq}. After removing these means, we compute the mean across i and q of $h_{iaq} - \delta_{iq}$ to initialize δ_a. We then estimate We set the missing values in the matrix of $h_{iaq} - \delta_{iq} - \delta_a$ to zero and extract principal components to initialize x_a. We then start from step 1 in the algorithm above.

Out-of-sample evaluation When estimating supervised embeddings, we are often interested in how well the embeddings explain y_{at} out of sample. For this reason, we allow for different mask weights, m^h_{iaq} and m^y_{at}. This allows us to use all of the holdings data but only, for instance, 80% of the supervised data to extract the embeddings.

D Data construction

D.1 CRSP, Compustat, and FactSet holdings data

We combine data on equity prices from CRSP, accounting data from Compustat, and holdings data from FactSet.

FactSet holdings data: 13F We build an end of quarter panel of institutional equity holdings using FactSet 13F Ownership data. FactSet provides security-level holdings data for each 13F filer, which we aggregate to the rollup entity level by summing the reported holdings. We measure reported holdings using the dollar value of holdings (`adj_mv`).

We compute security-level market equity in the FactSet data as the product of unadjusted prices and unadjusted shares outstanding. We merge end-of-quarter market equity data onto the 13F holdings data.We require that observations have positive security-level market equity and a positive value of holdings.

From the 13F data, we select rollup entities that are identified by FactSet as hedge funds using the variable `entity_sub_type` in `own_ent_institutions`.

FactSet holdings data: Funds We build an end of quarter panel of security-level fund holdings using FactSet Fund Ownership data. We measure holdings using the reported dollar value of holdings (`adj_mv`). Funds often report at a frequency higher than quarterly, so we sort the fund filings in increasing order by `report_date, filing_date, transfer_date,` and `form_type` and select the last report_date by fund. We require the `report_date` to be within 5 days of the last day of each quarter.

We select the subset of funds that have `fund_type` being one of mutual funds, ETFs, closed-end funds, or variable annuity funds. Our sample of holdings data therefore consists of holdings from the FactSet funds data combined with holdings of hedge funds from the FactSet 13F data.

CRSP and Compustat We link FactSet security-level holdings data to CRSP using historical CUSIPs. To do so, we first merge historical end-of-quarter CUSIPs onto the FactSet security-level holdings panel using `sym_cusip_hist`. We then merge on CRSP and Compustat data using historical CUSIPs which are available in CRSP. We then merge on characteristics and returns from Jensen et al. (2023) (JKP) by merging on permno and date.

Sample construction We aggregate characteristics and returns to the permco level by weighting by the permno level market equity from CRSP. We require at least one of the permno level market-equity variables not be nano or micro cap as indicated by the JKP data. We require a positive value for permco level book equity.

We use six core observed characteristics, potentially alongside log market equity and log book equity, namely the log book-to-market ratio (be_me), gross profits-assets (gp_at), asset growth (at_gr1), CAPM beta (beta_60m), momentum (ret_12_1), and dividend-assets (div_at).We drop any observation with missing core characteristics and winsorize all variables by quarter at the 1%- and 99%-percentiles, except log market equity, log book equity, and dividend-assets. We winsorize dividend-assets at the 99% level.

Due to limited coverage in the funds data at the beginning of the sample, we start our sample from from 2005 Q1 and end the sample in 2022 Q4. We drop investors with highly concentrated of positions, which we define as a single holding being greater than 75% of their portfolio. We also require each manager hold a minimum of 20 stocks and each stock be held by a minimum of 20 managers, which we iteratively enforce until both conditions are met within each quarter.

We construct the log dollar value of holdings and winsorize this variable as follows. Within each quarter, we first remove investor fixed effects using the median holding by investor. We then remove asset fixed effects by removing the median across assets. We then winsorize this centered value at the 2.5% and 97.5% level. Next, we remove investor and asset fixed effects using means. The total investor and asset fixed effects are the sum of the median and mean values that were removed for investors and assets, respectively.

D.2 Text-based asset embeddings

As a point of reference, we use text-based asset embeddings from Cohere and OpenAI that we access via their APIs. We use their latest models as of March 2024. From Cohere, we use the `embed-english-v3.0` model that provides embeddings of dimension 1,024. Depending on the downstream task, Cohere provides different text embeddings (to store documents in a vector database, to structure search queries, for classification tasks, and for text clustering). We use the embeddings for clustering tasks, as our goal is to identify similar stocks. To reduce the dimension of the embeddings, we use the UMAP method (McInnes et al., 2018).

Analogously, from OpenAI, we use their `text-embedding-3-large` model. In its most general form, OpenAI's model provides embeddings of dimension 3,072. However, we can obtain lower-dimensional embeddings from OpenAI and download embeddings of different dimensions for comparability.

As the text-based embeddings do not have a temporal dimension, we download the embeddings for the last quarter of our sample period (2022.Q4). We use the company names from CRSP to retrieve the embeddings. Both Cohere and OpenAI recommend using cosine-similarity distance to measure stocks' similarities.