MANIFOLD LEARNING FOR EMPIRICAL ASSET PRICING

by

Michael Ayres Baeder

A thesis submitted to Johns Hopkins University in conformity with the requirements for the degree of Master of Science

Baltimore, Maryland

May 2022

Abstract

This thesis develops a methodology for applying modern manifold embedding algorithms to the problem of empirical asset pricing. Our technique combines traditional linear compression with geometric dimensionality reduction in order to characterize the time-evolving distribution of a nonconstant dimensional time series using a small number of latent factors. We use this model to perform an asset pricing study on US equity data from 1980 to present, using a novel cross-validation approach suited to the problem. These preliminary results are competitive with, but do not significantly outperform, simpler models which are restricted to linear structure. We propose several extensions of the model and calibration techniques which may improve performance.

Primary reader and advisor: Burhan Sadiq

Secondary reader: Thomas Woolf

Acknowledgments

I would like to thank my advisor, Burhan Sadiq, for his guidance and mentorship throughout the research process. I am grateful for his help thinking through many challenges that have arisen during this work (and his patience entertaining many half-baked ideas). I would also like to thank my second reader, Thomas Woolf, for his support and feedback. I owe him a special thanks for introducing me to the fascinating topic of manifold learning, and for the reference which inspired this research.

I would also like to express my gratitude to my employer, Campbell & Company, for its generous support of my research. I am very thankful for the kind words, encouragement, and helpful feedback I've received throughout this process from my colleagues in the research department.

Finally, I would like to thank my friends and family for their support (and no small amount of patience) throughout this period.

Contents

List of Tables

List of Figures

1 Introduction

Advances in computing power, data availability, and statistical learning techniques have broadened the set of problems which can be addressed through direct, data-driven techniques. This work leverages recent developments in the field of manifold learning to address a central question in modern empirical finance: what common factors drive the risks and expected returns of firms?

Financial data has a complex covariance structure, but it is broadly believed that this arises from the interaction of a low-dimensional set of common risk factors and high-dimensional noise. Economic theory has argued rigorously for the existence of a latent risk factor structure which linearly spans the space of expected returns, but is not able to directly specify what these factors are. Traditional approaches in the literature choose a set of factors motivated by economic intuition, trading off misspecification risk for parsimony and interpretability. A more recent vein of research assimilates a large number of candidate factors using modern machine learning techniques to identify a compact set of risk factors in a data-driven fashion. Our work continues in this direction, identifying the similarity between the arbitrage pricing theory of Ross (1976) and the *manifold hypothesis* of machine learning in order to leverage the flexibility of nonlinear models while relying on parsimonious assumptions motivated by economic theory. The method that we propose is adapted from the approach of Lian et al. (2015), who apply manifold embedding techniques to time series data. We add a pre-processing step which allows our model to accomodate a dataset with nonconstant width. In addition to using the diffusion map algorithm investigated in that work to perform the embedding, we also consider the recently-developed UMAP algorithm of McInnes et al. (2020). To our knowledge, this work is also the first to apply manifold embedding techniques to the field of asset pricing.

We use this enhanced time series analysis methodology to perform an empirical study of

the cross-section of US equity returns. Our approach follows most closely the framework of Kelly et al. (2019), particularly the extension which uses firm-level characteristics in addition to returns. Relative to that work, our technique relaxes the restriction that latent factors lie on a hyperplane spanned by characteristics to the more general assumption that they lie on a low-dimensional manifold embedded in the ambient space. Our approach to estimating pricing errors via cross-sectional cross-validation, rather than splitting the data into training and validation sets temporally, is also distinct to this work. Our preliminary results do not significantly outperform simpler linear benchmarks; however, we identify ways to refine the methodology which may lead to improved performance.

The remainder of this paper lays out each of these topics in detail, according to the following structure. Section 2 provides an overview of the literature in manifold learning and empirical asset pricing, to contextualize this work and its contributions. Section 3 discusses our methodology in detail, including an extension of the model utilizing characteristic information, and benchmarking our approach against a more constrained linear model. We also establish our cross-validation technique for producing error estimates, which is crucial to the study and distinct from many others in the literature. In section 4, we perform an empirical study of US equity returns using our method and discuss the results. Section 5 concludes with a discussion of future avenues of research.

2 Literature survey

2.1 Manifold learning

This work utilizes two manifold embedding algorithms developed in the mathematics literature. They are both neighbor graph type methods, which attempt to integrate information about local similarity (the K nearest neighbor graph) into a global metric.

The first algorithm we consider is diffusion map, developed in Coifman and Lafon (2006). It begins by fitting a diffusion process to the data. The diffusion distance – the probability-weighted average length of all paths between two points along the embedded manifold – is then given by examining the eigenfunctions of the diffusion operator. This weighted-average definition of embedding distance is more robust to sampling noise than the sample geodesic distance, and can be computed fairly quickly even on high-dimensional data. By varying the time-scale and normalization of the diffusion kernel, structures at varying levels of localization can be detected. Other algorithms in this family include the Laplacian eigenmaps of Belkin and Niyogi (2003).

The second algorithm we consider is the Uniform Manifold Approximation and Projection (UMAP) algorithm of McInnes et al. (2020). While diffusion maps have their origin in stochastic differential equations, UMAP is motivated by algebraic topology. It constructs a local metric around each observed data point, forming a fuzzy simplicial set, and approximates the embedded manifold as the union of these. Mapping points on the fitted manifold to a lower-dimensional space is then a tractable optimization problem. Normalizing distance metrics allow the technique to scale well to high-dimensional datasets and to produce high-quality embeddings regardless of the target dimension.

Applying these techniques to time-series data presents a unique challenge, since the dis-

tribution from which the data is sampled is not constant across observations. This issue is raised in Coifman and Hirn (2014), which addresses the generic problem of manifold learning on changing data. The technique of Lian et al. (2015) builds on this and forms the foundation of the approach in this thesis. Their technique is motivated by the information-geometric dimensionality reduction of Carter et al. (2011), which focuses on the advantages of performing manifold learning in the space of distributions. Rather than treating the data as a point-cloud in Eucliean space and applying dimensionality reduction to the observations directly, Lian et al. (2015) apply manifold embedding in the space of distributions, using the symmetrized Kullback-Leibler divergence as their distance metric. In this way, the temporal structure of the data is incorporated through the point-in-time distribution and the problem is put on similar footing to the standard setting.

2.2 Empirical asset pricing

The technique proposed in this work was directly motivated by its potential applications to asset pricing in financial economics. This field has an extensive literature going back at least to Markowitz (1952), which established optimal portfolio theory and the central role of estimating risk and expected returns.

2.2.1 Supervised latent factor models

While the space of financial asset returns is extremely high-dimensional, researchers have long suspected that there are a relatively small number of latent factors which drive return co-variation. The capital asset pricing model (CAPM) of Sharpe (1964), among other contemporaneous work, was the first set of financial theories which attempted to establish a low-dimensional set of risks which drive common variation in asset returns. In CAPM, the covariation between a security's return and a broad-market index (called β, since it is estimated as the coefficient of a linear regression of each security's returns onto those of the market index) is assumed sufficient to describe all common variation between firms. This

forms an asset pricing theory because all remaining risk, which is idiosyncratic and uncorrelated among distinct stocks, will be diversified away in a large portfolio. The equilibrium theory thus argues that the expected return of each stock should be proportional to its β.

The arbitrage pricing theory (APT) of Ross (1976) generalized this line of reasoning to higher dimensions and argued for the existence of a latent factor structure. Ross' argument follows in a straightforward fashion from the intuition of CAPM. Risks that are common across firms cannot be diversified away by holding a large portfolio; therefore, investors must demand compensation (a risk premium) for bearing them. Risks which are firm-specific (idiosyncratic) *can* be mitigated via diversification and so should not be compensated. If expected returns are not proportional to common risk exposures, arbitrageurs could form riskless portfolios with positive returns. Therefore, in equilibrium we would expect competitive pressures to give rise to a factor structure, wherein expected returns are linear in common risk exposure.

The APT lays a firm theoretical foundation for the existence of a factor structure, but does not specify what exactly these common risk factors are or how they might be identified, giving rise to a large literature. The dominant approach in the literature has used factors which are chosen according to a mixture of economic intuition and strong empirical performance. Among the most influential have been the 3-factor model of Fama and French (1993), the 4-factor model of Carhart (1997), and the 5-factor model of Fama and French (2015). Taking Fama and French (1993) as an example, the factors chosen are the market index (as in CAPM), the returns for stocks with small capitalizations versus those with large capitalizations ("small-minus-big" or SMB) and stocks with high book values versus those with low book values relative to their market capitalization ("high-minus-low" or HML). These models have the benefit of being economically intuitive, but are also potentially prone to misspecifcation or selection bias effects.

2.2.2 Machine learning and semi-supervised asset pricing models

This work is firmly situated in an emerging trend which seeks to leverage machine learning techniques in asset pricing; see Gu et al. (2020) for an extensive survey. Chamberlain and Rothschild (1983) were among the first to recognize the potential for data-driven statistical techniques to identify latent factor structure. They establish the asymptotic sufficiency of principal components analysis, applied to the covariance matrix of asset returns, to uncover the latent risk factors when the market is infinitely large and the structure constant over time. In practice however, the restrictions of PCA limited its sucess in empirical applications. More recent work has been able to overcome these limitations and provide some synthesis of the data-driven approach with conventional factor models. The instrumental principal components analysis (IPCA) of Kelly et al. (2017) leverages a large set of characteristics, such as the SMB and HML factors in Fama and French (1993), as *instruments* rather than factors in and of themselves. By assuming that these characteristics are linear proxies of the latent risk factors, this framework can accommodate a large number of potential factors and allows the loadings of each individual security to vary with its characteristics, simultaneously mitigating the issues of model misspecification in traditional factor models and the problem of time-varying risk exposures in PCA. Kelly et al. (2019), which applies IPCA to an empirical study of US equity returns, is most similar in spirit to this work. Our approach relaxes the assumption that the latent factor structure is spanned *linearly* by the instruments, while preserving the geometric intuition that there is a low-dimensional structure (an embedded manifold) which describes risk and expected returns. A complementary extension is Gu et al. (2021), which fits a nonlinear mapping between observable characteristics and market returns using an autoencoder neural network architecture.

3 Methodology

We establish some notation for clarity. For a discretely sampled time series $\{v(t)\}$, we denote by $v(t)$ the vector of observations at time t. We use the notation $v(t_1, t_2)$ to refer to the matrix with first row $v(t_1)$, last row $v(t_2)$, and intermediate rows given by the samples in-between these observations. Subscripts indicate indexing into a given entry such that $v_i(t)$ is a scalar and $v_i(t_1, t_2)$ is a vector.

We let $r(t)$ denote the cross-section of returns. $X^L(t)$ refers to linear and $X^N(t)$ to nonlinear factors used in our pricing model.

3.1 Managing nonconstant width time series

In order to apply the methodology of Lian et al. (2015) we must have a time series of constant dimension, because the distance metric derived from the Kullback-Leibler divergence is not well-defined when the two distributions being compared have different sample spaces. This assumption is challenged in the domain of asset pricing because $\{r(t)\}$ has dimension $N^r(t)$ which varies strongly over time as new firms enter the market or existing firms exit through acquisition, bankruptcy, or delisting. To accomodate this, we pre-process the data by applying a principal components analysis to the estimated rolling covariance matrix of returns. Letting LB_L denote the number of observations used in our rolling window estimation, we begin by demeaning each market return on a rolling basis:

$$r'(t) = r(t) - \frac{1}{LB_L} \sum_{i=0}^{LB_L} r(t-i)$$

and then estimating the rolling covariance matrix as:

$$\hat{\Sigma}^r(t) = \frac{1}{LB_L - 1} r'(t - LB_L, t) r'(t - LB_L, t)^T.$$

We then perform an eigendecomposition of this matrix:

$$\hat{\Sigma}^r(t) = V(t)D(t)V^{-1}(t)$$

with $D(t)$ diagonal. We form a compressed representation of this matrix by taking the first N^l columns of $V(t)$, which are the eigenvectors associated with the N^l largest eigenvalues. Call this truncated $N^r(t) \times N^l$ matrix $L(t)$, and define the linear compression factors

$$X^L(t) = r(t)L(t)$$

This yields a vector of returns which has constant dimension N^l at each point in time. Each $X_i^L(t)$ return can be interpreted as the return to a portfolio whose weights are given by the i^{th} column of $L(t)$. In this way, we compress the time-varying cross-section of returns $r(t)$ into a constant-dimensional cross-section $X_i^L(t)$ of portfolios whose returns are independent and span the covariance structure of the raw cross-section as completely as possible. By choosing large N^l we address the problem of nonconstant dimensionality without prematurely constraining the downstream dimensionality reduction. In addition, by implicitly truncating the covariance structure by removing components associated with small eigenvalues, we reduce the impact of noise on the subsequent embedding estimation.

3.2 Uncovering nonlinear structure with manifold embedding

Having achieved a constant but still very high-dimensional compression of the data through the previous step, we can directly apply the technique of Lian et al. (2015). We model the time-varying distribution of the linear factor data $X^L(t)$ as approximately normal, using the rolling mean and covariance matrix:

$$\mathcal{L}(t) \sim \mathcal{N}(\hat{\mu}^L(t), \hat{\Sigma}^L(t))$$

where $\hat{\mu}(t)$ and $\hat{\Sigma}(t)$ are the rolling sample mean and covariance of the linear factors, respectively, computed in the same fashion as in section 3.1. It is this time series of distributions to which we apply dimensionality reduction. The distance metric which is used to describe local similarity is defined using Kullback-Leibler (K-L) divergence, which under the normal assumption is given by:

$$\mathcal{D}_{K-L}(\mathcal{L}(t_1), \mathcal{L}(t_2)) = \text{tr}(\hat{\Sigma}^L(t_2)^{-1}\hat{\Sigma}^L(t_1)) - N_l + \log \frac{\det\left(\hat{\Sigma}^L(t_2)\right)}{\det\left(\hat{\Sigma}^L(t_1)\right)} + (\hat{\mu}^L(t_1) - \hat{\mu}^L(t_2))^T \hat{\Sigma}^L(t_2)^{-1}(\hat{\mu}^L(t_1) - \hat{\mu}^L(t_2)).$$

Following Lian et al. (2015), the metric which is used for measuring similarity symmetrizes the K-L divergence:

$$d(t_1, t_2) = \frac{\mathcal{D}_{K-L}(\mathcal{L}(t_1), \mathcal{L}(t_2)) + \mathcal{D}_{K-L}(\mathcal{L}(t_2), \mathcal{L}(t_1))}{2}$$

This allows us to define a distance matrix D with $D_{i,j} = d(t_i, t_j)$ to which we apply the diffusion map and UMAP algorithms, which recover a low-dimensional embedding:

$$(X^L, D) \xrightarrow{\text{UMAP/Diffusion Map}} X^N$$

where we drop the time index to indicate that this process is performed once across all time-steps. The reduced time series $\{X^N(t)\}$ has constant dimension N^n with $N^n << N^l$ such that we compress the linear factors significantly. These nonlinear factors form the basis for our asset pricing model. The implementations used for this research expose two key parameters for both algorithms used in constructing a kernel matrix: a bandwidth parameter ϵ and a nearest-neighbor parameter N^{knn}. These are critical parameters as they tune the scale at which the embedding algorithms attempt to identify structure. Since we do not have strong theoretical guidance for choosing these parameters, we will fix the other aspects of

the model (that is, LB_L, N^l and so on) and consider a range of choices for these parameters in our cross-validation. The values of ϵ and N^{knn} will be parameterized as percentiles of the off-diagonal values of D and as a percentage of the number of observations, respectively, in order to yield an appropriate parameter scale for each run.

3.3 Asset pricing

Section 3.2 uncovers a low-dimensional embedding $\{X^N(t)\}$ which compactly characterizes the variation in the data. Unlike the linear factors $\{X^L(t)\}$, these factors are nonlinear and do not correspond to investible portfolios. In order to relate these to the risks and returns of financial assets, we follow the pricing framework of Fama and MacBeth (1973), which is composed of two steps. The first step measures the risk exposures of assets in the cross-section to the latent factors. In order to measure these, we perform a simple rolling regression per asset:

$$r_i(t - LB_\beta, t) \approx \hat{\alpha}_i(t) + \hat{\beta}_i(t) X^N(t - LB_\beta, t)$$

With $\hat{\alpha}_i(t)$ a market-specific constant. This yields a set of exposure estimates $\{\hat{\beta}(t)\}$ for each asset to each factor at each point in time. Note that each $\hat{\beta}(t)$ is an $N^r(t) \times N^l$ matrix. These risk exposures are then used in the second step to estimate the compensation for risk exposure – for each time in our sample, we perform a *cross-sectional* regression across all assets:

$$r(t) \approx \hat{\beta}(t-1)\hat{f}(t)$$

where $\hat{\beta}(t-1)$ includes a column of ones appended to reflect a cross-sectional constant term in the regression. The estimate $\hat{f}(t)$ recovered from this cross-sectional regression is our primary object of interest, because it describes the compensation for taking incremental

exposure to each of the risk factors. We can use it to form an explanatory estimate

$$\hat{r}_i^{Exp}(t) = \hat{\beta}(t-1)\hat{f}(t)$$

which corresponds to the forecast from the cross-sectional regression model. As the name implies, comparing $\hat{r}_i^{Exp}(t)$ to $r_i(t)$ gives us some sense of the power of the model to explain the realized cross-section of returns. In addition, we can define a factor risk premium $\hat{\lambda}(t)$ as the rolling full-sample average of the factor return:

$$\hat{\lambda}(t) = \frac{1}{t-1}\sum_{i=1}^{t}\hat{f}(t-i)$$

to define a predictive estimate:

$$\hat{r}_i^{Pred}(t) = \hat{\beta}(t-1)\hat{\lambda}(t)$$

which is a prediction of the return using only ex-ante information. Clearly, this imposes a much higher hurdle than explanatory estimates, and instead describes the *persistence* of factor returns. A strong correlation between $r_i(t)$ and $\hat{r}_i^{Pred}(t)$ would indicate not only that the risk factors can describe returns, but that the compensation provided for bearing risk is stable over time. We will elaborate on this distinction in greater detail when discussing our performance metrics.

3.4 Cross-validation

Manifold embedding algorithms have multiple tuning parameters which must be chosen – mostly crucially the kernel bandwidth and number of nearest neighors used in forming the similarity matrix. To evaluate our model, it is critical that we have out-of-sample pricing error estimates for each data point. The approach in much of the literature is to report in-sample errors, which are biased because they informed the selection of risk factors and

estimation of factor premia. More recent research motivated by the statistics and machine learning literature employs a temporal train/test split, where the model is tuned using data from one period, then evaluated in a subsequent testing period. While there are techniques for performing such a split using manifold embedding techniques, they are complicated significantly by the fact that these models are nonparametric. In particular, our use of a non-standard distance metric (relative to the library implementations of diffusion maps and UMAP) makes such an effort much more complex. Instead, we obtain unbiased error estimates by splitting the data cross-sectionally.

Our approach is as follows. At each point in time, we randomly assign each market to a unique cross-validation fold. The steps outlined in sections 3.1 and 3.2 are performed on all markets *except* those in our fold of interest, yielding a set of risk factors defined on totally distinct data. These factors are then used with markets in our fold as in section 3.3 to produce pricing errors. In this way, the information in the cross-section used to identify risk factors is completely separated from the set of test assets used to assess those risk factors. The existence of common information across distinct sets of markets is, of course, what we hope to discover, and so should not pose an undue hurdle for a genuinely successful model. By repeating this process across all folds we obtain out-of-sample error estimates for each market in the cross-section.

The full process is summarized in Figure 1. We continue with a discussion of an extension to the model, benchmarking, and performance assessment.

3.5 Incorporating firm characteristics

The methodology as specified in the previous sections uses only the returns of the assets in the cross-section. As an extension, we may also consider exogenous information in the form of *characteristic portfolios*. Similar to the principal components analysis compression, char-

acteristic portfolios are linear combinations of assets. Compared to that process, however, the weights in each portfolio are determined by information from outside the model. For example, in the style of the HML factor of Fama and French (1993), a characteristic could be the book-to-market ratio of each firm in the cross-section, at each point in time.

Let $\{c(t)\}$ denote the time series of characteristics, such that $c(t)$ is a $N^r(t) \times N^C$ matrix, where N^C is the number of characteristics. Continuing with our example, $c_i^{HML}(t)$ would refer to the book-to-market ratio of firm i at time t. We define characteristic portfolios according to the weighting scheme:

$$w_i^j(t) = \text{percentile}(c_i^j) - 0.5$$

where the percentile is defined cross-sectionally across all markets for a given point in time t and characteristic j. Markets whose value exceeds the median will be assigned a positive weight, and those below the median will be assigned a negative weight (corresponding to a short sale). We then compute the characteristic portfolio return simply as:

$$X_j^L(t) = w^j(t-1)r(t)$$

where the lag in w indicates that characteristic portfolios should be described using only ex-ante information. Defining the return in this way allows the market composition to vary with chacteristic loadings. Concluding with our example of book-to-price as a characteristic, $\{X_L^{HML}(t)\}$ gives us a time series of returns which corresponds to being long the stocks with the highest price-to-book ratios versus short the others. As each individual firm's ratio changes, so too will its weight in the portfolio. By repeating this across all characteristics we obtain an N^C-dimensional time series $\{X_L(t)\}$ of characteristic portfolio returns.

As our notation suggests, this approach achieves a similar goal as the dimensionality reduc-

tion described in section 3.1. In the characterstic-based extension, we use these portfolios as the constant-dimensional representation of the data in lieu of the PCA compression. Using a large number of characteristic portfolios achieves the same goal of constant dimensionality while still affording the manifold embedding step significant flexibility. This seamlessly integrates the exogenous information in the characteristics without requiring any alteration to the downstream methodology.

3.6 Benchmark models

Both the original model specification and the characteristic extension use manifold embedding to identify a small set of risk factors which form the basis of the pricing model. In order to benchmark our approach, we also consider simpler models which are restricted to linear formulations. As an extreme example, we consider a full linear model which uses the set of linear factors obtained from step (2) of Figure 1 directly, without any subsequent compression. This serves as an upper bound on the potential of the pricing model and a measure of the information that can be extracted from the linear factors. We also fit a restricted linear model which compresses the data linearly, again using principal components analysis, to extract the same number of factors as used in the manifold models. The restricted model provides a more direct comparison of the value added from the nonlinear manifold embedding step.

3.7 Performance assessment

Following Kelly et al. (2019), we assess the performance of our models primarily on the basis of their *total* and *predictive* R^2 values. They are both simply defined using the explanatory and predictive estimates $\hat{r}_i^{Exp}(t)$ and $\hat{r}_i^{Pred}(t)$ defined in section 3.3. First, we have:

$$R^2_{Total} = 1 - \frac{SSE_{Explanatory}}{SSE_{Total}} = 1 - \frac{\sum_{i,t}(\hat{r}_i^{Exp}(t) - r_i(t))^2}{\sum_{i,t} r_i(t)^2},$$

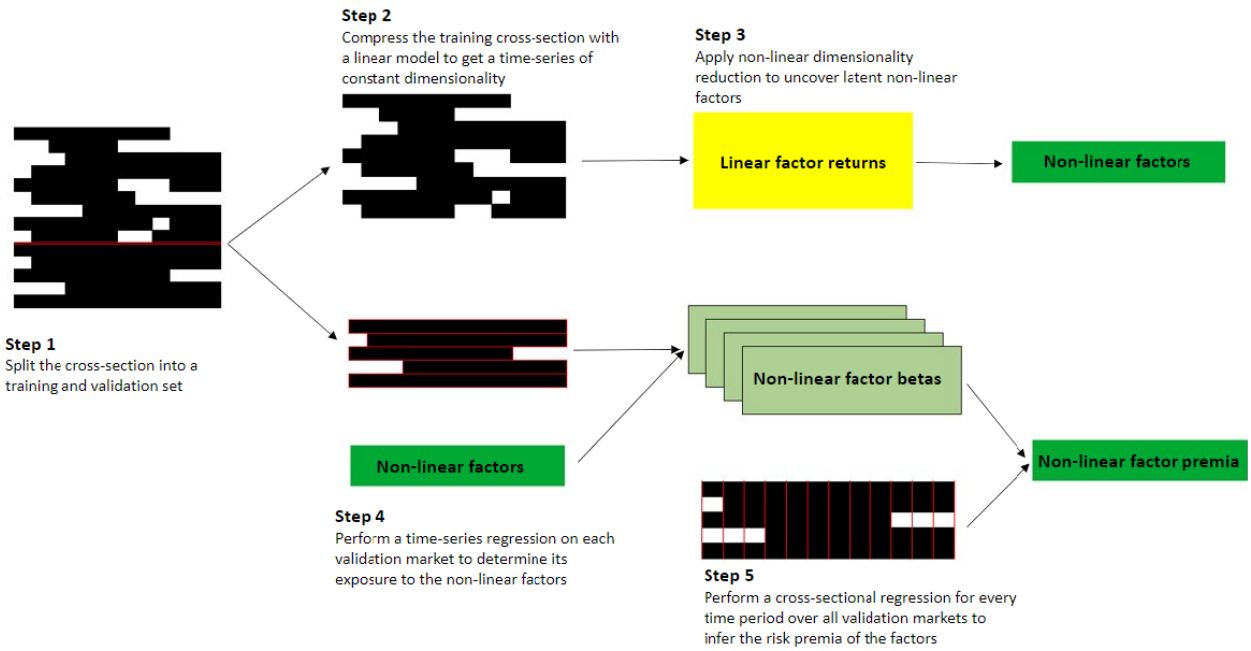

Figure 1: A flow diagram which summarizes the approach to identifying latent factors and producing out-of-sample asset pricing errors using cross-validation.

where $\hat{r}_i^{Exp}(t) = \hat{\beta}(t-1)\hat{f}(t)$ estimates the return using the contemporaneous factor return estimates. An R^2_{Total} value of 1 would indicate the model is able to explain, contemporaneously, all of the variation in the cross-section. Importantly, this does not mean that returns are predictable, since the factor returns themselves $\hat{f}(t)$ could have strong explanatory power but be impossible to forecast ex-ante. The metric R^2_{Pred} addresses the notion of predictability, and is defined as

$$R^2_{Pred} = 1 - \frac{SSE_{Predictive}}{SSE_{Total}} = 1 - \frac{\sum_{i,t}(\hat{r}_i^{Pred}(t) - r_i(t))^2}{\sum_{i,t} r_i(t)^2}$$

where $\hat{r}_i^{Pred}(t) = \hat{\beta}(t-1)\hat{\lambda}(t)$ is a prediction using only information available prior to the period when the return is observed. $\hat{\lambda}(t)$ plays the role of a simple forecast of the next-period factor return: its rolling full-sample average up until that point. A high value of R^2_{Pred} would indicate both that the latent factors can describe significant amounts of return variation, and that the factor returns themselves are highly persistent. It is possible for R^2_{Pred} to be negative, indicating that the variance of prediction errors is larger than the

variance of the cross-section itself, but it is worth noting that recent work by Kelly et al. (2021) argue that the investment implications of $R^2_{Pred} < 0$ are underdetermined because the *investment performance* of such a model when constructing e.g. optimal portfolios will also be influenced by its bias. As they demonstrate, there exist models with negative R^2_{Pred} whose improvement in prediction bias is sufficient to offset their increased variance. While we will not focus on such considerations in this document, we note that this a known limitation of R^2_{Pred} in summarizing the quality of a pricing model.

4 Empirical study

Having established our method for building asset pricing models with manifold embedding, we apply this model in an empirical study of the cross-section of US stock returns. We begin with a discussion of our data sources. We provide additional details on the software used to perform this analysis in Appendix A.

4.1 Data

4.1.1 CRSP

Historical security returns, prices, volumes, and shares outstanding were downloaded from the monthly stock file from the Center for Research in Security Prices (CRSP; see University of Chicago (2020)). We download monthly data beginning in 1965, covering all securities listed on NYSE, NASDAQ, and BATS with prices above $5. The data was accessed using Wharton Research Data Services (WRDS) in February 2022. We use monthly data, rather than daily, because it substantially reduces the computational burden of fitting the model compared to daily data, while still being economically relevant (and perhaps more so, as daily returns, particularly for small stocks, may be prone to predictable but economically insignificant illiquidity effects).

4.1.2 Open Source Asset Pricing

For characteristics, we use the dataset created by the Open Source Asset Pricing project of Chen and Zimmermann (Forthcoming). They aggregate 202 predictors advanced in the literature for the full cross-section of stocks in CRSP. These signals form the basis of our characteristics-based extension. We subset from their dataset the 50 factors with the greatest level of mutual availability with our markets, which have been continuously available since 1965. We construct the returns of the characteristic portfolios independently, in order to accommodate the cross-validation scheme outlined in the previous section.

4.2 Historical market set construction

With the flexibility of our technique comes increased sensitivity to data processing decisions. In order to stabilize our results and mitigate the impact of possibly erroneous datapoints, we construct historical market sets which impose several additional requirements on markets:

1. 80% of observations not missing in the last five years, and none missing in the last year.
2. No months with zero volume.
3. No more than 5% of months in the last five years with zero return.

It is critical that these filters are applied only on the basis of information that would have been available at the time; otherwise, the analysis will be contaminated by survivorship effects. It is for this reason that we use high-quality data providers which build their databases from point-in-time historical data.

Each year, we apply our screen to all of the firms in the dataset. From the firms satisfying this criteria, we take the top 1,000 by market capitalization (if more than that remain) as of the end of the formation period. This market set is then fixed for one year, and we repeat the process the following year. This yields a sequence of market sets, one per year, to which we fit our latent factor model and analyze pricing performance. As indicated in figure 2, in all periods after 1980 we have more than $1,000$ markets fitting our screening criteria.

The intention of this process is that we have a set of fairly "well-behaved" markets which are unlikely to have major data issues, which would require greater expertise in this dataset to troubleshoot and resolve, and which have the potential to distort results. The formation based on only ex-ante properties of the markets ensures that we are not incorporating knowledge of future events which could bias our results. We argue that this process still yields a set of markets which are economically important and interesting. Figure 3 shows

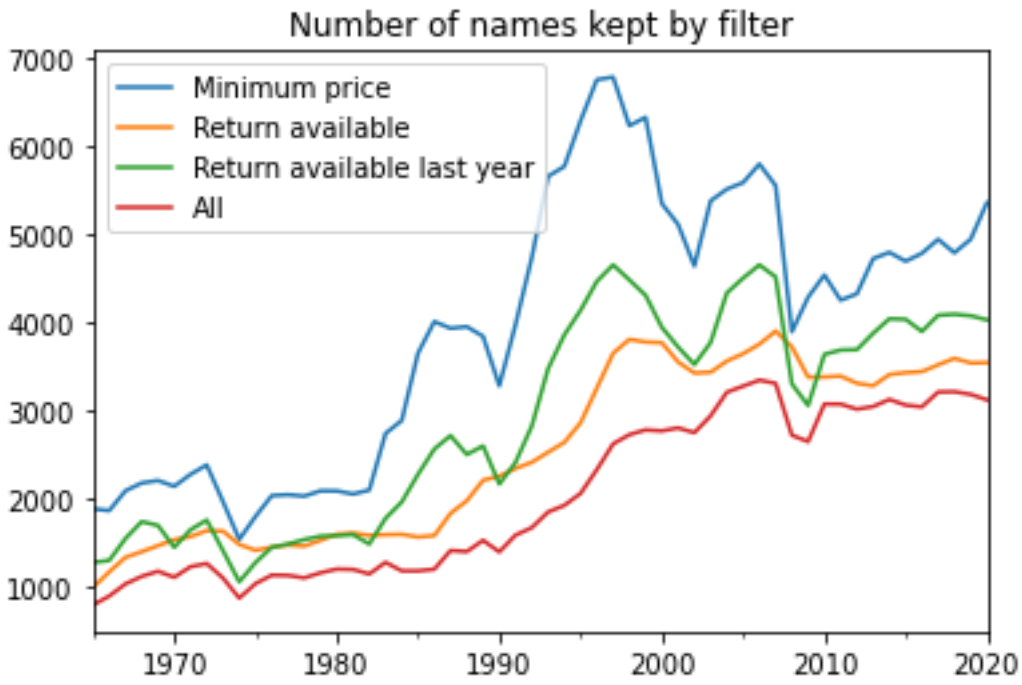

Figure 2: The number of firms satisfying our various screening criteria. We do not plot the zero volume or zero return filters here as the vast majority of firms satisfy these requirements.

that although the number of firms we use is relatively small compared to the entirety of the cross-section, they represent the vast majority of the total market capitalization of the US stock market. The orange line in this figure indicates that even if we were to extend our market set to cover all firms meeting our criteria, the gain in terms of market cap coverage would be small.

It is important to note that this screening process will focus our analysis on stocks with medium to large market capitalizations. This is a double-edged sword; on the one hand, small, illiquid firms may be more prone to outliers and other data issues which could destabilize our pricing estimates. On the other, because of their illiquidity and high trading costs, small firms could exhibit stronger pricing accuracy and more predictable returns. Studies which perform their analysis on subsets of the market often find qualitatively different results for small and large stocks – for example, Kelly et al. (2019) document lower R^2_{Total} but

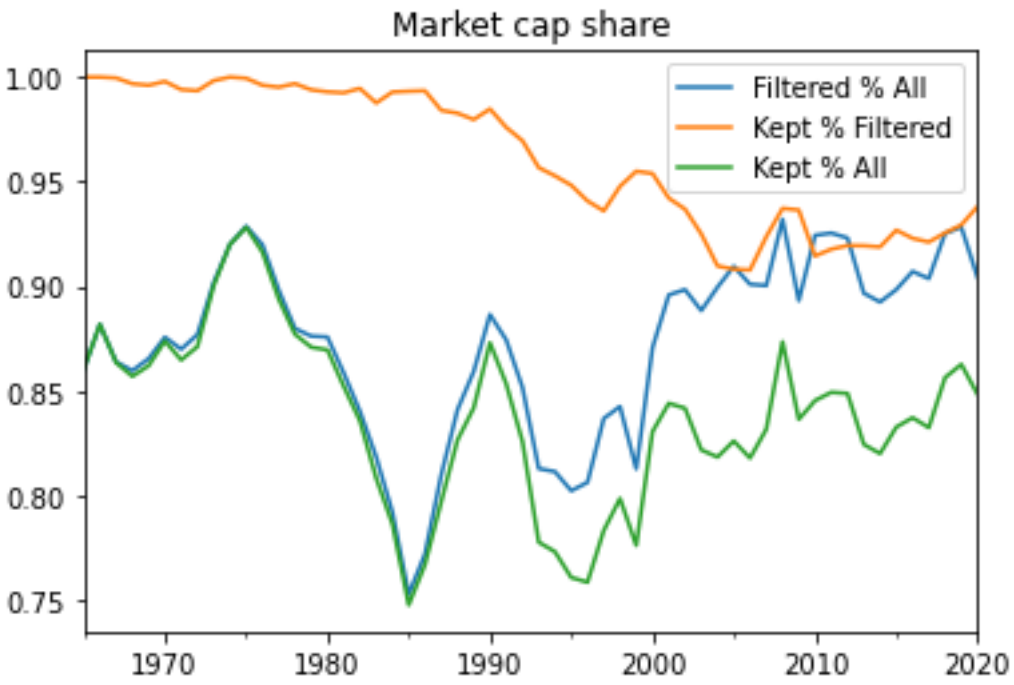

Figure 3: A comparison of the market cap share for firms kept in our historical market set. The blue line indicates the share of market cap which passes all of our filters. The green line indicates the share of market cap in our historical market set. The orange line re-normalizes the market cap share kept by only including eligible firms.

higher R^2_{Pred} when their analysis is restricted to small firms. For this reason, we caution direct comparison of our results with those of other studies which may have included more small stocks, and instead emphasize comparisons to our simple benchmark models which hold the market set constant.

4.3 Setting expectations

For readers not familiar with the empirical finance literature, we wish to emphasize that signal-to-noise ratios in financial data are extremely low. Asset pricing seeks to model the structure of expected returns which reflect the information available to a representative investor. Realized returns will, of course, be influenced by significant events which are not predictable ex-ante and by information which a marginal investor would not have known (or, due to regulatory restrictions, could not have acted on). As a result, even successful

models tend to have very low R^2 values compared to studies in the physical sciences. Particularly in the predictive dimension, we expect competitive pressure to *force* low R^2 values, as strong return predictability creates opportunities for arbitrageurs, whose exploitation of this predictability is self-correcting.

4.4 Parameter settings

There are a number of free parameters in our methodology which must be defined. In order to focus our analysis on the value of manifold embeddings, we fix as many parameters as appropriate, while allowing the two critical parameters which drive the embedding – the kernel bandwidth parameter ϵ and number of nearest neighbors N^{knn} – to vary. For the identification of risk factors, we prefer a fairly long lookback of 60 observations (corresponding to five years) to keep the estimation stable, while for the beta estimation in asset pricing we use a shorter lookback of 36 months (three years) to allow the market-level risk loadings to be more dynamic. We set the number of linear factors used in the pre-processing step to $N^l = 50$; this is ad hoc but of a similar order of magnitude as studies which process a large number of cross-sectional factors such as Kelly et al. (2019). From this, we extract $N^n = 5$ latent factors using the manifold embedding algororithms; this is the same number of risk factors as used in Fama and French (2015) and in the upper range of the number of latent factors used in the studies of Kelly et al. (2019) and Gu et al. (2021).

For the embedding parameters, we specify the N^{knn} parameter as a percentage of available observations, and ϵ as a percentile of (off-diagonal values of) the distance matrix D, which allows the values to be specified on a consistent scale for each iteration. Our choice of parameter values is summarized in Table 1. In our cross-validation, we use 5 folds to which each market is assigned at random, which means that each fold contains approximately 200 markets. This number of folds is computationally feasible, and the 200 market validation set for each iteration ensures there is a substantial number of markets to which we can fit a

Term	Meaning	Value
N^l	Number of factors used in the pre-compression	50
N^n	Number of latent factors in the embedding	5
LB_L	Lookback for the PCA pre-compression*	60 months
LB_N	Lookback for fitting the distribution	60 months
LB_β	Lookback for estimating market-factor betas	36 months**
N^{knn}	KNN parameter in manifold embedding algorithms	20-80% of observations
ϵ	Bandwidth parameter in manifold embedding algorithms	10-50$^{\text{th}}$ percentile of D

Table 1: The parameter values that we use in our empirical study. The embedding algorithm parameters N^{knn} and ϵ are varied in a range of 5 values each. *For the characeristics extension, there is no pre-compression. **The large benchmark model uses 60 months for the beta estimation, since this lookback must be larger than the number of factors (50).

pricing model.

4.5 Results

4.5.1 Latent factors

We begin by considering a few examples of the factors extracted by our method. First, we consider the linear (PCA-based) factors constructed in the pre-processing step. Recall that the eigenvectors extracted from the covariance matrix, which form our PC factors, correspond to linear combinations of markets and so can be interpreted as portfolios. Figure 4 shows an example of the first five factors constructed from a cross-validation fold, plotting the cumulative return of the portfolio corresponding to each factor. The first principal component, by construction, captures the greatest amount of common volatility in the covariance matrix. The next four factors appear to move significantly less, indicating that the volatility absorbed by the first pricipal component is much greater than the others (or equivalently, that the eigenvalue spectrum of the rolling covariance matrix decays rapidly). Readers familiar with market history may recognize that the first factor's cumulative return history looks very similar to the *negated* return of a broad-market index. Note, for example, the peaks up in the late 2000's (corresponding to the global financial crisis) and the blip in

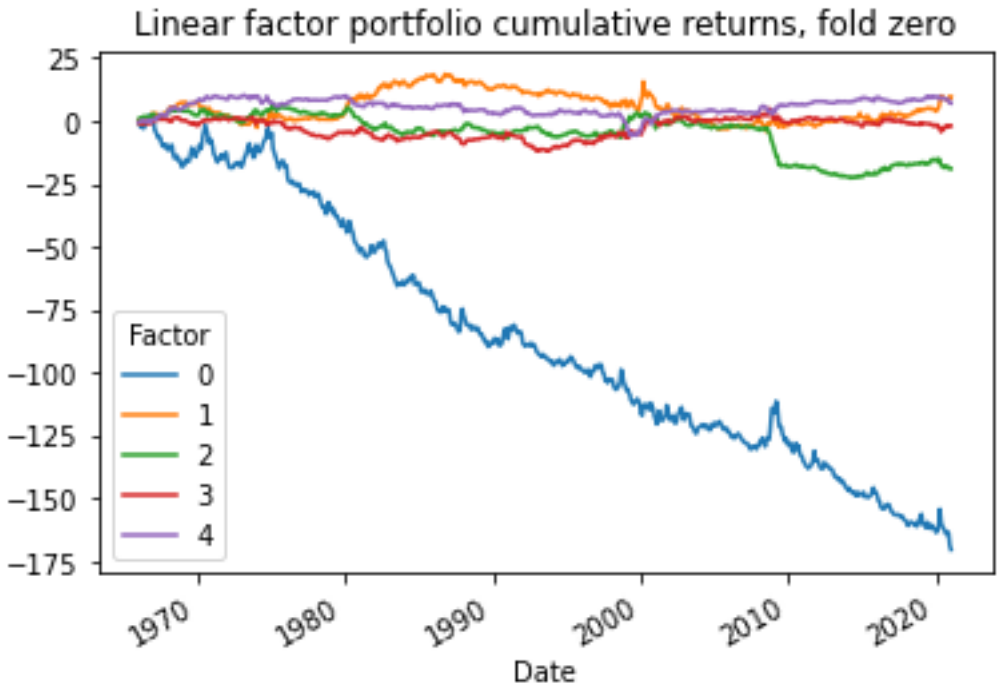

Figure 4: An example of the factors discerned from the PCA pre-transformation, which yields our time-series of constant dimensionality. Here we plot the cumulative sum of the factor values, since these factors correspond to investible portfolios and hence the cumsum is their total return.

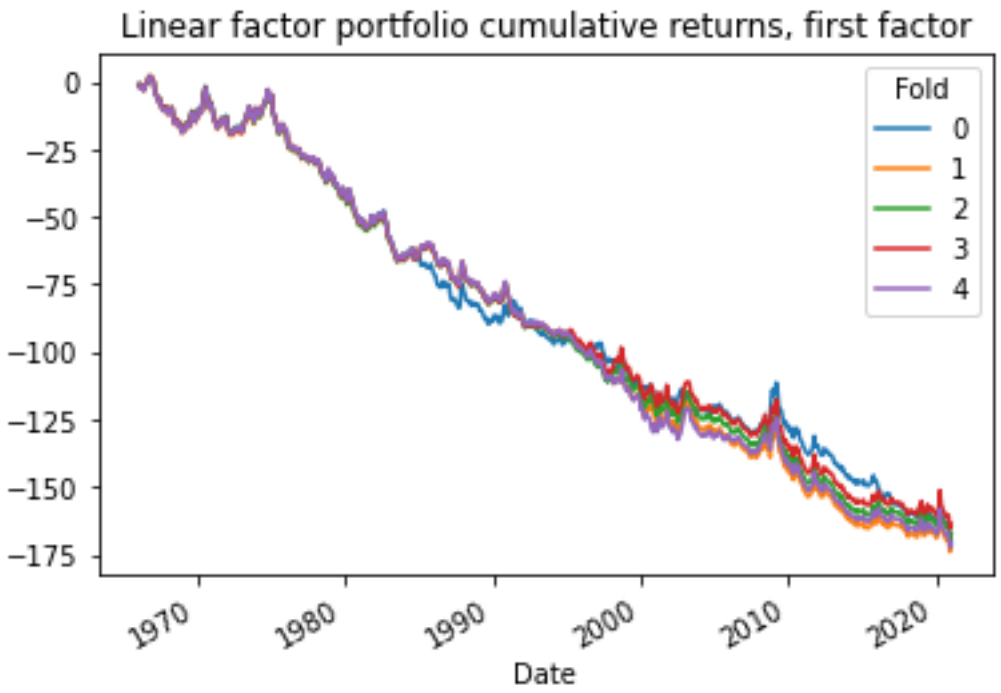

Figure 5: The first PCA-based factor for each fold. It is consistent stylized fact that the first principal component of the market covariance matrix is a broad "market" factor. Note that because of the rotational invariance of PCA, factor is "short" (weights are mostly negative), such that this would be strongly *negatively* correlated to a market index.

early 2020. This is consistently the case; Figure 5 compares the first principal component across each cross-validation fold. In each case, we see a similar pattern of the first principal component capturing a broad "market" factor. The fact that this first factor has negative returns, rather than positive, points to a limitation of principal components analysis that will affect the ability of our methodology to identify structure. The decomposition of the market covariance matrix into principal components is rotationally invariant. If L corresponds to a mapping constructed from PCA and R a rotation matrix satisfying $RR^T = I$, then the covariance matrix re-constructed from the mapping RL is $(RL)^T RL = L^T L$, identical to the original. The fact that this rotation is arbitrary, and may vary over time as we perform the analysis on a rolling basis, introduces unnecessary variability into our factor data which may affect the subsequent dimensionality reduction.

With that caveat aside, we turn our attention to the nonlinear latent factors extracted by our manifold embedding algorithms. Unlike the linear PCA-based factors, the latent factors extracted from these algorithms are a nonlinear (and nonparameteric) function of the data, and so do not correspond to portfolios. As a result, there is no instrinsic scale on which to compare the data; we instead focus on how the embeddings vary over time and across different choices of parameters.

Figure 6 shows an example of the latent factors extracted for the first cross-validation fold. The top figure shows the UMAP embeddings. We can see that although they are locally smooth, they appear to be strongly oscillatory. Compared to the PCA-based factors, there is no strong ordering of the latent factors in terms of their time-series variation. Large jumps in the embeddings may correspond to windowing effects from our rolling covariance matrix. The bottom chart shows the same set of factors extracted by diffusion maps, using the same set of parameters. In contrast to UMAP, the latent factors extracted by diffusion maps do not appear to strongly oscillate. We again see windowing effects such as the late 1980s,

wherein all factors appear to change significantly at the same time.

Figures 7 and 8 let us examine how the embedding changes as a function of the localization parameters ϵ and N^{knn}, focusing on the first latent dimension identified. Beginning with Figure 7 on the left, we can see that most values of N^{knn} behave similarity. There appears to be a discontinuity between using 60% and 80% of observations, where we see qualitatively different behavior. On the right chart of the same figure, we see less variation in the embedding as we vary the kernel bandwidth parameter. For very high values of ϵ, which would indicate a higher threshold for measuring local similarity, we see that the values appear to be significantly more oscillatory. This may indicate that distances between distributions which are farther away are subject to a greater degree of measurement noise, a problem we will investigate further later on. Figure 8 provides a similar comparison for the diffusion map embeddings. Relative to UMAP, the situation is reversed. The embedding seems largely invariant to the choice of N^{knn}, but there is a qualitative shift in behavior as the kernel bandwidth ϵ parameter is increased, which causes the entire time-series of embeddings to flip sign. This may again indicate a kind of rotational invariance in the latent factor extraction, which could pose challenges for fitting an asset pricing model.

4.5.2 Asset pricing performance

We now investigate the performance of asset pricing models which use the latent factors we examined in the previous section. For each market, we follow the cross-validation procedure as discussed prior, so that the latent factors used to explain returns are constructed without reference to the markets on which they are evaluated. This gives us a set of pricing model errors for each choice of the embedding parameters ϵ and N^{knn}, which we now consider in detail.

The explanatory power of the models is illustrated in Figure 9, which shows the performance metric R^2_{Total} for each pair of parameters considered. Although it was difficult to

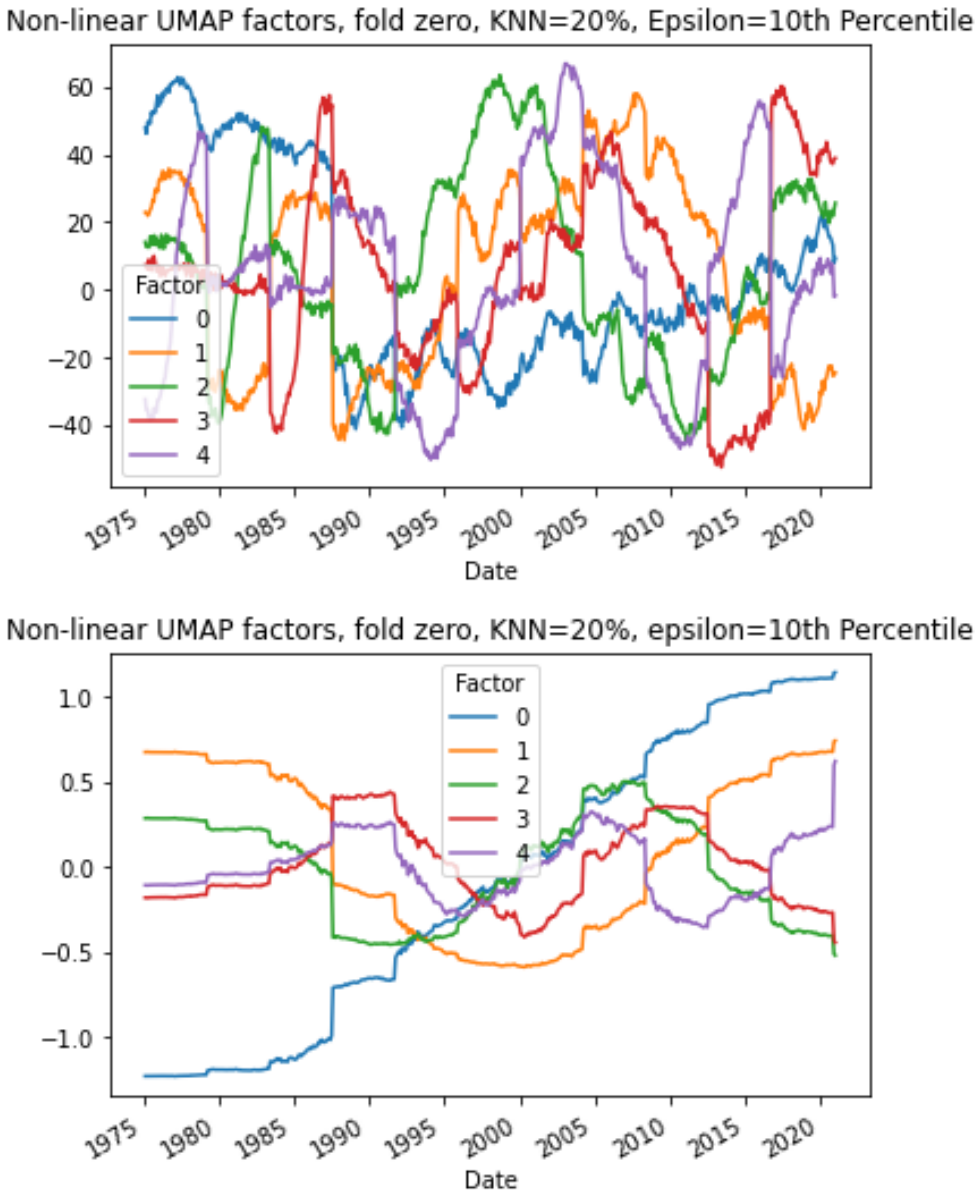

Figure 6: An example of factors extracted from the linear PCA factors using UMAP (top) and diffusion map (bottom). Both algorithms appear to extract factors with oscillatory behavior. Unlike principal-component factors, the different factors do not appear to have dramatically different volatilities.

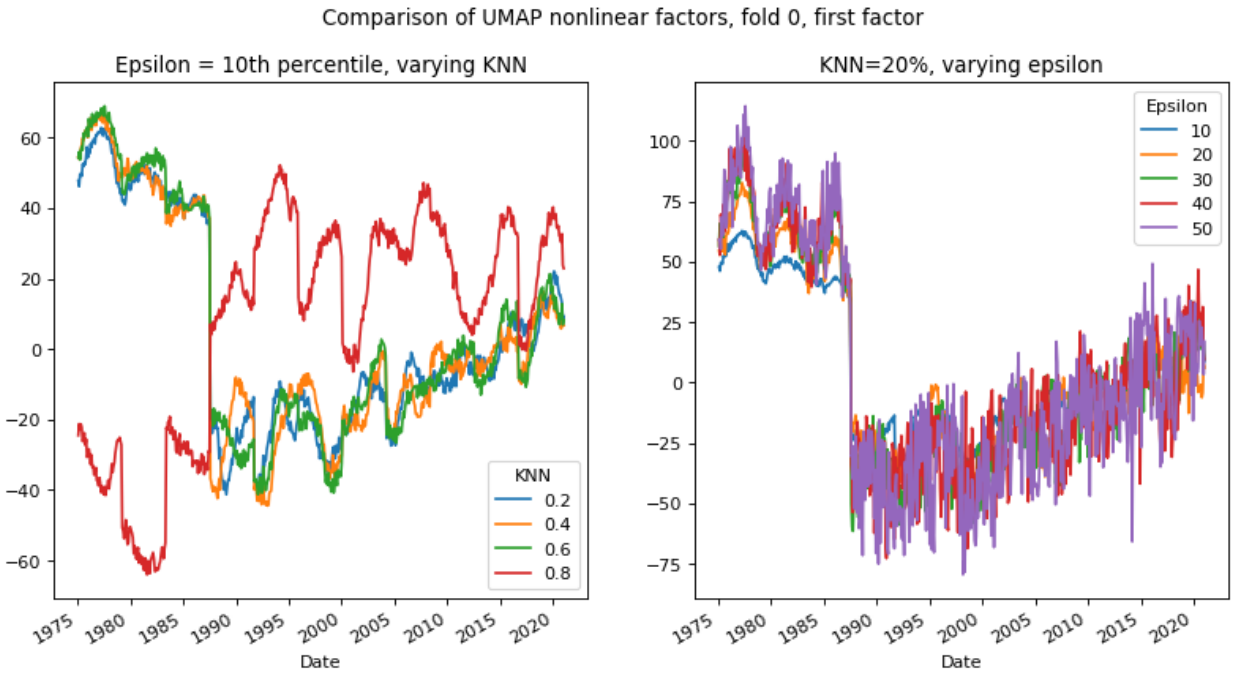

Figure 7: An example of the first factor extracted by UMAP for various KNN parameters (left) and bandwidth parameters (right).

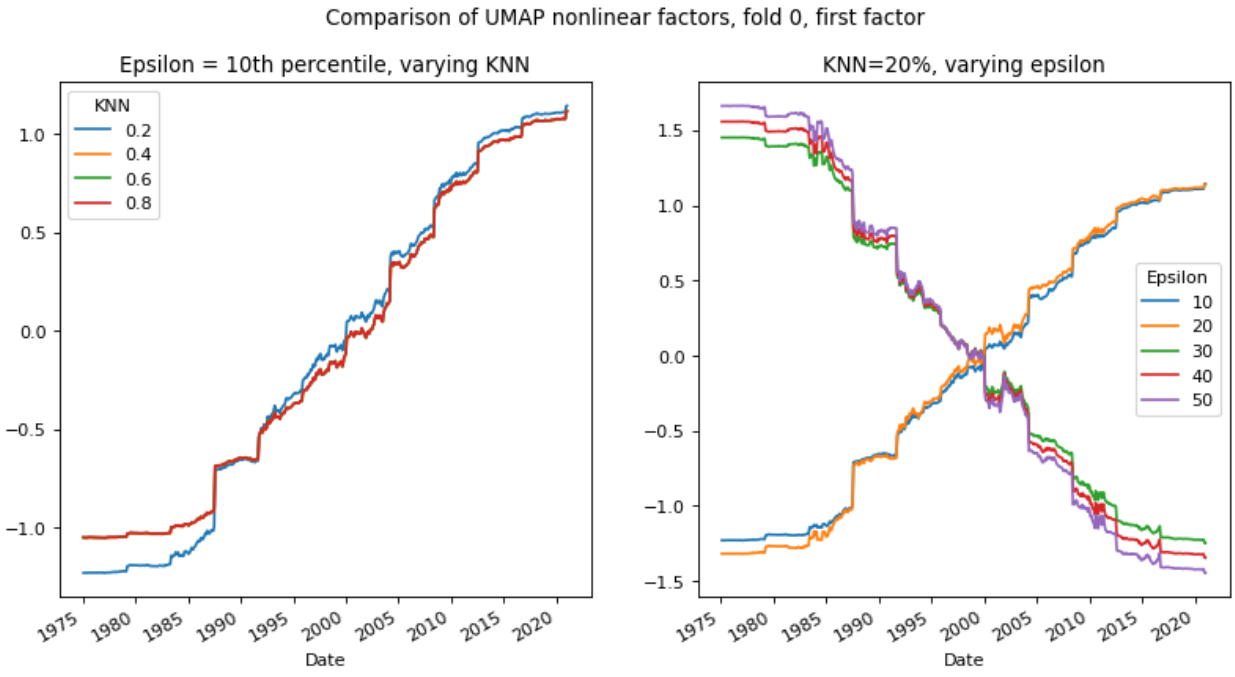

Figure 8: An example of the first factor extracted by diffusion map for various KNN parameters (left) and bandwidth parameters (right).

tease out the influence of the embedding parameters on the latent trajectories, we see that for both UMAP and Diffusion Map it appears that parameter sets emphasizing fine-grained, local structure (smaller values of ϵ and N^{knn}) produce marginal improvements to the explanatory power of the model. By comparing the two figures, we also see that for each choice of parameters diffusion maps offers a higher explanatory R^2, although the difference between the two is fairly small.

Turning to predictive performance, shown in Figure 10, the picture is far less rosy. All parameter choices have a negative R^2_{Pred}, indicating that the variance of the rolling-average-based predictions is *larger* than the variance of the market returns themselves. The R^2_{Pred} values for UMAP are of course consistently negative but within a realistic range, while those of diffusion map are massive, which may indicate numerical precision issues. Focusing on UMAP for this reason, we also make an observation that parameter sets emphasizing small-scale local structure – which produced *better* explanatory performance – here appear to result in more negative predictive performance. Leaning on the intuition of Kelly et al. (2021), it could be that the higher-localization specifications allow for greater flexibility in the pricing model, which will tend to make the predictive R^2 more negative.

We have seen a highly mixed picture in terms of performance, which varies considerably with the choice of parameters. To condense our analysis and get an upper bound on the performance of the embedding models, we focus on the best-performing set of parameters for each of our performance metrics. Recall that we look at two benchmarks. The full linear model uses the 50 principal component factors directly, without performing any dimensionality reduction, providing an upper bound on the information content in the compressed representation. The restricted linear model, in contrast, uses only the first five principal components, so that it is constrained to the same level of flexibility as the manifold-based models but can only identify linear structure in the data.

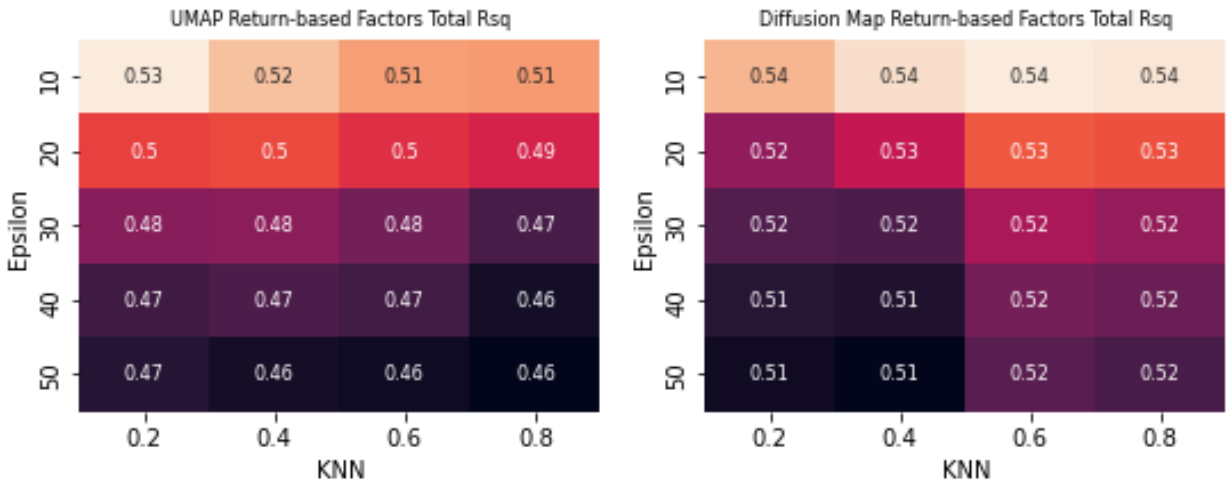

Figure 9: A comparison of the total R^2 for the pricing models which use manifold embeddings to identify latent factors. This measure captures the abiliy of the pricing models to explain variation in contemporaneous returns. UMAP is shown on the left, and diffusion map on the right. Each entry corresponds to one choice of the two key embedding parameters, the kernel bandwidth ϵ and the number of nearest neighbors N^{knn}.

The results are shown in Table 2. Starting with predictive R^2, we can see that the manifold embedding models are in good company, as the full linear model – which is afforded significant explanatory power – also has negative performance. On the other hand, the restricted linear model manages a small positive R^2_{Pred}. Now looking at the total R^2, we can see that manifold embedding algorithms (again, in the best case) are able to extract slightly more explanatory power from their five dimensions than the comparable linear benchmark. However, the full linear model, which uses 50 factors, is able to capture almost all of the cross-sectional variation in the data. We stress that this is an extremely high degree of flexibility, as these 50 latent factors are used to describe the returns of only 200 markets, and so this model gives us only an upper bound on the information content which can be captured. Its strong performance would suggest either that there simply is no lower-dimensional nonlinear structure for the algorithms to identify, or that measurement noise or other estimation challenges prevent them from being captured in a way that adds significant value to the pricing model.

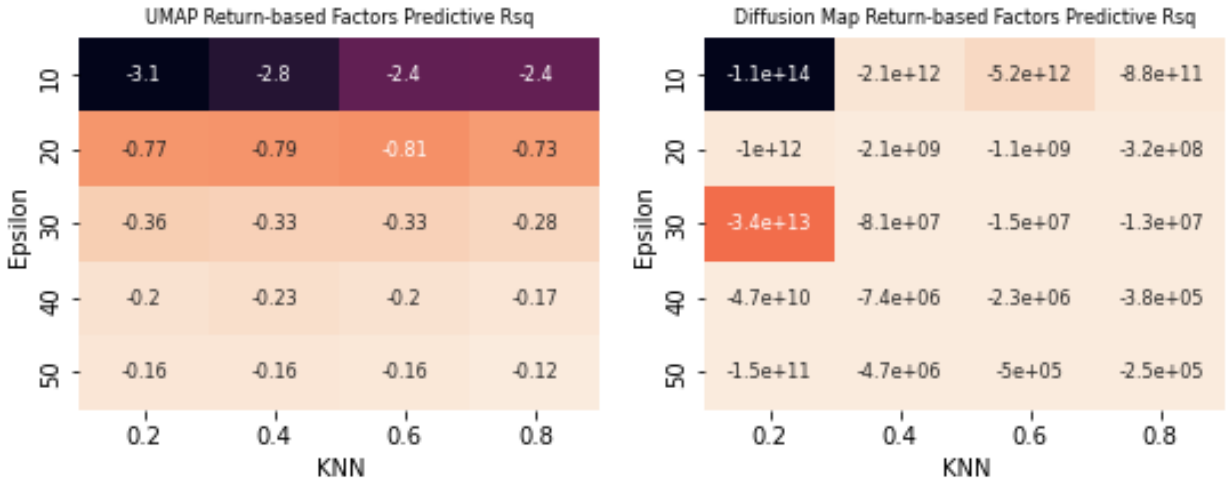

Figure 10: A comparison of the predictive R^2 for the pricing models which use manifold embeddings to identify latent factors. This measure captures the extent to which factor return premia are persistent. UMAP is shown on the left, and diffusion map on the right. Each entry corresponds to one choice of the two key embedding parameters, the kernel bandwidth ϵ and the number of nearest neighbors N^{knn}.

R^2 comparison

Metric	Predictive	Total
Diffusion Map	-	0.541
UMAP	-	0.527
Restricted Linear Model	0.012	0.471
Full Linear Model	-	0.946

Table 2: A comparison of the asset pricing performance of our models. For UMAP and Diffusion Map, we show the best performer across the range of embedding parameters N^{knn} and ϵ considered. Negative R^2 values are suppressed.

4.5.3 Characteristics-based extension

We now discuss the extension of the model which uses characteristic portfolios, rather than principal components analysis, to perform the initial dimensionality reduction. This difference aside, all aspects of our study are the same.

We begin by again examining a heatmap which shows R^2_{Total} and R^2_{Pred} for each choice of embedding parameters. Looking first at Figure 11, which shows the explanatory performance, our results seem qualitatively simillar and marginally better than the models using PCA-based factors. We again see that Diffusion Map performs marginally better than UMAP, and that parameterizations emphasizing local structure outperform those which which measure similarity at a coarser scale.

Our motivation for performing this characteristics-based extension is the study of Kelly et al. (2019), who find that incorporating exogenous information via characteristics brings a dramatic increase to the predictive power of asset pricing models. In particular, in their study a PCA-type model applied to market returns has negative R^2_{Pred}, while their IPCA based model has positive performance. Unfortunately, this is not the case here, as we can see in Figure 12. As in our original model, it appears that the diffusion map model is destabilized and produces huge negative R^2_{Pred} values. The UMAP model appears to perform significantly better with characteristic information – compare the top-left corner here of -0.16 to -3 in Figure 10 – but all values remain negative.

Relative to our previous study, here the restricted linear benchmark, rather than using some of the return-based factors directly, performs a PCA on the full set of characteristic portfolio returns, extracting the top five principal components. Following the same route as our earlier analysis, we contextualize our results by comparing with the benchmark linear models shown in Table 3. The results are qualitatively the same: the manifold models have

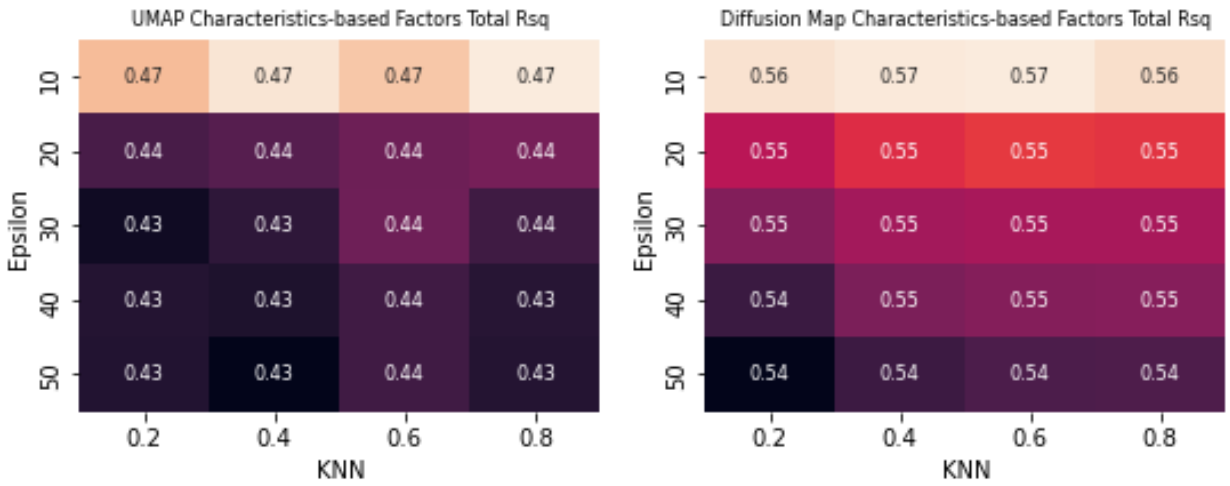

Figure 11: A comparison of the total R^2 for the pricing models which use manifold embeddings to identify latent factors, and use characteristic portfolios as the input rather than principal component factors. This measure captures the abiliy of the pricing models to explain variation in contemporaneous returns. UMAP is shown on the left, and diffusion map on the right. Each entry corresponds to one choice of the two key embedding parameters, the kernel bandwidth ϵ and the number of nearest neighbors N^{knn}.

explanatory performance which is competitive with, but does not significantly outperform, the restricted linear model which is constrained only to identify linear structure in the data. We again see that the full linear model is able to explain nearly 100% of the variation in the return cross-section, and that the restricted model has a small but positive predictive R^2.

R^2 comparison

Metric	Predictive	Total
Diffusion Map	-	0.565
UMAP	-	0.470
Restricted Linear Model	0.009	0.481
Full Linear Model	-	0.954

Table 3: A comparison of the asset pricing performance for the characteristics-based model, where the return of characteristic portfolios is used, rather than PCA-based factors, to compute the embeddings. For UMAP and Diffusion Map, we show the best performer across the range of embedding parameters N^{knn} and ϵ considered. Negative R^2 values are suppressed.

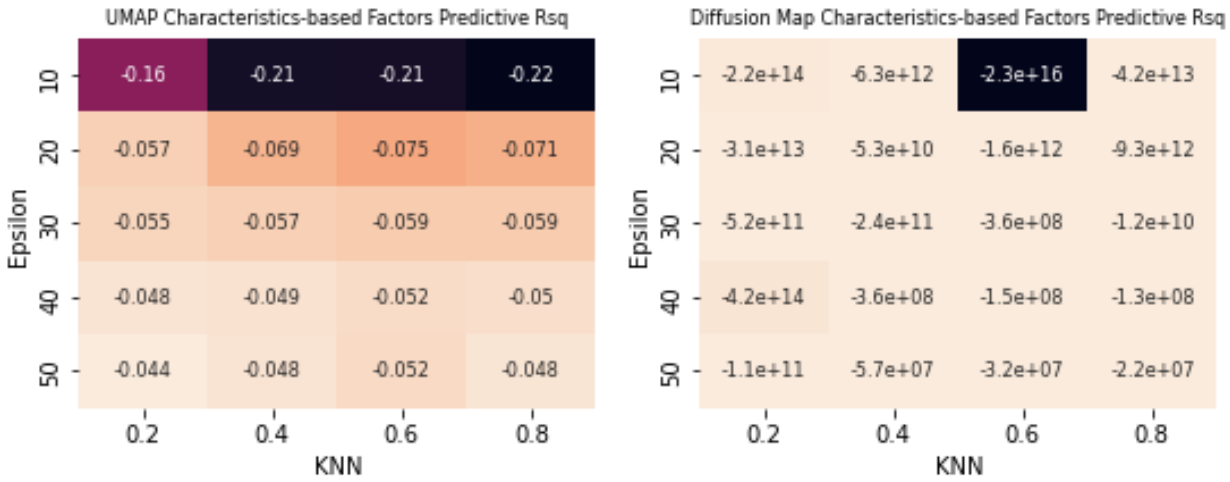

Figure 12: A comparison of the predictive R^2 for the pricing models which use manifold embeddings to identify latent factors, and use characteristic portfolios as the input rather than principal component factors. This measure captures the extent to which factor return premia are persistent. UMAP is shown on the left, and diffusion map on the right. Each entry corresponds to one choice of the two key embedding parameters, the kernel bandwidth ϵ and the number of nearest neighbors N^{knn}.

4.5.4 Example results at the market and date level

We have now seen that in both model structures we have studied, the manifold embedding models fail to identify latent structure that would lead to strong asset pricing performance. Given this, in the following two sections we perform some exploratory analysis which may provide some insight into the weaknesses of these models and areas for potential improvement.

We begin with a closer look at the performance of the models at the individual market level. Throughout, we focus on the UMAP model with return-based factors, where N^{knn} is set to be 20% of the total observations, and ϵ the 10^{th} percentile of the off-diagonals of the distance matrix. Our choice is ad hoc but necessary given that there are 100 models (two algorithms, two types of compression (PCA or characteristic portfolios), and twenty-five parameter sets); we prefer the parameters tuned for the most localized level of structure because this presents in a sense the most flexible model. Recall that this set of parameters

tended to have relatively better explanatory performance and worse predictive performance. We focus on UMAP given the apparent numerical issues with Diffusion Map, a problem we will investigate in the next section.

Beginning with the explanatory performance, we recalculate the R^2_{Total}, restricting our computation to the observations in a given month across all markets, or to a given market across all time. In the latter case, we drop any markets with less than 12 observations (one year of data) to stabilize the calculations; no such filtering is necessary for the calculation over time since there are generally close to the full 1000 available. The results are shown in Figure 13. On the left, we see that the explanatory R^2 is very noisy, but it does not appear that there is any one period which performs particularly well (or poorly). On the right, we show a histogram of the explanatory R^2 by market. The distribution is roughly normal, with a peak around 55% and a fat left tail – unsurprising given that this metric is bounded above (at 1), but not below. It appears that the "average" market performs well, and that the explanatory performance for almost all markets is positive.

Turning our attention now to predictive performance in Figure 14, we present the same analysis by date and by market. We see that there are two periods – around the end of the dot com era in the mid 2000s as well as in the post financial crisis era of the early 2010s – when R^2_{Pred} turned steeply negative. These market turning points may present unique challenges to the pricing models, because there may be a significant change in the market covariance structure as well as a large number of names coming into or going out of the market set due to delistings or firm consolidation. However, even outside these periods we see that the performance was still consistently negative. The market-level analysis affirms this conclusion. We again see a long left tail in the performance of individual markets, but it appears that the vast majority of markets have $R^2_{Pred} < 0$. This, unfortunately, suggests that the challenges to the model are broad-based and not an issue of bad data or a few

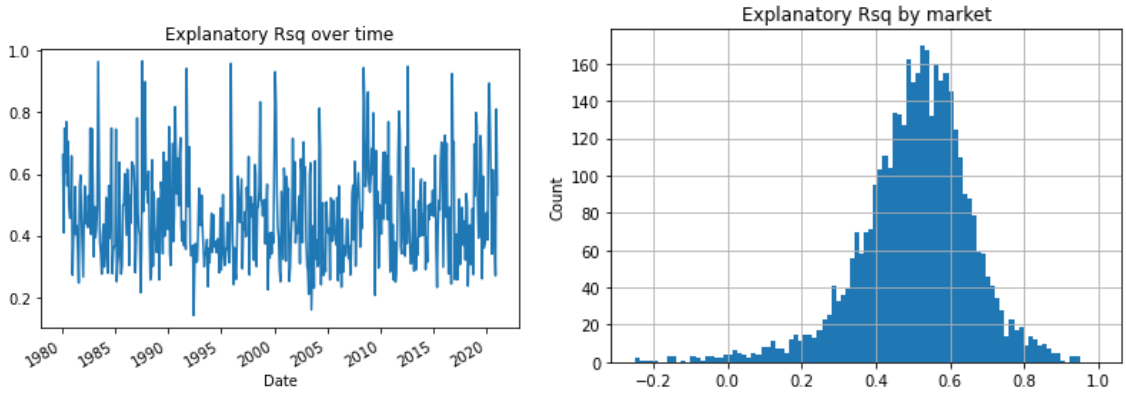

Figure 13: Explanatory R^2 over time and by market for a pricing model computed at the month-level (left), and market level (right).The pricing model here uses UMAP for the embeddings and return (PCA-based) factors. N^{knn} is set to be 20% of the total observations, and ϵ the 10^{th} percentile of the off-diagonals of the distance matrix.

poorly-behaved markets skewing our metrics. Either our methodology is not data-efficient enough to identify latent structure in the data, or there simply is no such nonlinear structure to exploit.

4.5.5 Conditioning problems in the covariance matrix

Recall that our embedding algorithms perform dimensionality reduction in the space of distributions using the K-L divergence. For the rolling normal model which we fit to the data, this metric requires inverting the sample covariance matrix. If this covariance matrix is ill-conditioned, this implies that our distance metrics may lose considerable precision. This appears to be the case and may explain why the manifold embedding models do not outperform the simple benchmarks.

We begin our exploration of this issue with a simple exercise whose result is shown in Figure 15. At the end of 1995, we construct the market covariance matrix using the last five years of observations. Because the number of markets being used vastly exceeds the number of observations, this matrix is by construction not positive definite. However, the compression to principal components - a low rank approximation - can be. For a varying number of

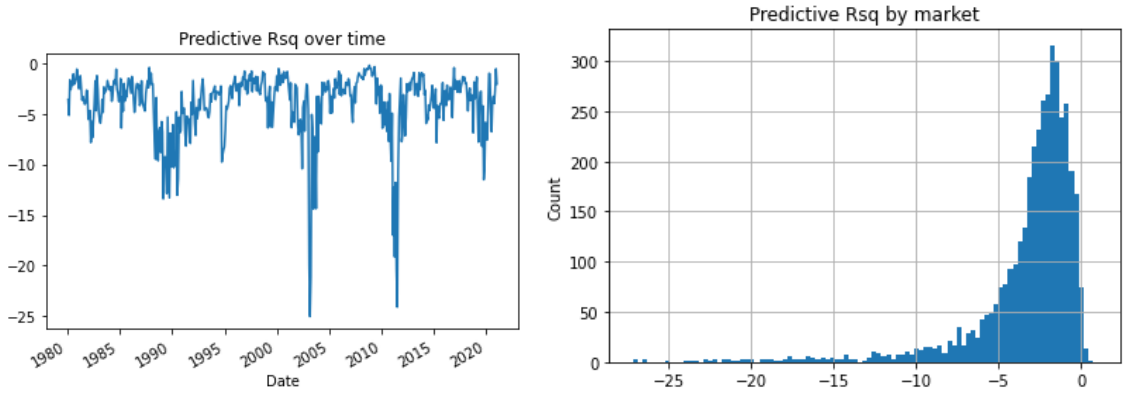

Figure 14: Predictive R^2 over time and by market for a pricing model computed at the month-level (left), and market level (right).The pricing model here uses UMAP for the embeddings and return (PCA-based) factors. N^{knn} is set to be 20% of the total observations, and ϵ the 10^{th} percentile of the off-diagonals of the distance matrix.

principle components, we compute the compression and the resulting covariance matrix of principal components, finally computing its condition number. As seen in Figure 15, we can see that the condition number increases steadily as more principal components are used. This indicates that as we reach further into the structure of the market covariance matrix, we are able to measure covariances with less and less precision. This is unsurprising, since each additional principal component we add has, by definition, lower variance than the others being used. This would also be apparent examining the eigenvalue spectrum of this matrix, which decays rapidly.

Having built some intuition about the situation, we now confirm that this is a practical issue in Figure 16, which compares the condition number of the fold zero covariance matrix over time in both the PCA-based model and the characteristic-based extension. The result is unambigous: in both cases, the linear factor covariance matrix is very poorly conditioned. Using the rule of thumb that the number of digits lost is $\log 10$ of the condition number, in the return-based case we lose between three and four digits of precision, and the characteristics-based case even more, between seven and eight. By definition our principal component factors are uncorrelated to within numerical error, but this is not the case for

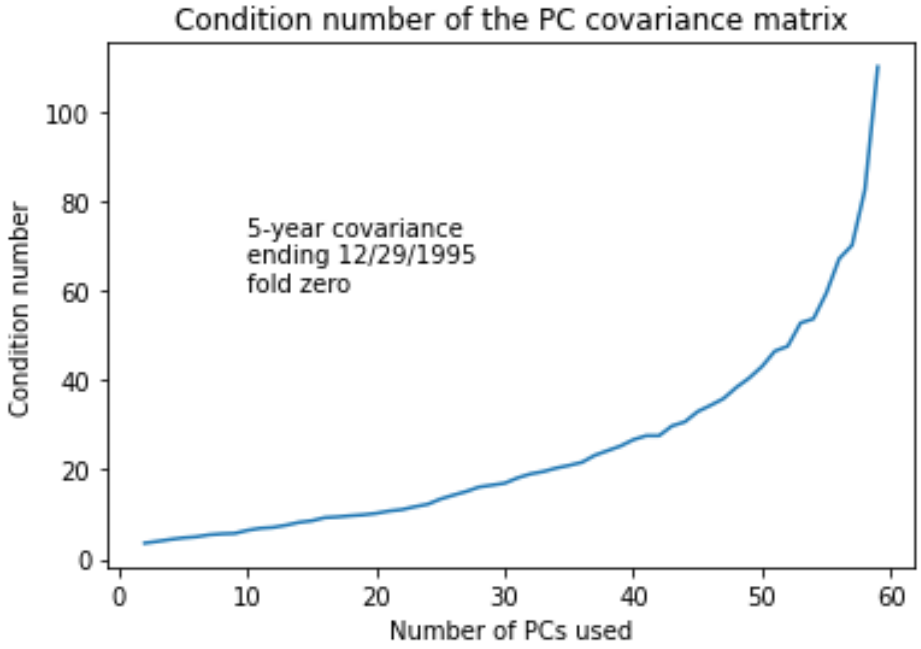

Figure 15: An example of the condition number of the market covariance matrix, compressed to a given number of principle components.

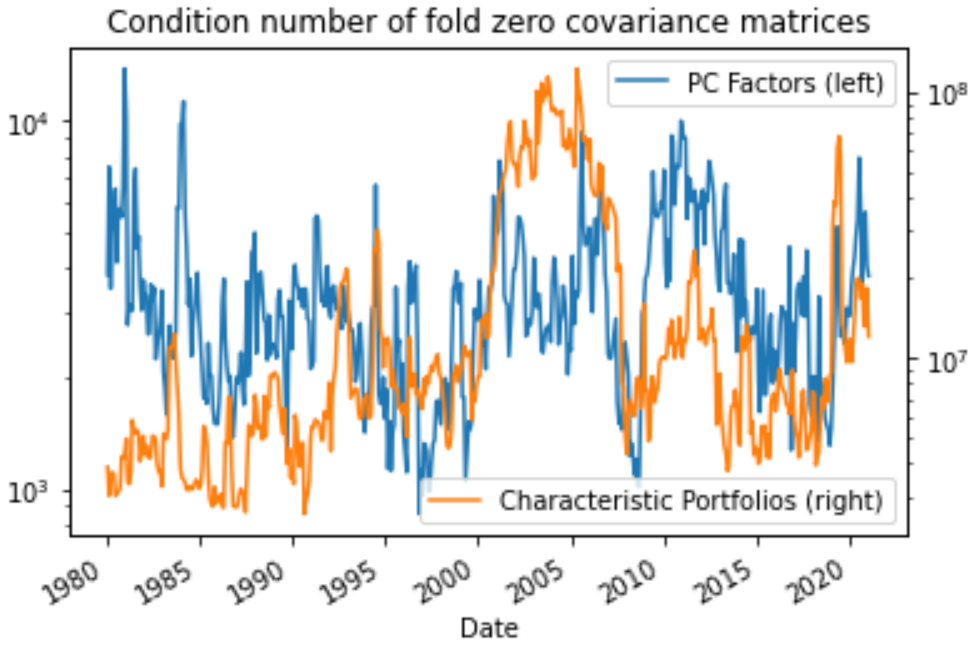

Figure 16: A comparison of the condition numbers of the factor covariance matrices for fold zero, for the market return (PCA-based) factors,on the left, and characteristic portfolios, on the right.

the characteristic portfolios. It is likely that collinearity of individual characteristics is what drives the condition number significantly higher. In either case, we see that there is likely substantial loss in precision when inverting the linear factor covariance matrices. This means that the input to the embedding algorithms, which must invert these covariance matrices to compute distance, are also subject to substantial losses in precision, which may explain some of the disappointing performance.

The intention for using a very large number (50) of linear factors was to avoid prematurely constraining the manifold embedding algorithms, but this appears to come with a severe tradeoff. One simple solution would simply be to use a smaller number of linear factors, perhaps chosen as the largest number which avoids a worst-case condition number, which will reduce model flexibility but also mitigate the numerical issues. An alternative approach would be to precondition the matrix before calculating the K-L divergence in order to identify a sparse approximation which will be more numerically well-behaved. A complementary measure would be to shrink the off-diagonals (particularly in the characteristics extension) in order to reduce the influence of potentially large and unstable sample correlations on the inversion.

We further explore this issue in appendix B. We re-fit the model with smaller numbers of latent factors, with the idea that this will reduce the ill-conditioning problem which introduces numerical noise during the embedding step. While we do see an improvement in conditioning, this does not translate to significantly better predictive performance. For models with fewer linear factors (smaller N_l), the numerical blow-up in the Diffusion Map R^2_{Pred} is resolved, but we continue to see negative predictive performance across the board for the manifold embedding-based pricing models.

5 Future work and conclusion

We have seen from our empirical study that the manifold learning-based techniques do not offer a substantial improvement over their more constrained, linear counterparts. We conclude this work with some additional thoughts on the drivers of this underperformance, and suggestions for future work which could improve on the model and our understanding of the problem.

5.1 Future work

5.1.1 Improved model tuning and parameter search

In our empirical study, the choice of model parameters was decidedly ad hoc. There are a large number of parameters to choose, and while for many we may draw some analogy to the empirical finance literature to guide our decision, for others - particularly the crucial embedding parameters N^{knn} and ϵ - there is no obvious parallel or intuitive guide.

A more thoughtful approach would be to leverage techniques from stochastic search and optimization to iteratively update parameters, directly optimizing the cross-validated explanatory performance. Because the manifold embeddings do not have a closed functional form from which to compute a gradient, the workhorse method from the neural network literature – stochastic gradient descent – cannot be applied. The simultaneous perturbation stochastic approximation (SPSA) algorithm of Spall (1992) provides an efficient approach to estimate the gradient numerically using only two evaluations per iteration. It is likely that a directed search using this technique, rather than our coarse "grid search", could identify a more promising range of parameters, even given the same limited budget of evaluations (given the significant amount of time and computing power required to re-fit the model). This would help mitigate the degree to which poor choices of parameters, rather than model structure, prevents our approach from succeeding.

5.1.2 Simulation studies

As we saw, the model investigated in this research failed to outperform more constrained benchmarks. It is important to understand whether this is because, given the parameters and structure of the model, it failed to converge, or whether there is simply no nontrivial structure for the more complex manifold embedding algorithms to identify and utilize.

An avenue to address this issue would be to perform simulation or Monte Carlo studies. The manifold embedding algorithms could be fit using inputs from a known data-generating process, which has an embedded manifold structure and similar stylized properties as financial data (such as positive long-term drift, a sharply decaying covariance eigenvalue spectrum, and volatility clustering). In such a study, the true structure would be known and could be compared against the model fit. This would give insight into the speed of convergence and finite-sample properties of the model. If the model performs extremely well on the simulated data, it would suggest that the core assumption of an embedded manifold structure in the cross-section is incorrect. If the model fails to converge even on idealized, simulated data, further analysis may suggest areas for improvement.

5.1.3 Parameteric embeddings and manifold regularization

A limitation of our approach is that the embeddings produced by UMAP and Diffusion Map are nonparameteric: we do not have a functional form that allows us to map new observations onto the set of latent dimensions. This is a well-known challenge and non-parametric techniques have been developed in the literature. The Nyström extension method, following the technique of Bengio et al. (2003), allows spectral methods such as diffusion map to extend their learned embeddings to new observations. Similarly, the UMAP embedding can be extended to new observations in a fairly straightforward manner using a similar objective function to the original embedding. The primary stumbling block here is in the implementation, as our use of a non-standard distance metric departs from the support of

the high-quality libraries `pydiffmap` and `umap`.

A complementary area of research provides some synthesis between manifold embedding algorithms and deep learning by learning a functional form for the embedding using a neural network. Sainburg et al. (2021) develop such an algorithm for UMAP and find that the in-sample performance is competitive with the non-parameteric embeddings, while allowing new data to be mapped much more quickly. Duque et al. (2020) consider manifold learning in general and show that this technique corresponds to a kind of geometric regularization, which drives the neural network to prioritize identifying global structure in the crucial bottleneck layer. This kind of regularization could be adapted to the technique of Gu et al. (2021), which directly uses an autoencoder for asset pricing.

A functional form that can be used to compute out-of-sample embeddings would improve the computation time of the algorithm. In addition, having a way to embed new data points would also allow us to perform the temporal train/validate/test split which is more common in the recent empirical finance literature, and to "walk forward" the model from one point in time to the next, which would of course be crucial for applications in industry.

5.1.4 Factor-instrumented embeddings

A shortcoming of our approach is that it is qualitatively very "data hungry": we first consume return information to construct PC factors or characteristic portfolio returns, then use the information in the time series dynamics to construct manifold embeddings. Because the resulting embeddings have no functional form and do not represent investible portfolios, we must *then* estimate the risk exposures of each market to each latent factor using a rolling regression before computing asset pricing performance. Each of these steps consumes a large amount of data and introduces sampling error, and so it is possible that our methodology would perform well with arbitrarily large datasets but is simply not efficient enough to con-

struct reasonable estimates given the limited amount of data available.

The step of estimating risk factor exposures is particularly irksome, as the long lookback required to produce stable beta estimates may prevent the dynamics of the latent factors from factoring into the pricing model. With an additional twist, the parameterized embeddings discussed in section 5.1.3 could be used to circumvent this step. Rather than using a neural network to parameterize the embedding in terms of the input data, we could use a more general neural network which parameterizes the embedding directly in terms of the linear factors. If a reasonable approximation can be obtained, this would give us a functional form that allows us to express the market-level latent risk factor exposures (betas) as a non-linear function of the factor (principal component loading or characteristic value) weights. This would sidestep the beta-estimation step in our methodology, allowing the market risk exposures to be estimated in a more efficient and consistent way, as well as allowing them to immediately respond to changes in their principal component loadings or characteristics without the auto-correlation imposed by a rolling regression calculation. This suggestion borrows from the augmented autoencoder architecture developed by Gu et al. (2021), who see large gains in predictive power from incorporating characteristic information and for allowing nonlinearity in the mapping from observable market information to latent risks.

5.1.5 Analysis of other asset classes

If the model could be improved to show promise on equities data, it would be equally interesting if not moreso to examine its performance modeling the cross-sectional variation of other asset classes. The data-driven nature of this model reduces the risk of misspecification compared to models which rely on factors hand-picked by a researcher, and (following some of the paths suggested in the previous section) can even benefit from the vast literature on proposed factors by using them as instruments. Statistical models have even more to contribute in other asset classes where the factor structure is less well understood, and there

are fewer proposed factors.

Two asset classes of particular interest would be the corporate bond market and currencies. Corporate bonds share many similarities to cash equities and have a younger but rapidly growing literature on cross-sectional return variation, see e.g. Kelly et al. (2020). Our technique could contribute to this and perhaps even merge the information from cash equities and corporate bonds, using the flexibility of manifold embedding to bridge the strongly coupled but nonlinear relationship between the two. Currencies (including foreign exchange, cryptocurrencies, and precious metals), which lack an intuitive or fundamentally-motivated factor structure, constitute an interesting area of study precisely because of their differences with equities. Strong performance from empirical models in this realm could shed light on a less-studied (and perhaps, more difficult) problem, and become a tool for developing and testing macrofinancial theories.

5.2 Conclusion

This thesis develops an asset pricing model using modern manifold learning techniques. Our approach takes a step closer to directly representing the latent linear structure articulated in Ross (1976), but comes with significant estimation and numerical challenges. While the model developed herein fails to outperform more constrained linear models, we hope that this is an "interesting failure." Further investigation could determine if the shortcomings of this approach can be overcome, or perhaps if there simply is not compelling evidence of latent manifold structure in the return cross-section. Either outcome would advance our knowledge of this difficult and complex problem, and so constitutes a promising avenue for future research.

A Software

A.1 Python packages

The analysis was produced using Python 3.9 using the CPython interpreter. The standard scientific packages `numpy`, `scipy`, and `scikit-learn` were used. `pandas` was used for data analysis and `matplotlib` and `seaborn` for plotting. The libraries `pydiffmap` and `umap-learn` were used for implementations of diffusion maps and UMAP, respectively.

A.2 Diffusion maps implementation

The `pyDiffMap` library provides an excellent implementation of the diffusion maps algorithm. Unfortunately, it does not support passing a pre-computed distance matrix for the kernel calculation. Because this matrix is $T \times T$, where T is the number of days in the sample, computing this matrix is extremely slow and makes running the algorithm many times (e.g. for parameter tuning) infeasible.

As a workaround, the library was forked and the code modified to support passing a pre-computed distance matrix. This drops the runtime substantially and was tested to produce the same result to within machine precision. The forked repository is available publicly at: `https://github.com/michael-baeder/pyDiffMapPrecomputeDistance`.

This implementation can be used by taking the following steps:

1. Create a local copy of the linked repository, either by downloading the files directly from GitHub or cloning them to a local repository.
2. Open a command window which has access to `pip`.
3. From the command window, run `pip install -e path\to\saved\files`. This will install the local version of the package.

4. To verify that the files have been successfully installed, open a Python session and run `import pydiffmap` followed by `diffmap.__file__`. This should return the same location where the files were saved.

B Additional studies

In this appendix, we briefly expand on the results presented in Section 4 to consider pricing models based on different numbers of linear factors. We saw in that section that the very large number ($N^l = 50$) used in the primary study leads to numerical precision problems in the linear factor covariance matrix, a crucial input to the manifold embedding step. There is a tradeoff inherent in this process – using large N^l affords the manifold embedding a great deal of flexibility, but also makes the method more sensitive to numerical precision issues.

Our results suggest that the conditioning problem is indeed resolved by using a smaller number of linear factors, with a marked improvement with $N^l = 25$, a number still relatively large compared to the latent dimensions ($N^n = 5$) recovered. Unfortunately, the improvement in conditioning does not have a large impact on explanatory power and does not produce positive predictive performance. While we do see improvement in R^2_{Pred} across all models when using fewer linear factors, the values are still consistently negative, and hence continue to underperform the restricted linear benchmark.

B.1 Study parameters

Our parameters follow exactly as in 5.1.3 with one difference: we vary the number of linear factors N_l created in the pre-embedding step. This process is straightforward, as it merely means selecting fewer principal components or characteristics. We repeat our analysis for $N_l = 5, 10$ and 25.

B.2 Results

B.2.1 Ill-conditioning

Figure 17 compares the condition numbers of the linear factor covariance matrices by N_l, for both the return and characteristics-based models. We can see from these figures that the ill-conditioning problem is substantially reduced when we use fewer factors. In both models, the expected precision improvement (roughly $\log_{10} \kappa$) is around 2 digits for going from 50 to 25 linear factors, with a more marginal improvement for reducing the factor set further. This improvement suggests we may still be able to use a fairly flexible set of inputs while avoiding the numerical issues with the $N^l = 50$ case presented in our main study.

B.2.2 Asset pricing performance

We now turn our attention to the asset pricing performance. As in Section 4, we report the performance of the *best*-performing choice of parameters for UMAP and Diffusion Maps, which is an upper bound on their expected performance. We omit the full linear model for brevity, as its role was primarily to provide an upper bound on model flexibility which was achieved in the main study. Figure 13 compares the R^2_{Total} for each model as we vary the N^l parameter. In all cases except for the characteristic-based UMAP model, we see a general upward trend in performance as the number of linear factors is increased. This is intuitive, as the additional factors afford the models greater flexibility in uncovering latent common factors. Note that we see basically no change in performance for the restricted linear model of returns, which is intuitive: since that model simply subsets to the $N^n = 5$ principal components with the largest variance, its representation of the latent factors is not affected by changing the dimension of the input set.

Table 4 presents the predictive performance. Unfortunately, despite the fact that using smaller values of N^l addresses the conditioning problem, this does not result in positive

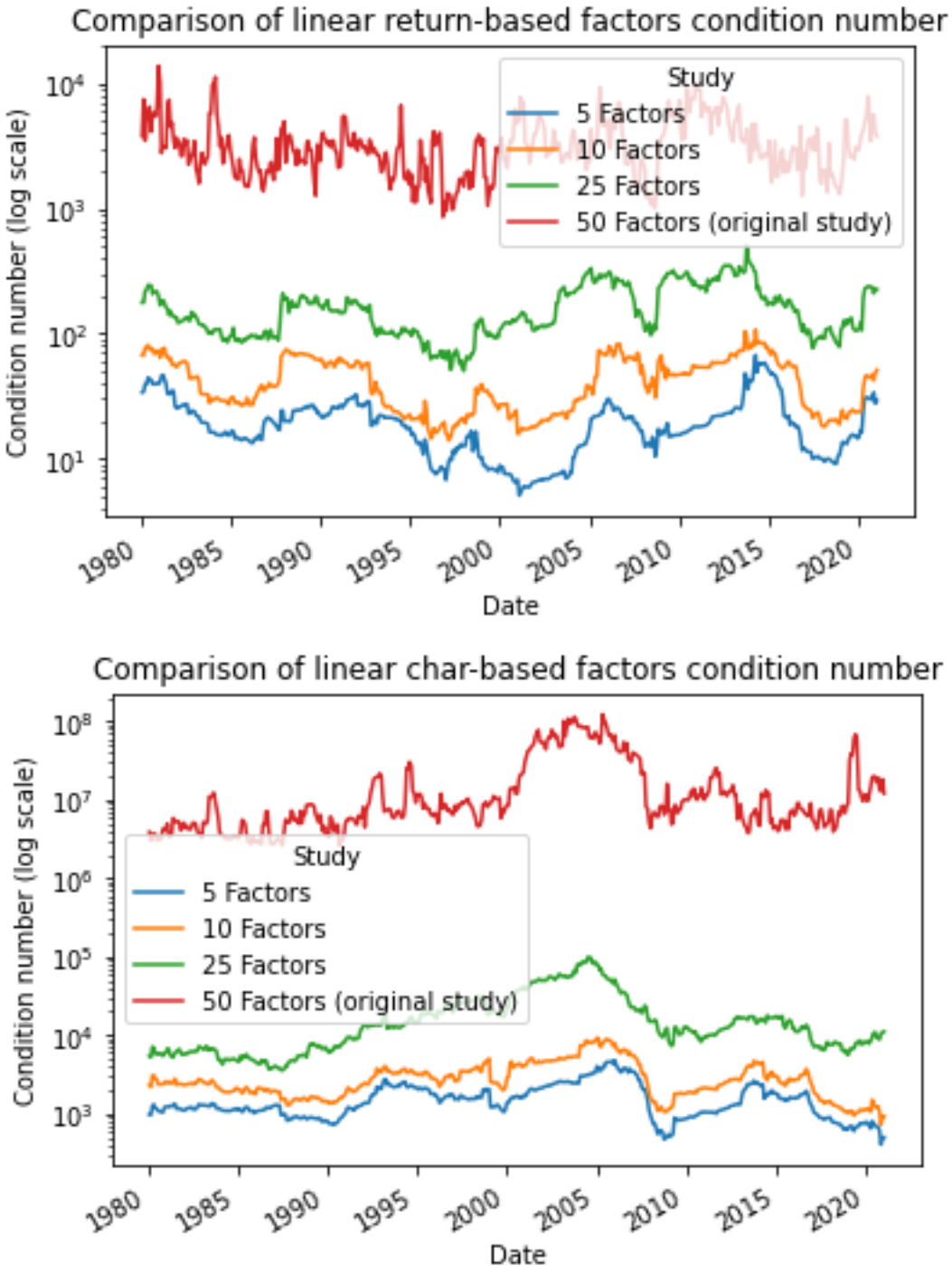

Figure 17: A comparison of the condition number for the linear factor covariance matrix for fold zero in the additional studies. We use a log-scale to make the results easier to compare.

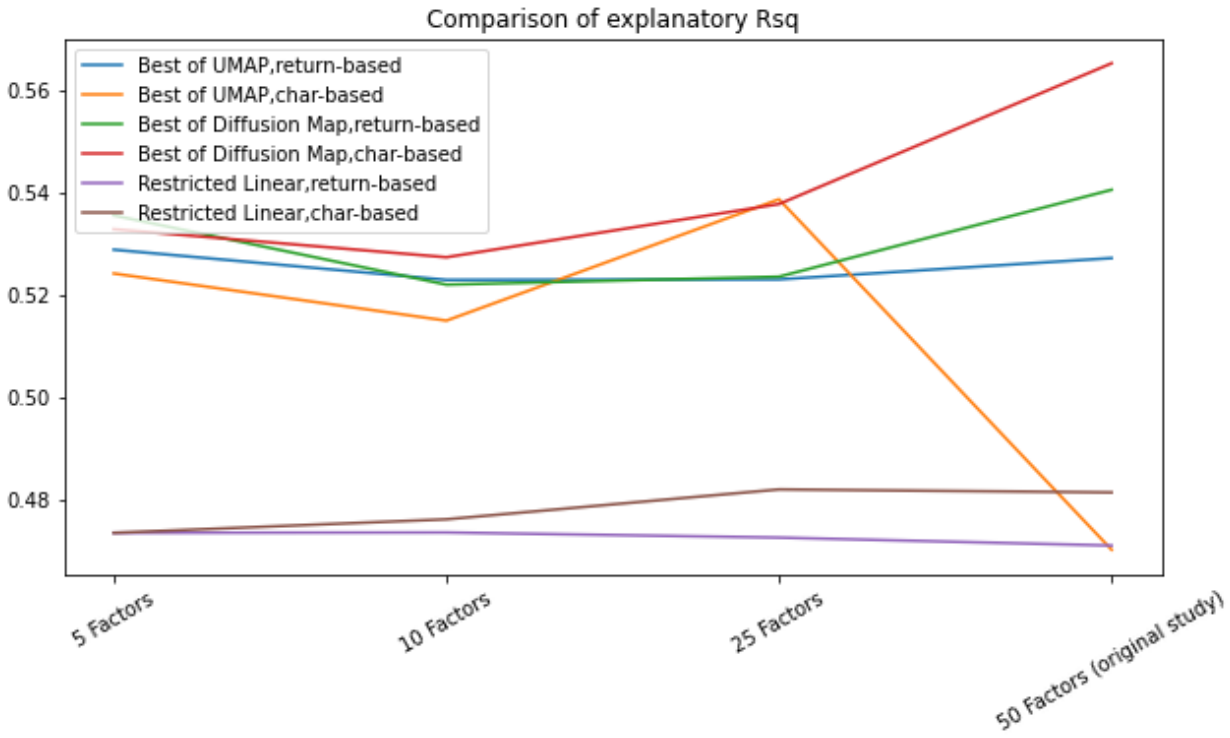

Figure 18: A comparison of R^2_{Total} values for the additional studies. For the manifold embedding models, we show the *best* performer among the set of parameters considered. Note that the y-axis range is fairly small. See section B.1 for additional details on the parameters.

R^2_{Pred} **comparison**	5 Factors	10 Factors	25 Factors	50 Factors
Best of UMAP, return-based	-1.11E+01	-9.89E+00	-3.77E+00	-1.24E-01
Best of UMAP, char-based	-4.10E+00	-1.75E+00	-1.73E+00	-4.43E-02
Best of Diffusion Map, return-based	-8.32E+00	-2.82E+01	-2.23E+00	-2.48E+05
Best of Diffusion Map, char-based	-2.30E+00	-7.70E-01	-2.20E+00	-2.20E+07
Restricted Linear, return-based	1.12E-02	1.25E-02	1.18E-02	1.18E-02
Restricted Linear, char-based	1.05E-02	1.08E-02	1.09E-02	8.86E-03

Table 4: A comparison of R^2_{Pred} values for the additional studies. Rows indicate the model (algorithm used to construct pricing factors, and the way that linear factors are constructed), while the columns indicate the number of factors used in the linear pre-processing step. For the manifold embedding models, we show the *best* performer among the set of parameters considered. See section B.1 for additional details on the parameters.

predictive performance. Focusing particularly on $N^l = 25$ versus 50, we see R^2_{Pred} is dramatically improved for Diffusion Map, reflecting the influence of numerical precision issues on our initial study. In the case of UMAP, the picture is less clear, as using a smaller number of factors appears to hurt the predictive performance. Across the board, we again see that the manifold embedding models fail to achieve a positive R^2_{Pred} and lag behind the restricted linear benchmark.

Bibliography

Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15, 2003.

Yoshua Bengio, Jean-François Paiement, Pascal Vincent, Olivier Delalleau, Nicolas Le Roux, and Marie Ouimet. Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering. *NIPS'03: Proceedings of the 16th International Conference on Neural Information Processing Systems*, pages 177–184, 2003.

Mark M. Carhart. On persistence in mutual fund performance. *The Journal of Finance*, 52 (1):57–82, 1997. doi: 10.1111/j.1540-6261.1997.tb03808.x.

Kevin M. Carter, Raviv Raich, William G. Finn, and Alfred O. Hero, III. Information-geometric dimensionality reduction. *IEEE Signal Processing Magazine*, 28(2):89–99, 2011. doi: 10.1109/MSP.2010.939536.

Gary Chamberlain and Michael Rothschild. Arbitrage, factor structure, and mean-variance analysis on large asset markets. *Econometrica*, 51(5):1281–1304, 1983. doi: 10.2307/1912275.

Andrew Y. Chen and Tom Zimmermann. Open source cross sectional asset pricing. *Critical Finance Review*, Forthcoming.

Ronald Coifman and Matthew Hirn. Diffusion maps for changing data. *Applied and Computational Harmonic Analysis*, 36(1):79–107, 2014.

Ronald Coifman and Stéphane Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 2006.

Andres F. Duque, Sacha Morin, Guy Wolf, and Kevin Moon. Extendable and invertible manifold learning with geometry regularized autoencoders. *2020 IEEE International Con-*

ference on Big Data (Big Data), Dec 2020. doi: 10.1109/bigdata50022.2020.9378049. URL http://dx.doi.org/10.1109/BigData50022.2020.9378049.

Eugene F. Fama and Kenneth R. French. Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics*, 33(1):3–56, 1993. doi: 10.1016/0304-405X(93)90023-5.

Eugene F. Fama and Kenneth R. French. A five-factor asset pricing model. *Journal of Financial Economics*, 116(1):1–22, 2015. doi: 10.1016/j.jfineco.2014.10.010.

Eugene F. Fama and James D. MacBeth. Risk, return, and equilibrium: Empirical tests. *Journal of Political Economy*, 81(3):607–636, 1973. doi: 10.1086/260061.

Shiaho Gu, Bryan Kelly, and Dacheng Xiu. Autoencoder asset pricing models. *Journal of Econometrics*, 222(1):429–450, 2021. doi: 10.1016/j.jeconom.2020.07.009.

Shihao Gu, Bryan Kelly, and Dacheng Xiu. Empirical asset pricing via machine learning. *The Review of Financial Studies*, 33(5):2223–2273, 2020. doi: 10.1093/rfs/hhaa009.

Brian Kelly, Diogo Palhares, and Seth Pruitt. Modeling corporate bond returns. *Journal of Finance (forthcoming)*, 2020. doi: 10.2139/ssrn.3720789.

Bryan Kelly, Seth Pruitt, and Yinan Su. Instrumented principal components analysis. Working paper, Chicago Booth and ASU WP Carey, 2017. Available at SSRN: https://ssrn.com/abstract=2983919.

Bryan T. Kelly, Seth Pruitt, and Yinan Su. Characteristics are covariances: a unified model of risk and return. *Journal of Financial Economics*, 134(3):501–524, 2019. doi: 10.1016/j.jfineco.2019.05.001.

Bryan T. Kelly, Semyon Malamud, and Kangying Zhou. The virtue of complexity in machine learning portfolios. Research Paper 21-90, Swiss Finance Institute, 2021.

Wenzhao Lian, Ronen Talmon, Hitten Zaveri, Lawrence Carin, and Ronald Coifman. Multivariate time-series analysis and diffusion maps. *Signal Processing*, 2015.

Harry Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, 1952. doi: 10.2307/2975974.

Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction, 2020.

Stephen A. Ross. The arbitrage theory of capital asset pricing. *Journal of Economic Theory*, 13(3):341–360, 1976. doi: 10.1016/0022-0531(76)90046-6.

Tim Sainburg, Leland McInnes, and Timothy Q Gentner. Parametric umap embeddings for representation and semi-supervised learning, 2021.

William F. Sharpe. Capital asset prices: A theory of market equilibrium under conditions of risk. *The Journal of Finance*, 19(3):425–442, 1964.

J.C. Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, 37(3):332–341, 1992.