Asset Pricing in China's Stock Market*

Shuaiyu Jiang†
School of Management and Economics, The Chinese University of Hong Kong, Shenzhen

Peiyuan Zhang‡
Department of Economics, Peking University

This paper uses traditional machine learning methods and deep neural networks based on both firm-specific characteristics and macroeconomic variables to price China's A-share stock market. The main contributions are as follows: We give the stochastic discount factor a flexible form and compare different models' performances. Since the Chinese government adopts various policies to maintain financial stability, we borrow the idea from generative adversarial network to find the true SDF by selecting moment condition that minimizes return volatility. Additionally, we compare this model's performance with Chen's work and find that this model can obtain higher Sharpe ratio and R^2.

Keywords: Asset Pricing,Machine Learning,Neural Network,Chinese Market

I. INTRODUCTION

The Chinese capital market, despite its short history, has become the second largest in the world. An essential feature of China's stock market is that Chinese government leans against short-term market fluctuations actively to promote financial stability. The government achieves this goal through frequent policy changes, ranging from changes in interest rates and bank reserve requirements to stamp taxes on stock trading, suspensions and quota controls on IPO issuances, modifications to rules on mortgage rates and first payment requirements, and direct trading in asset markets through government-sponsored institutions. These tight controls once made China's market isolated and hard to understand. As a consequence of these large-scale interventions, China's financial markets are highly speculative and largely populated by inexperienced retail investors. In 2008, the China Securities Regulatory Commission issued the China Capital Markets Development Report, which shows retail accounts with a balance of less than 1 million RMB contributed to 45.9% of stock positions and 73.6% of trading volume on the Shenzhen Stock Exchange. This fact highlights speculative behavior of small investors and lack of mature institutional investors as important 'Chinese Characteristics'. The market experiences high price volatility and the highest turnover rate among major stock markets in the world.

This special market structure motivates me to consider how to price the stock market correctly and whether the market is efficient. On the one hand, small investors' actions are hard to predict because they are not well equipped with financial knowledge and their behaviour is full of uncertainty. Therefore, the market might be more consistent with the random walk hypothesis. On the other hand, since there are irrational factors in such a financial market, speculation might make traditional models fail to price stocks successfully and cause larger pricing errors. Additionally, Chinese government's frequent intervention plays an important role in reducing market volatility, which prevent stocks' prices from changing too much. Xiong Wei, etc develop a comprehensive theory framework for China's model of managing the financial system in which investors with a highly speculative nature only care about short-term return rate and asset fundamental is unobserved because of realistic information frictions faced by investors and policy makers. Noise traders reflect inexperienced retail investors in China's stock market and they contribute to price volatility and instability. Under these conditions government's intervention is necessary and there is a trade off between ensuring financial stability and improving information efficiency because government intervention makes noise in government policy an extra factor in asset pricing.

The main idea of this paper is to model asset prices under the background of China's stock market. Firstly we give a brief introduction of the framework of asset pricing theory and how the theory can be applied on real data under the framework of generalized method of moments. This part is mostly based on John Cochrane's outstanding work. Then we start from a simple linear regression model to predict stock prices, but OLS estimator may suffer from too many covariates thus the estimation is not reliable. Machine learning methods, such as Lasso, Elastic Net and Tree regression propose a solution to deal with high dimension data. Through adding penalty to norm of coefficients and non-parametric methods, models' performances become much better. Additionally, as computers become increasingly powerful, deep learning network is increasingly widely used for better non-linear performance. We compare these models' results using a comprehensive dataset of China's A-share stock market including both firm-specific characteristics and macroeconomic variables. However, empirical results show that

* We gratefully acknowledges careful and patient guidance from Shaoran Li.

† shuaiyujiang@cuhk.link.edu.cn
‡ zpyuan@pku.edu.cn

these models fail to make accurate out-of-sample prediction and the R^2 is low, which means that existing theory and empirical methods don't offer powerful economic explanation behind the estimated parameters although they may have good in-sample performances.

A necessary condition to price China's stock market correctly is to build the model on economic theory foundation. Taking China's financial system's complexity into consideration, we cannot ignore government's essential role in influencing stock prices both directly and indirectly. A proper pricing model for China should be a combination of financial theory and econometric method. Therefore, we choose Xiong's model for China's financial system as our theory basis. In Xiong's model the government needs to trade against noise traders according to its conditional expectation. Without the government, the myopia of investors and the price insensitivity of noise traders jointly lead to market breakdown. We extend the model to answer two questions: How to estimate the intensity of government intervention? And how to find the proper empirical asset pricing model using semi-parametric econometrics method?

Finally, inspired by the idea that government's intervention can be a pricing factor we construct a generative adversarial network. A large amount of literature uses various machine learning methods to explore the relationship between excess return and covariates. These machine learning methods take advantage of big data and partially address 'Curse of Dimension'. Chen proposes a novel network based on Hansen's theory that estimating an SDF minimizing the largest pricing error is closest to an admissible true SDF in the least square distance. His work is a perfect combination of asset pricing theory and deep learning. In this paper we also use GAN model to find the SDF but we construct the moment condition by minimizing stocks' return volatility. Specifically, the moment condition in this network minimizes return's variance in a given period. The intuition behind this method is that when an asset's volatility is too large then it will be restricted by policies and government's actions. Therefore, adding such a procedure as an optimization condition can be seen as adding government's intervene into the pricing kernel and make the network work better.

II. LITERATURE REVIEW

Our paper contributes to an emerging literature that uses fancy non-linear model for asset pricing. Gu, Kelly and Xiu (2020) conduct a comparison of different machine learning methods for predicting the panel of US stock returns. Their estimates of the expected risk premia of stocks map into a cross-sectional asset pricing model. Bryzgalova, Pelger and Zhu (2020) include the no-arbitrage condition constraint into deep learning networks and obtain better results for asset pricing. Additionally, they identify the dynamic pattern in macroeconomic time series using Long Short Term Memory unit. Freyberger, Neuhierl and Weber (2020) use Lasso selection methods to estimate the risk premia of stock returns as a non-linear additive function of characteristics. Rossi (2018)) uses Boosted Regression Trees to form conditional mean-variance efficient portfolios based on market portfolio and risk-free asset. Gu, Kelly and Xiu (2019) use an auto-encoder neural network to extend traditional linear model. Bryzgalova, Pelger and Zhu (2020) use decision trees to build a cross-sectional of asset returns. Avramov, Cheng and Metzker (2020) propose that trade frictions may have a negative effect on the performance of machine learning algorithm. Cong, Tang, Wang, and Zhang (2020) propose reinforcement-learning-based portfolio management to directly optimize investors' objectives under trading friction constraints. However, there are not too much literature using such statistical learning methods for China's stock market. We compare different models' performances based on both firm characteristics and macroeconomic variables.

Based on Fama and French's unprecedented masterpiece, see Fama and French (1993) and Fama and French (2015), new statistical methods have been developed to study the cross-section of returns in the linear framework but accounting for large amount of conditional information. Lettau and Pelger (2020) extend PCA to account for no-arbitrage condition. They show that a no-arbitrage penalty term can overcome the low signal-to-noise ratio problem in financial data and find appropritiate pricing kernel. Kozak, Nagel and Santoshr (2020) estimate the SDF based on charateristic-sorted factors with a modified elastic net regression. Pelger (2020) apply PCA to stock returns projected on characteristics to obtain a conditional multi-factor model where the loadings are a linear combination of characteristics. Pelger (2020) applies PCA on high-frequency data to capture the time-variant factor risk. Pelger and Xiong (2019) show that macroeconomic variables are relevant to capture time variation in PCA-based factors. Bansal and Viswanathan (1993) and Chen and Ludvigson (2009) propose using a given set of conditional GMM equations to estimate the SDF with neural networks, but restrict themselves to a small number of conditional variables. We are firmly convinced that imposing theoretical economic structure on learning algorithm can substantially improve model's performance. A similar idea is also proposed by Lewis and Syrgkanis (2018) for non-parametric instrumental variable regressions. We also construct test portfolios based on economic theory and we think imposing theoretical structure on learning algorithm can significantly improve model's performance and make it more powerful to explain the reality.

Then it is important to find out which conditions should be added into the pricing model. China's stock market has its own speciality caused by government's active intervention in financial markets. Brunnermeier, Sockin and Xiong (2020) develop a theoretical framework in which interventions prevent a market breakdown and a volatility explosion caused by the reluctance of short-term investors to trade against noise traders. In this model intervention becomes an additional factor driving asset prices and worsens information efficiency of asset prices. Packer and Spiegel (2016) find a significant positive relationship between the number of IPOs and the market index return in China's stock market, which confirming China Securities Regulatory Commission(CSRC)'s policy to lean against the market cycle. In addition, on July 8, 2015, the CSRC imposed a lockup on shareholders with 5% or more of their companies. People's Bank of China(PBC), the central bank, also adopt policies to lean against the stock market cycle. During the 2015 stock market crash, PBC cut interest rates and reduced reserve ratios to boost the liquidity of financial system. Xing, Pan and Wangl (2018) provide an empirical overview of the Chinese capital market's historical development and main empirical characteristics. Based on Xiong's work, We propose an estimation procedure to test the role of government in maintaining financial system as well as a semi-parametric method to price China's A-share stocks.

We describe the dataset and basic models I use in Section 3. Section 4 develops the theoretical framework and econometric methods to test the hypothesis. Section 5 builds a generative adversarial network(GAN) with an objective to minimize price volatility and Section 6 concludes.

III. FRAMEWORK OF ASSET PRICING THEORY AND EMPIRICAL RESULTS

A. Stochastic Discount Factor

We start from basic concept of stochastic discount factor. A discount factor m is a random variable that generates prices from payoffs, $p = E(mx)$. m is function of observed data which means $m = f(D)$. Generally speaking, these two equations describe the whole process of asset pricing.

1. Payoff Space, Law of One Price and Existence of Discount Factor

The payoff space $\underline{X}$ is the set(or subset) of all the payoffs that investors can purchase. If there are complete contingent claims to S states of nature, then $\underline{X} = R^S$. The payoff space includes primitive assets, and it is often assumed that investors can also form new payoffs by forming any portfolio, which is shown in axiom 1. If investors can form portfolios of basic payoffs, then the payoff space consists of all linear combinations of original payoffs $\underline{X} = c'x$ where c is a vector of portfolios weights. Law of one price states that if two portfolios have the same payoffs in every state of nature, then they must have the same price. This law makes sure that investors cannot make instantaneous profits by repackaging portfolios, which is shown in axiom 2.

AXIOM 1: $x_1, x_2 \in \underline{X}$ implies $ax_1 + bx_2 \in \underline{X}$ for any real number a, b.

AXIOM 2: $p(ax_1 + bx_2) = ap(x_1) + bp(x_2)$.

THEOREM 3.1: Given free portfolio formation axiom 1 and the law of one price axiom 2, there exists a unique payoff $x^* \in \underline{X}$ such that $p(x) = E(x^*x)$ for all $x \in \underline{X}$.

x^* is a discount factor. This theorem means any linear function on a space $\underline{X}$ can be represented by inner product with a vector that lies in $\underline{X}$. There may be other discount factors not in $\underline{X}$. Unless the market is complete, there are an infinite number of random variables that satisfy $p = E(mx)$ then $p = E[(m+\epsilon)x]$ for any any ϵ orthogonal to x, $E(\epsilon x) = 0$. This construction generates all of the discount factors. Reversing the argument, x^* is the projection of any stochastic discount factor m on the space $\underline{X}$, this discount factor is known as the mimicking portfolio for m. Algebraically,

$$\begin{aligned} p &= E(mx) \\ &= E[(proj(m \mid \underline{X}) + \epsilon)x] \\ &= E[proj(m \mid \underline{X})x]. \end{aligned} \tag{1}$$

THEOREM 3.2: No arbitrage implies the existence of a strictly positive(positive in each state) discount factor, m$\geq 0, p = E(mx), \forall x \in \underline{X}$.

2. Factor Pricing Models

A traditional asset pricing model is consumption-based model. To be specific, consider the standard power utility function

$$u'(c) = c^{-\gamma}. \tag{2}$$

Then, excess returns should obey

$$0 = E_t(\beta(\frac{c_{t+1}}{c_t})^{-\gamma} R^e_{t+1}). \tag{3}$$

Given a value of γ, we can use data on consumption and returns to check whether actual expected returns are in accordance with the formula. However, it has a poor empirical performance. This motivates us to find alternative asset pricing models, which means searching for other functions for stochastic discount factor m. A widely used method is factor model that model marginal utility in terms of other variables directly. Factor model specifies taht the stochastic discount factor is a linear function of a set of proxies,

$$m_{t+1} = a + b_A f^A_{t+1} + b_B f^B_{t+1} + ... \quad (4)$$

The factors are selected as plausible proxies for marginal utility, for example, events that describe whether typical investors are happy or unhappy. A famous single-factor model is the Capital Asset Pricing Model

$$m_{t+1} = a + bR^W_{t+1}, \quad (5)$$

where R^W is the rate of return on market portfolio (such as value averaged NYSE portfolio). The international Capital Asset Pricing Model suggests macroeconomic variables such as GNP and inflation and variables that forecast macroeconomic variables or asset returns as factors. Term structure models such as the Cox-Ingersoll-Ross model specify that the discount factor is a function of a few term structure variables such as short rate of interest and a few interest rate spreads.

3. Arbitrage Pricing Theory

The bad performance of consumption-based model motivates us to tie the discount factor m to other data. Typically we use linear factor pricing models and they dominate discrete time empirical work. APT starts from a statistical characterization. There is a big common component to stock returns: when the market goes up, most individual stocks also go up. Beyond the market, groups of stocks move together. Finally, each stock's return has some completely idiosyncratic movement. This is a characterization of realized returns, outcomes or payoffs. The point of the APT is to start with this statistical characterization of outcomes, and derive something about expected returns or prices. The intuition behind the APT is that the completely idiosyncratic movements in asset returns should not carry any risk prices, since investors can diversify them by holding portfolios. Therefore, risk prices or expected returns on a security should be related to the security's covariance with the common components or "factors" only.

The APT models the tendency of asset payoffs(returns) to move together via a statistical factor decomposition

$$\begin{aligned} x^i &= \alpha_i + \sum_{j=1}^{M} \beta_{ij} f_j + \epsilon^i \\ &= \alpha_i + \beta_i' f + \epsilon^i, \end{aligned} \quad (6)$$

f_j are the factors and β_{ij} are factor loadings. Additionally, the factor model is equivalent to the expression of risk premium regression model, which means:

$$m_{t+1} = a + b' f_{t+1} \iff E(R_{t+1}) = \alpha + \beta' \lambda. \quad (7)$$

B. Mean-Variance Frontier and Beta Representations

1. Expected Return-Beta Representations

Much empirical work in finance is cast in terms of expected return - beta representations of linear factor pricing models, of the form

$$E(R_i) = \alpha + \beta_{i,a}\lambda_a + \beta_{i,b}\lambda_b + ..., i = 1, 2..., N. \quad (8)$$

The β terms are defined as the coefficients in a multiple regression of returns on factors,

$$R_t = \alpha + \beta_a f_{a,t} + \beta_b f_{b,t} + ... + \epsilon_t, t = 1, ..., T. \quad (9)$$

This is a time-series regression, since we run a regression across time for each security. Factors f mean proxies for marginal utility growth such as consumption growth or the return on market portfolio(CAPM). We run returns on contemporaneous factors because this regression is not about predicting returns from variables seen ahead of time. It aims to measure contemporaneous relations or risk exposure.

The point of beta model is to explain the variation in average returns across assets. $\beta_{i,a}$ is interpreted as the amount of exposure of asset i to factor a risks, and λ_a is interpreted as the price of such risk exposure. The betas cannot be asset-specific or firm-specific characteristics, such as the size of the firm or book to market ratio. It is true that expected returns are associated with or correlated with many such characteristics. Stocks of small companies or of companies with high book-to-market ratios do have higher average returns. But the correlation must be explained by some beta. The proper beta should drive our any characteristics in cross-sectional regressions. For example, expected returns are truly related to size, one could buy many small companies from a large holding company. It would pay low average returns to shareholders while earn a large average return on its holdings. The problem is that "large" holding company will still behave like a portfolio of small stocks. Thus, only if asset returns depend on how you behave rather than who you are - on betas rather than characteristics - can a market equilibrium survive such simple repackaging schemes.

If there is a risk free rate, then $R_f = \alpha$. We often examine factor pricing models using excess returns directly. Differencing between R_i and R_f we obtain

$$E(R^{ei}) = \beta_{i,a}\lambda_a + \beta_{i,b}\lambda_b + ..., i = 1, 2..., N. \quad (10)$$

Here, β_{ia} represents the regression coefficient of the excess return on the factors. Start with $1 = E(mR^i) = E(m)E(R^i) + Cov(m, R^i)$, thus

$$E(R^i) = \frac{1}{E(m)} - \frac{Cov(m, R^i)}{E(m)}$$
$$= \alpha + (\frac{Cov(m, R^i)}{Var(m)})(-\frac{Var(m)}{E(m)}), \tag{11}$$

where $\alpha = \frac{1}{E(m)}$. Define $\beta_{i,m} = (\frac{Cov(m,R^i)}{Var(m)})$ and $\lambda_m = (-\frac{Var(m)}{E(m)})$, we get a single beta representation,

$$E(R^i) = \alpha + \beta_{i,m}\lambda_m. \tag{12}$$

It is often useful to express a pricing model in a way that the factor is a payoff rather than a real factor such as consumption growth. It is even more useful if the reference payoff is a return because the factor risk premium is also the expected excess return.

2. *An Orthogonal Characterization of the Mean-Variance Frontier*

Define R^* as the return corresponding to the payoff x^* that can act as the discount factor. The price x^* is $p(x^*) = E(x^*x^*)$ and $R^* = \frac{x^*}{p(x^*)}$. Then define

$$R^{e*} = proj(1 \mid \underline{R^e}), \quad \underline{R^e} = \{x \in X \ s.t. \ p(x) = 0\}, \tag{13}$$

and we can get

$$E(R^e) = E(1 \times R^e)$$
$$= E[proj(1 \mid R^e) \times R^e] \tag{14}$$
$$= E(R^{e*}R^e),$$

THEOREM 3.3: Every return R^i can be expressed as $R^i = R^* + w^iR^{e*} + n^i$

where w^i is a number and n_i is an excess return with property $E(n^I) = 0$. The three components are orthogonal,

$$E(R^*R^{e*}) = E(R^*n^i)$$
$$= E(R^{e*}n^i) = 0. \tag{15}$$

This theorem quickly implies the characterization of the mean variance frontier,

THEOREM 3.4: R^{mv} is on the mean-variance frontier if and only if

$$R^{mv} = R^* + w^iR^{e*}. \tag{16}$$

A relation between the Sharpe ratio of an excess return and the volatility of discount factors is,

$$\frac{\sigma(m)}{E(m)} \geq \frac{\mid E(R^e) \mid}{\sigma(R^e)}. \tag{17}$$

Quckly,

$$0 = E(mR^e)$$
$$= E(m)E(R^e) + \rho\sigma(m)\sigma(R^e). \tag{18}$$

This implies a beautiful duality,

$$\min_{\{\text{all m that price } x \in \underline{X}\}} \frac{\sigma(m)}{E(m)}$$
$$= \max_{\{\text{all excess returns } R^e \ in \ \underline{X}\}} \frac{E(R^e)}{\sigma(R^e)}. \tag{19}$$

The return $R^* = \frac{x^*}{E(x^{*2})}$ can also serve as the factor in a beta pricing model.

THEOREM 3.5: $1 = E(mR^i)$ implies an expected return - beta model with $x^* = proj(m \mid \underline{X})$ as factors, e.g.

$$E(R^i) = \alpha + \beta_{i,x^*}\lambda_{x^*}$$
$$= \alpha + \beta_{i,R^*}[E(R^*) - \alpha]. \tag{20}$$

Suppose we have an expected return - beta model such as CAPM, APT, etc. An expected return - beta model is equivalent to a model for the discount factor that is a linear function of the factors in the beta model.This result gives the connection between the discount formulation and the factor model formulation common in empirical work.

THEOREM 3.6: Given the model

$$m = 1 + b'[f - E(f)]; \ E(mR^e) = 0. \tag{21}$$

one can find λ such that $E(R^e) = \beta'\lambda$, where β are the multiple regression coefficients of excess returns R^e on the factors.

3. *Conditioning Information*

If payoffs and discount factors were independent and identically distributed (i.i.d) over time, then conditional expectations would be the same as unconditional expectations and we would not have to worry about the distinction. But stock price or dividend ratios, bond and option prices all change over time. So the model should be written as

$$p_t = E_t[m_{t+1}x_{t+1} \mid I_t]. \tag{22}$$

One approach is to specify and estimate explicit statistical models of conditional distributions of asset payoffs and discount factor variables(e.g. consumption growth). But as we make the conditional mean, variance, covariance and other parameters of the distribution of N returns depend flexibly on M information variables, the number of required parameters can quickly exceed the number of observations.

Take unconditional expectations to obtain $E(p_t) = E(m_{t+1}x_{t+1})$. Suppose we multiply the payoff and price by an instrument z_t observed at time t. Then

$$z_t p_t = E_t(m_{t+1}x_{t+1}z_t). \tag{23}$$

Take unconditional expectation, we can get

$$E(p_t z_t) = E(m_{t+1}x_{t+1}z_t). \tag{24}$$

Group $(x_{t+1}z_t)$ and call it a payoff $x = x_{t+1}z_t$ with price $p = E(p_t z_t)$. So we can think of it as a price and a payoff, and apply all the asset pricing theory directly. $z_t x_{t+1}$ are the payoffs to managed portfolios. An investor who observes z_t invest in an asset according to the value of z_t. Practically every test uses managed portfolios. For example, the size, beta, industry, book-market ratio and so forth portfolios are all managed portfolios, since their composition changes every year in response to conditioning information.Checking the expected price of all managed portfolios is, in principle, sufficient to check all the implications of conditioning information.

THEOREM 3.7: If $E(z_t) = E(m_{t+1}R_{t+1}z_t), \forall z_t \in I_t$, then $E(m_{t+1}R_{t+1} \mid I_t) = 1$.

$E(m_{t+1}R_{t+1} \mid I_t)$ is equal to a regression forecast of $m_{t+1}R_{t+1}$ using every variable $z_t \in I_t$ which means every variable and every nonlinear measurable transformation of every variable. But there is a practical limit to the number of instruments z_t because only variables that forecast returns or m add new information.

C. Machine Learning Methods

1. Lasso and Elastic Net Regression

Lasso(Least Absolute Shrinkage and Selection Operator) is proposed by Robert Tibshirani. It can get a refined model by constructing a penalty function to shrinkage some coefficients. Mathematically,

$$\hat{b} = \underset{b}{argmin} \mid Y - \sum_{j=1}^{p} X_j b_j \mid^2, s.t. \sum_{j=1}^{p} \mid b_j \mid \leq t. \tag{25}$$

This expression is equivalent to

$$\hat{b} = \underset{b}{argmin} \mid Y - \sum_{j=1}^{p} X_j b_j \mid^2 + \lambda \sum_{j=1}^{p} \mid b_j \mid, \tag{26}$$

where λ and t are one-to-one correspondent tuning parameters. Through changing values of t we can shrinkage overall coefficients. Cross-validation method can be used for determination of t, which is proposed by Efron and Tibshirani in 1993.

Lasso's requirement for data is relatively low. No matter the data is discrete or continuous, Lasso can deal with it. Additionally, Lasso can decrease the complexity of selecting variables.

Elastic Net is a linear regression model using $L1$ and $L2$ as prior regularization matrices. This combination is used for sparse models with few non-zero weights, such as Lasso, but it can maintain Ridge's regularization property. We can use $L1$ ratio parameter to adjust the convex combination of L1 and L2 (a special kind of linear combination). Elastic Net is useful when multiple variates are related to another variate. Lasso prefers to choose one on them at random, while Elastic Net prefers to choose both. In practice, one advantage of the trade-off between Lasso and Ridge is that it allows Ridge's stability to be inherited in the under rotate process. The objective function of Elastic Net is:

$$\min_{w} \frac{1}{2n}||Xw - y||_2^2 + \lambda_1||w||_1 + \lambda_2||w||_2^2. \tag{27}$$

2. Boosted Regression Trees and Random Forest

Regression trees have become a popular machine learning approach for incorporating multi-way predictor interactions. Unlike linear models, trees are fully non-parametric and posses a logic that departs markedly from traditional regression. At a basic level, trees are designed to find groups of observations that behave similarly to each other. A tree grows in a sequence of steps. At each step, a new branch sorts the data leftover from the proceeding step into bins based on one of the predictor variables. This sequential branching slices the space of predictors into rectangular partitions and approximates the unknown function with the average value of the outcome variable within each partition. Formally, the prediction of a tree $\mathcal{T}$, with K leaves(terminal nodes), and depth L, can be written as

$$g(z_{i,t}; \theta, K, L) = \sum_{k=1}^{K} \theta_k \mathbf{1}_{\{z_{i,t} \in C_k(L)\}}, \tag{28}$$

where $C_k(L)$ is one of the K partitions of the data. Each partition is a product of up to L indicator functions of the predictors. The constant associated with partition

k is defined to be the sample average of the predictors. To grow a tree is to find the optimal bins that discriminate among the potential outcomes. The specific predictor variable upon which a branch is based and where the branch is split, is chosen to minimize the forecast error. The expanse of potential structures, however, precludes exact optimization. We follow Breiman algorithm whose basic idea is to myopically optimize forecast error at the start of each branch. At each new level, we choose a sorting variable from the set of predictors and split value to maximize the discrepancy among average outcomes in each bin. The loss associated with the forecast error for a branch C is often called "impurity", which describes how similarly observations behave on either side of the split. We choose the most popular l_2 impurity for each branch of the tree:

$$h(\theta, C) = \frac{1}{\mid C \mid} \sum_{z_{i,t} \in C} (r_{i,t+1} - \theta)^2, \tag{29}$$

where $\mid C \mid$ denotes the number of observations in set C. Given C, it is clear that the optimal choice of $\theta : \theta = \frac{1}{|C|} \sum_{z_{i,t} \in C} r_{i,t+1}$. The procedure is equivalent to finding the branch C that locally minimizes the impurity. An important advantage of a tree model is that it is invariant to monotonic transformations of predictors. Additionally, it can accommodate categorical and numerical data in the same model and approximate potentially severe non-linearities. A tree of depth L can capture $(L-1)$-way interactions.

Trees are among the prediction methods most prone to overfit, and therefore must be heavily regularized. The first regularization method is boosting, which recursively combines forecasts from many over-simplified trees. Shallow trees on their own are weak learners with minuscule predictive power. The theory behind boosting suggests that many weak learners may comprise a single strong learner with greater stability. The second regularization method is random forest which is an ensemble method combining forecasts from many different trees. It is a variation on a more general procedure known as bootstrap aggregation. Baseline tree bagging procedure draws B different bootstrap samples from the data, fits a separate regression tree to each, then averages their forecasts.

D. Empirical Results

We collect monthly individual stock returns data for all China A-share stocks. The sample period spans February 2000 to December 2019, totaling 20 years. The number of stocks in my sample is almost 3500, with average number of stocks per month around 2500. In addition, we build a large set of stock level predictive characteristics based on the cross section of stock return literature. We divide the data into 19 years of training sample and 1 year of test sample.Table I presents the comparison of machine learning techniques in terms of their out-of-sample predictive R^2.

Figure 1 presents the comparison of machine learning techniques in terms of their out-of-sample predictive R^2. We compare 14 models in total, including OLS with all covatiates, OLS with three factors(size, book-to-market and momentum as the only covaiates), PLS, PCR, elastic net, GLM, RF, GBRT and neural networks with one to five layers. The third value in each model is R^2 for the entire sample. Traditional statistical methods are likely to suffer from overfitting and have bad out-of-sample performance. A sparse parameterization, regularization or penalizing the specification generates a substantial improvement. Dimension reduction methods, such as PLS and PCR, also raise R^2 compared with OLS-3. The improvement of dimension reduction over variable selection via elastic net suggests that characteristics are partially redundant and fundamentally noisy signals. Combining them into low-dimension components can eliminate some noise and improve signal-to-noise ratio. Lasso fails to improvement the performance because it includes no interaction among different characteristics. This implies linear model with univariate expansions provide little increment although it selects more features compared with elastic net.

Nonparametric methods, such as boosted trees, random forest and neural networks, have relatively better performance. This fact shows interaction between different characteristics have powerful explanation for cross-section returns. Incorporating these complex interactions which are embedded in tree and neural network models but missed in other techniques is important. However, more layers in network is not necessary for better performance. NN5 is not as good as NN2, NN3 and NN4, which means in the monthly return setting, the benefits of deep learning is limited. Figure 2 shows sharpe ratio of different models and the pattern is very similar to performence of R^2.

Now we investigate the relative importance of individual covariates. For each model, we calculate the reduction of R^2 from setting all values of a given predictor to 0 within each training sample, and average these into a single importance measure for each predictor. Figure 3 shows the results. Variable importance is normalized to sum to 1 for convenience. Figure4 presents results of portfolio analysis.

Recall that an important feature of China's stock market is that it is full of noise traders. Consequently the government have to be active in intervening the stock market by trading against myopic investors. Government adopts necessary policies to maintain financial stability and these policies have both direct and indirect effects on stock returns. This may be one of the source that can improve model's performance.

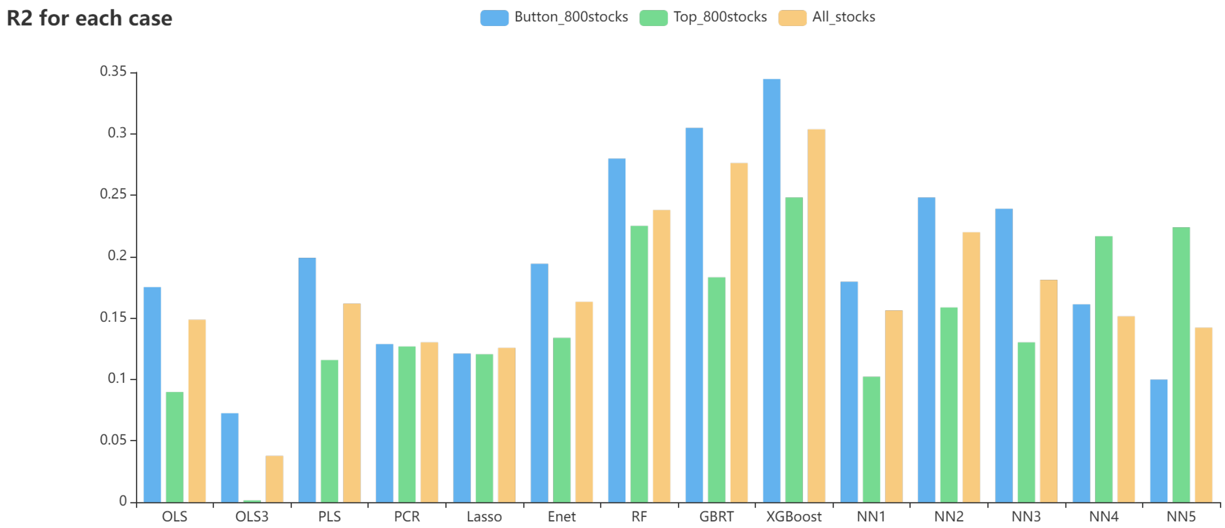

FIG. 1. R^2 of Different Machine Learning Models

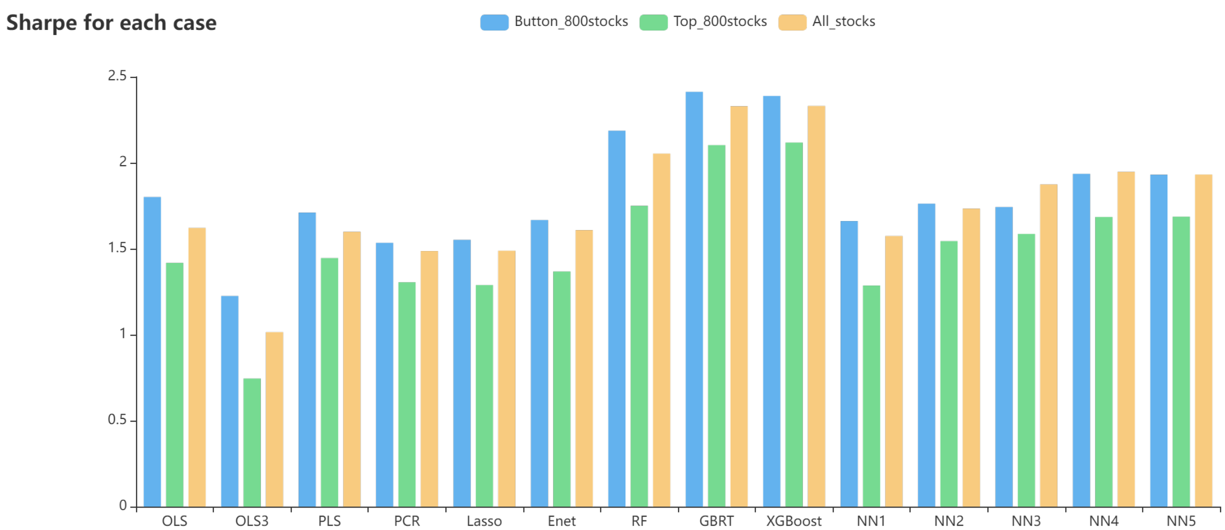

FIG. 2. Sharpe Ratio of Different Machine Learning Models

IV. GMM AND GENERATIVE ADVERSARIAL NETWORK

A. Interpreting GMM Procedure

$E(mR^e) = 0$ can be translated to a predicted expected return $E(R^e) = -\frac{Cov(m,R^e)}{E(m)}$ and we can write the pricing error as

$$g(\boldsymbol{\theta}) = \frac{1}{R_f}(E(R^e) - (-\frac{Cov(m,R^e)}{E(m)})), \qquad (30)$$

If we express the model in expected return-beta language $E(R^{ei}) = \alpha_i + \beta_i' \lambda$, then the GMM objective is proportional to the Jensen's alpha measure of mis-pricing $g(\boldsymbol{\theta}) = \frac{1}{R_f}\alpha$.

Ideally we should pick $\boldsymbol{\theta}$ to make every element of $g_T(\boldsymbol{\theta}) = 0$ and thus have the model price assets perfectly. However, there are usually more moment conditions(returns times instruments) than there are parameters. So we choose $\boldsymbol{\theta}$ to make $g_T(\boldsymbol{\theta})$ as small as possible., by minimizing a quadratic form,

$$\underset{\{\boldsymbol{\theta}\}}{min}\ g_T(\boldsymbol{\theta})' W g_T(\boldsymbol{\theta}),$$

W is a weighting matrix that directs GMM to emphasize some moments or linear combination of moments

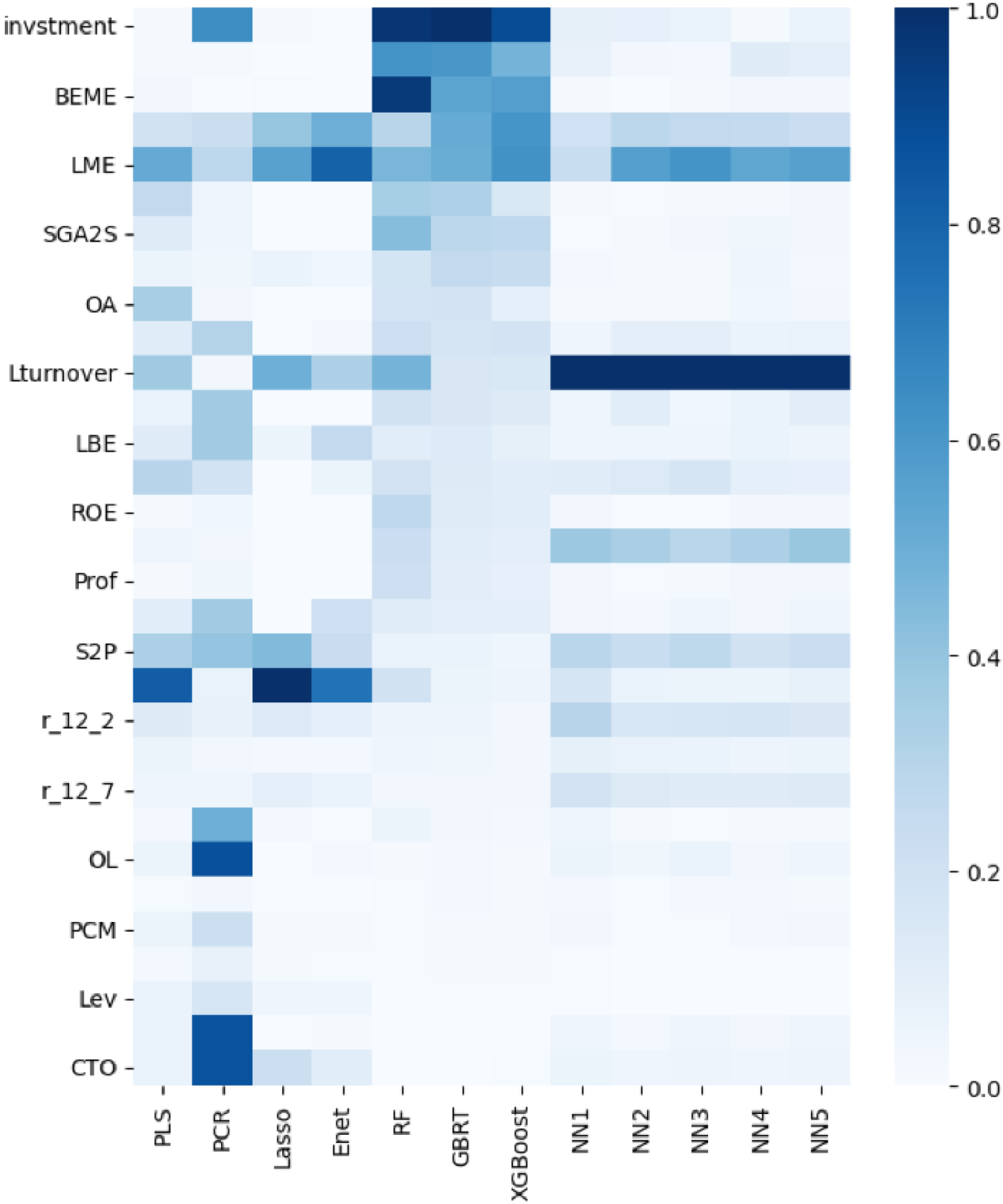

FIG. 3. Sharpe Ratio of Different Machine Learning Models

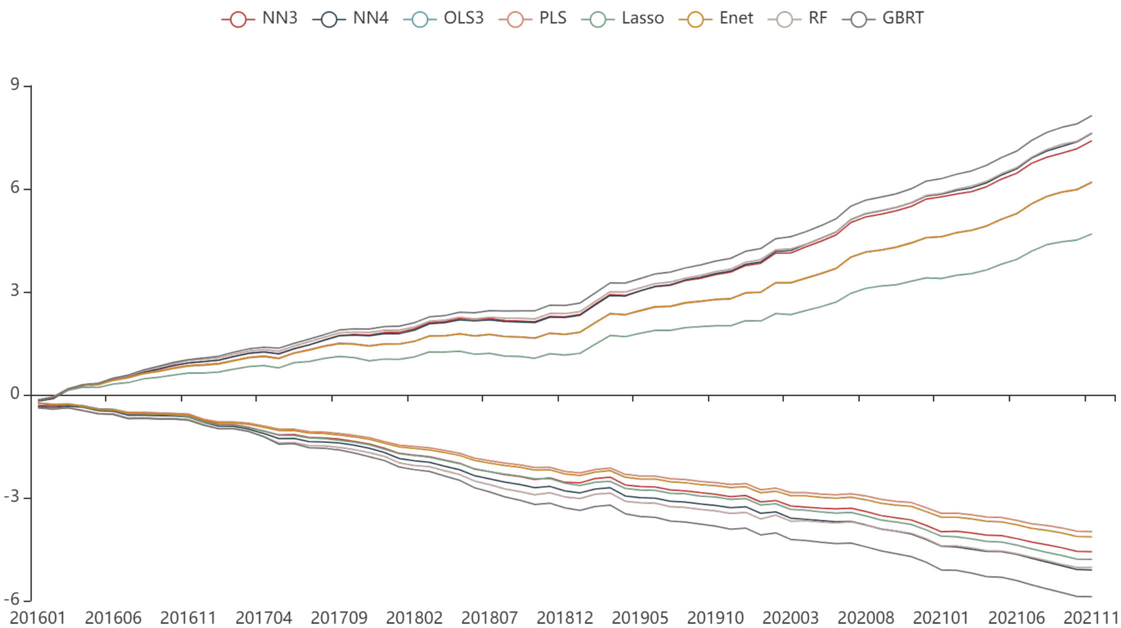

FIG. 4. Portfolio Analysis

TABLE I. Empirical Results of Different Models[a]

Model	OLS	OLS-3	PLS	PCR	ENet	RF	GBRT	NN1	NN2
Low	-1.53	-0.99	-1.54	-1.68	-1.57	-1.29	-1.46	-1.47	-1.51
2	-1.04	-0.84	-1.02	-0.94	-1.01	-0.92	-1.03	-1.00	-0.99
3	-0.82	-0.73	-0.78	-0.68	-0.80	-0.80	-0.87	-0.78	-0.83
4	-0.66	-0.64	-0.64	-0.54	-0.64	-0.70	-0.73	-0.64	-0.68
5	-0.52	-0.57	-0.50	-0.41	-0.48	-0.60	-0.57	-0.54	-0.61
6	-0.40	-0.47	-0.39	-0.30	-0.39	-0.48	-0.45	-0.43	-0.50
7	-0.23	-0.39	-0.22	-0.18	-0.22	-0.32	-0.30	-0.26	-0.34
8	-0.10	-0.22	-0.10	-0.05	-0.10	-0.12	-0.03	-0.11	-0.14
9	-0.12	-0.03	0.13	0.13	0.13	0.17	0.25	0.12	0.17
10	-0.69	0.55	0.65	0.46	0.64	0.84	0.93	0.66	0.80

[a] *This table shows empirical results of different machine learning models. OLS has a relatively poor performance because of curse of dimension. Dimension reduction methods can partly solve the problem. Neural network has a better performance because it allows for flexible nonlinear expression.*

at the expense of others. The second-stage estimate picks a weighting matrix based on statistical considerations. Some assets may have much more variance than other assets. For those assets, the sample mean $g_T = E_T(m_t R_t - 1)$ will be a much less accurate measurement of the population mean $E(mR-1)$, since the sample mean will vary more from sample to sample. Therefore, we should pay less attention to pricing errors from assets with high variance. Specifically, using the inverse of variances of $E_T(m_t R_t - 1)$ on the diagonal.

Add instruments z_t observed at time t to models and replace payoffs with returns, the moment conditions are quadratic form,

$$E[m_{t+1}(\boldsymbol{\theta})(R_{t+1} \otimes z_t) - (1 \otimes z_t)] = 0. \tag{31}$$

There are several problems about empirical asset pricing. Firstly, a large amount of literature consider SDF as a linear combination of some characteristics. However, this assumption might be too strong and seems to be mis-specified. We need a more flexible non-parametric method that capture the non-linear relationship. Secondly, we should find the correct test assets. Thirdly, SDF depends on variables that can forecast economic states. Exposure and compensation for risk should depend on the economic conditions.In this paper I estimate a general non-linear asset pricing model with deep neural networks for all China A-Share stocks based on a substantial set of macroeconomic and firm-specific information. Finding the SDF weights is equivalent to solving a method of moment problem. The conditional no-arbitrage moment condition implies infinitely many unconditional moment conditions

$$E[m_{t+1} R^e_{i,t+1} g(I_t, I_{i,t})]$$

for any function $g(\cdot) : R^p \times R^q \to R^D$, where $I_t \times I_{i,t} \in R^p \times R^q$ denotes all the variables in the information set at time t and D is the number of moment conditions. I use two kinds of adversarial approach to estimate an asset pricing model for individual stock returns that take advantage of both fundamental and macroeconomic information. The first one is to use no-arbitrage condition as criterion function for constructing test assets with the largest pricing error. The objective function can be written as

$$\underset{w}{min}\ \underset{g}{max}\frac{1}{N}\sum_{j=1}^{N}\|E[(1-\sum_{i=1}^{N}w(I_t,I_{i,t})R^e_{i,t+1})R^e_{j,t+1})g(I_t,I_{i,t})]\|^2.$$

The other model takes Chinese government's intervene into consideration and uses government's goal as optimization objective. Specifically, for each stock, I find the optimal moment condition to minimize the volatility of stock monthly returns. This moment condition describes government's actions in maintaining financial stability and reflects China A share stock market's 'Chinese characteristics'.

$$g^* = \underset{g}{argmin}\frac{1}{N}\sum_{i=1}^{N}\frac{1}{T}\sum_{t=1}^{T}[m_{t+1}R^e_{i,t+1}g(I_t,I_{i,t}) - \frac{1}{T}\sum_{t=1}^{T}m_{t+1}R^e_{i,t+1}g(I_t,I_{i,t})]^2. \tag{32}$$

B. Deep Neural Network

A feedforward network is a flexible non-parametric estimator for a general functional relationship $y = f(x)$. FFN not only estimates non-linear relationship between covariates and variable but also captures interaction effects between a large dimensional set of covariates. A simple one-layer neural network combines the original covariates linearly and applies a non-linear transformation. This non-linear transformation is based on an element-wise operating activation function, for example, rectified linear unit which is defined as

$$ReLU(x) = max(x_k, 0). \tag{33}$$

The output is simply a linear transformation of the output from the hidden layer.

$$\begin{aligned} x^{(1)} &= ReLU(W^{(0)T}x^{(0)} + w_0^{(0)}) \\ &= ReLU(w_0^{(0)} + \sum_{k=1}^{K^{(0)}} w_k^{(0)}x_k^{(0)}), \end{aligned} \tag{34}$$

$$y = W^{(1)T}x^{(1)} + w_0^{(1)}. \tag{35}$$

THEOREM 4.1: If $\phi(\cdot)$ is a non-constant, bounded monotone increasing continuous function, $L_D = [0,1]^d$. Then for any $f(x) \in C(L_D)$, there exists an integer m, a group of real numbers $v_i, b_i \in \mathbb{R}$ and a real vector $w_i \in \mathbb{R}^d$, such that we can define

$$F(\boldsymbol{x}) = \sum_{i=1}^{m} v_i\phi(\boldsymbol{w_i^T x} + b_i) \tag{36}$$

that satisfies for any $\varepsilon \geq 0$,

$$\| F(\boldsymbol{x}) - f(\boldsymbol{x}) \| \leq \varepsilon, \forall \boldsymbol{x} \in L_d. \tag{37}$$

This theorem guarantees that a feedforward network can approximate any bounded function with arbitrary accuracy.

A Recurrent Neural Network with Long-Short-Term-Memory estimates hidden macroeconomic state variables. Instead of passing macroeconomic variables I_t as covariates to the FFN directly, I extract their dynamic patterns with a specific RNN and only pass on a small number of hidden states capturing these dynamics. RNNs are a family of neural networks for processing sequences of data. They estimate non-linear time-series dependencies for vector-valued sequences in a recursive form. A standard RNN is model takes the current input variable x_t and previous hidden state h_{t-1}^{RNN} and performs a non-linear transformation to get the current state h_t^{RNN}.

$$\begin{aligned} h_t^{RNN} &= h^{RNN}(x_0, ...x_t) \\ &= \sigma(W_h h_{t-1}^{RNN} + W_x x_t + w_0), \end{aligned} \tag{38}$$

where σ is a non-linear activation function. Intuitively, a simple RNN combines two steps: First, it summarizes cross-sectional information by linearly combining a large vector x_t into a lower dimensional vector. Second, it is a non-linear generalization of an auto-regressive process where the lagged variables are transformations of the lagged observed variables. The LSTM model is designed to deal with lags of unknown and potentially long duration in the time series, which makes it well-suited to detect business cycles. Specifically, LSTM is composed if a cell and three regulators: an input gate, a forget gate and an output gate. Intuitively, the cell is responsible for keeping track of the dependencies between the elements in the input sequence. The input gate controls the extent to which a new value flows into the cell, the forget gate controls the extent to which a value remains in the cell and the output gate controls the extent to which the value in the cell is used to compute the output activation of the LSTM unit.

Take $x_t = I_t$ as the input sequence of macroeconomic information and the output is the state processes h_t. At each step, a new memory cell $\widetilde{c_t}$ is created with current input x_t and previous hidden state h_{t-1}

$$\widetilde{c_t} = tanh(W_h^{(c)}h_{t-1} + W_x^{(c)}x_t + w_0^{(c)}). \tag{39}$$

The input and forget gates control the memory cell, while the output gate controls the amount of information stored in the hidden state.

$$input_t = \sigma(W_h^{(i)} h_{t-1} + W_x^{(i)} x_t + w_0^{(i)}), \tag{40}$$

$$forget_t = \sigma(W_h^{(f)} h_{t-1} + W_x^{(f)} x_t + w_0^{(f)}), \tag{41}$$

$$output_t = \sigma(W_h^{(o)} h_{t-1} + W_x^{(o)} x_t + w_0^{(o)}). \tag{42}$$

Denoting the element-wise product by $\circ$, the final memory cell and hidden state are given by

$$c_t = forget_t \circ c_{t-1} + input_t \circ \widetilde{c_t},\ h_t = out_t \circ tanh(c_t). \tag{43}$$

We use the state processes h_T instead of the macroeconomic variables I_t as input to the SDF network.

Then we build two models inspired by generative adversarial network in which we choose conditioning function $g(\cdot)$ that leads to the largest pricing error and the smallest return volatility, respectively. There are three major steps to train the model. In the first step, we choose an SDF that minimizes unconditional loss. In the second step we maximize the loss and minimize the variance respectively by optimizing parameters in the conditional network given SDF obtained in step 1. In the third step, given the conditional network we update the SDF network to minimize the conditional loss. The two LSTMs that summarize macroeconomic information are based on the criteria function of the two networks, that h_t are the hidden states that can minimize the pricing errors while h_t^g generate the test assets with largest pricing errors and smallest volatility respectively.

More specifically, model 1 minimizes the maximum pricing error in the second step

$$\{\hat{w}, \hat{h_t}, \hat{g}, \hat{h_t^g}\} = \min_{w, h_t} \max_{g, h_t^g} L_1(w \mid \hat{g}, \hat{h_t^g}, h_t, I_{i,t}), \tag{44}$$

where

$$L_1 = \frac{1}{N} \sum_{j=1}^{N} \| E[(1 - \sum_{i=1}^{N} w(h_t, I_{i,t}) R_{i,t+1}^e) R_{j,t+1}^e g(h_t^g, I_{i,t})] \|^2. \tag{45}$$

Model 2 minimizes return volatility, which can be seen as a two-step optimization problem.

In step 1:

$$\{\hat{g}, \hat{h}_t^g\} = \min_{g, h_t^g} Var[M_{t+1} R_{t+1}^e g(I_t, I_{i,t})], \tag{46}$$

In step 2:

$$\{\hat{w}, \hat{h}_t\} = \min_{w, h_t} \frac{1}{N} \sum_{j=1}^{N} E[(1 - \boldsymbol{w}(h_t, I_{i,t})^T \boldsymbol{R}_{t+1}^e) R_{j,t+1}^e \hat{g}(\hat{h}_t^g, I_{i,t})] \|^2. \tag{47}$$

Figure 1 gives a detailed description of the network. At first I put macroeconomics variables into a RNN network and use LSTM model to estimates the economic states. Then adding firm-specific characteristics to the FFN network to get a candidate portfolio weights. RNN and FFN are also used to find optimal moment conditions. With these conditions and returns I can calculate the loss function. These two networks compete with each other until convergence, that is neither the SDF nor the test assets can be improved.

Due to the high dimensionality and non-linearity of the problem, training a deep neural network is a complex task. I prevent the model from overfitting and deal with the large number of parameters by using ”Dropout”, which is a form of regularization that has generally better performance than conventional regularization. I optimize the objective function accurately and efficiently by employing an adaptive learning rate for a gradient-based optimization.

C. Data and Training Process

The data comes from RiceQuant database, which is supported by financial engineering laboratory of school of economics, Peking University. We use 34 firm-level characteristics of all China’s A-share stocks from December 2000 to November 2021. For macroeconomic variables, we use monthly data of 119 macroeconomic variables from December 2000 to November 2021. Details of firm characteristics and macroeconomic variables are in appendix A and appendix B, respectively. Missing values are filled with average value adjusted by stride. If data is missing in the last period, then it is filled with the last true value. There are total 252 months in our sample. Training set is the first 97 months, validation set is the next 60 months and test set is the last 95 months. We keep stocks that appear in all of the three sets and the number of stocks in total is 1040. Finally We have a panel data of 224992 observations.

As for training process, firstly we train an unconditional network. For the asset pricing equation $E(M_{t+1} R_{t+1}^e g(I_t, I_{i,t})) = 0$, set $g(I_t, I_{i,t}) = 1$. Since $M_{t+1} = 1 - w(I_t, I_{i,t})^T R_{t+1}^e$, where I_t and $I_{i,t}$ denotes macroeconomic variables and firm characteristics, respectively. Then the input for the network are R_{t+1}^e, I_t and $I_{i,t}$ and the loss function for the pre-trained unconditional network for SDF can be written as:

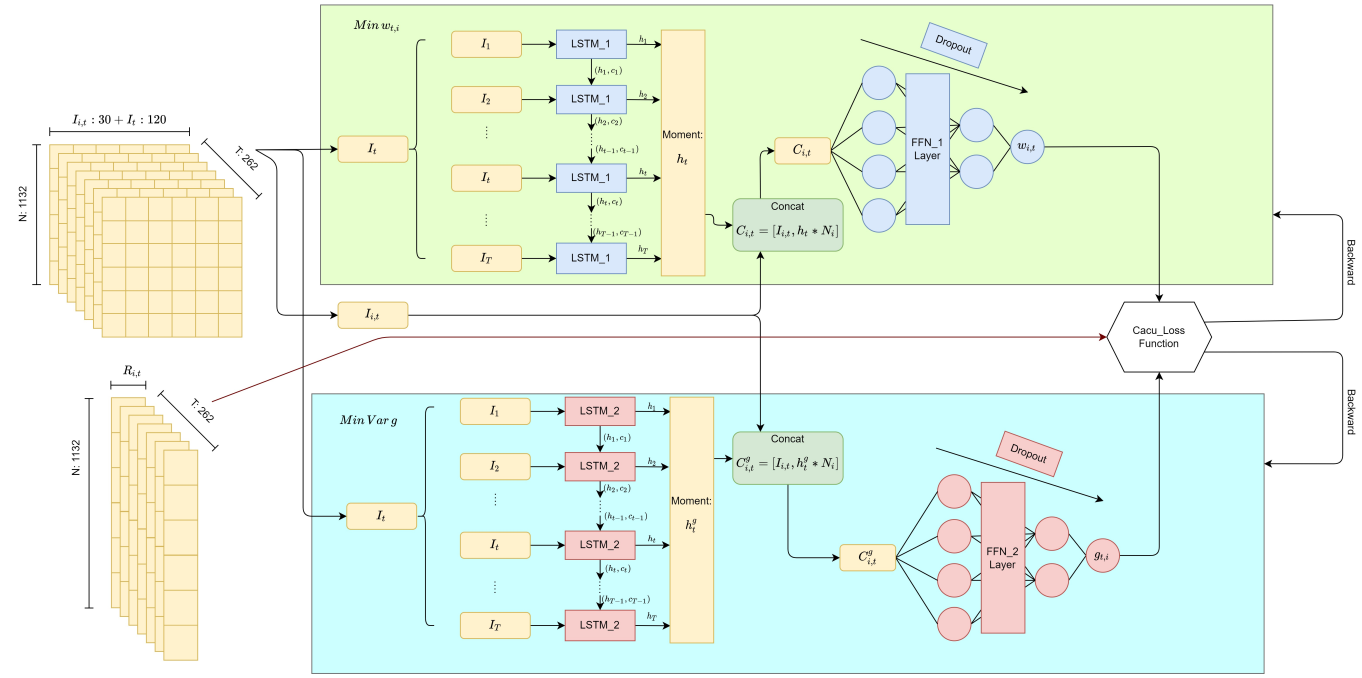

FIG. 5. Generative Adversarial Network Structure

$$\sum_{t=1}^{T}(\frac{1}{N}\sum_{i=1}^{N}(1-w(I_t,I_{i,t+1})^T R^e_{t+1})R^e_{i,t+1})^2.$$

Now we obtain an unconditional SDF network. The next step is to train the moment condition $g(I_t, I_{i,t})$. The network for training $g(I_t, I_{i,t})$ contains an LSTM and an FFN. The LSTM network's input is macroeconomic variable whose dimension is 252 × 119(number of months times number of macroeconomic variables). The output's dimension is 252 × 16. We use LSTM network to extract features and capture the business cycles. For each period t, we have individual stock's information of 34 characteristics. Then we splice 34 characteristics and 16 features obtained by the LSTM network and get a new dataset of 50(34+16) dimension. This 50-dimension data is the input for the FFN network and the output is an 8-dimension moment condition $g(I_t, I_{i,t})$. For all stocks in all periods, dimension of $E(M_{t+1}R^e_{t+1}g(I_t, I_{i,t}))$ is 1040 × 252 × 8.

Then for each of the 8 moment conditions $g(I_t, I_{i,t})$ we can calculate the variance of individual stock i:

$$\begin{aligned}&Var_{i,g}(M_{t+1}R^e_{i,t+1}g(I_t,I_{i,t})))\\=&\frac{1}{T}\sum_{t=1}^{T}(M_{t+1}R^e_{i,t+1}g(I_t,I_{i,t})-E(M_{t+1}R^e_{i,t+1}g(I_t,I_{i,t})))^2.\end{aligned} \tag{48}$$

Next, we calculate the average of Var_g and use it as our loss function to obtain the conditioning function $g(I_t, I_{i,t})$:

$$E(Var_{i,g})=\sum_{i=1}^{N}\sum_{g=1}^{8}Var_{i,g}(M_{t+1}R^e_{i,t+1}g(I_t,I_{i,t})). \tag{49}$$

Using this loss function we can obtain $g(I_t, I_{i,t})$, which is the moment condition that minimizes variance of the pricing error. We choose such a moment condition because it can mimic government's behaviour to maintain financial stability in China. Above is what is done in the first step. In the second step, we obtain optimal weight $w(I_t, I_{i,t})$ given the moment condition $g(I_t, I_{i,t})$. It is worth pointing out that we take $g(I_t, I_{i,t})$ as a known function. In this step we want to minimize the pricing error, so the loss function is:

$$\frac{1}{N}\sum_{j=1}^{N}\|E[(1-\boldsymbol{w}(h_t,I_{i,t})^T\boldsymbol{R}^e_{t+1})R^e_{j,t+1}\hat{g}(\hat{h}^g_t,I_{i,t})]\|^2.$$

Then we iterate step 1 and step 2 until they finally convergent. Thus we get the optimal weight $w(I_t, I_{i,t})$ and moment condition $g(I_t, I_{i,t})$. Since M_{t+1} can be seen as tangency portfolio, we calculate the Sharpe Ratio using $\frac{E(M_{t+1})}{\sqrt{Var(M_{t+1})}}$.

D. Empirical Results

Table II shows empirical results of the two models, using maximum pricing error and minimum return volatility as optimization condition respectively. Compared

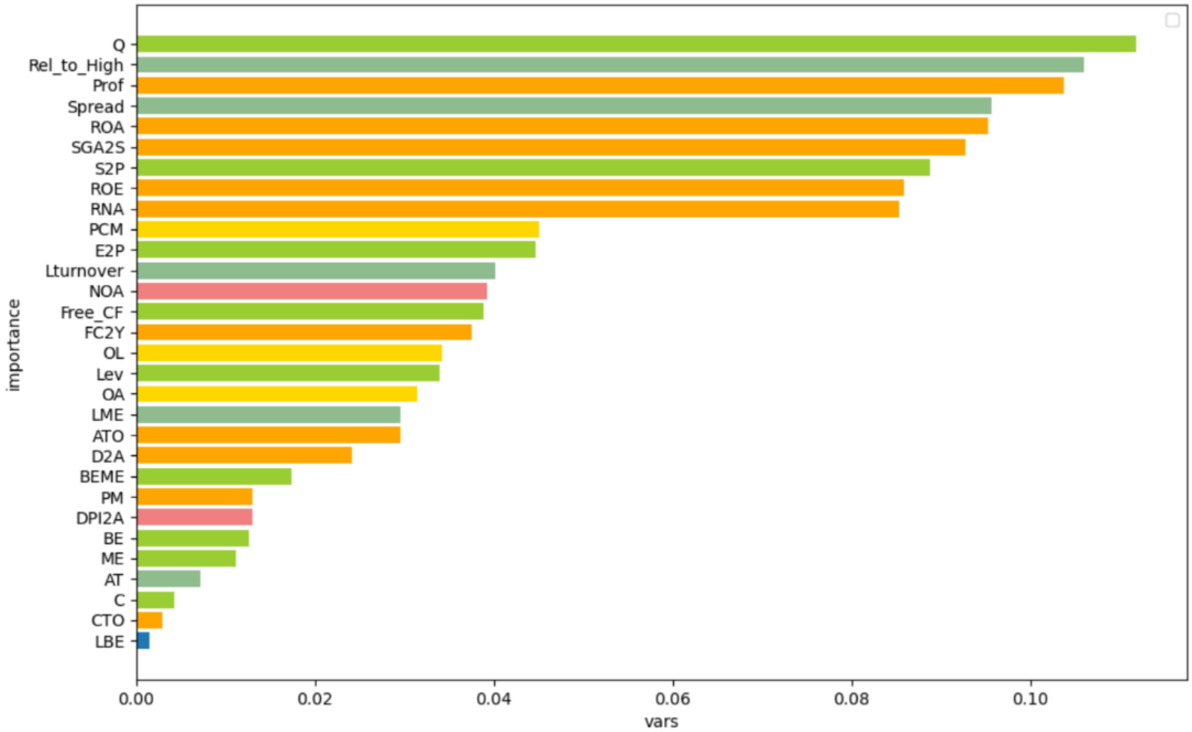

FIG. 6. Feature Importance in Model 1

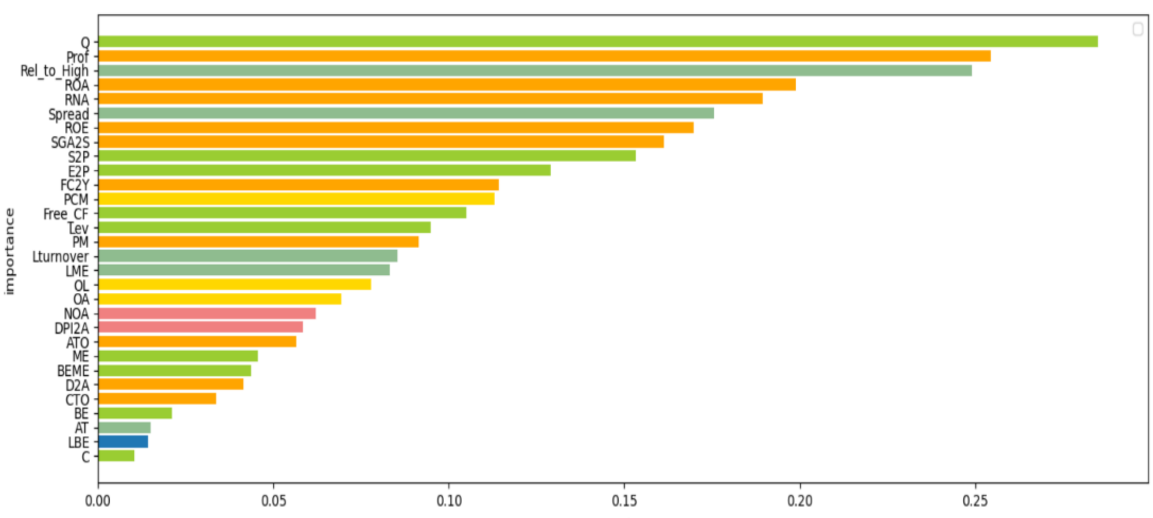

FIG. 7. Feature Importance in Model 2

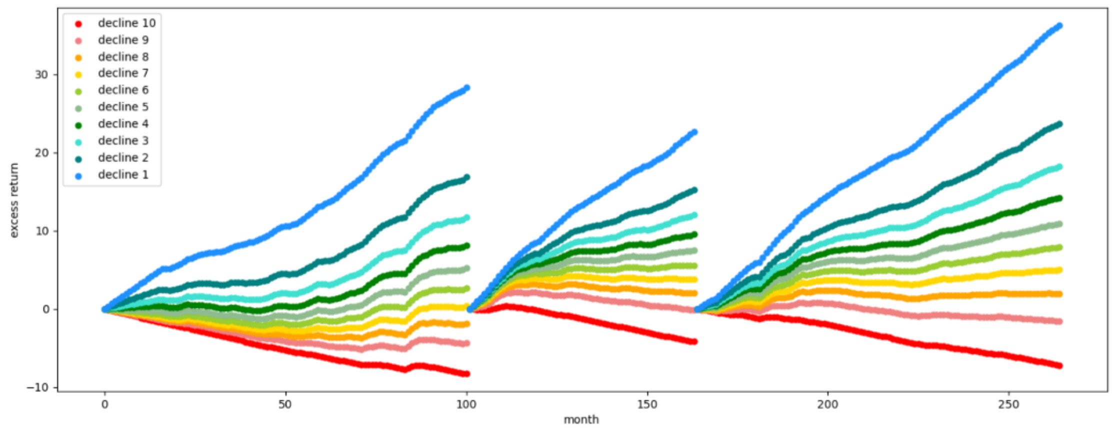

FIG. 8. Predictive Accumulative Excess Return in Model 1

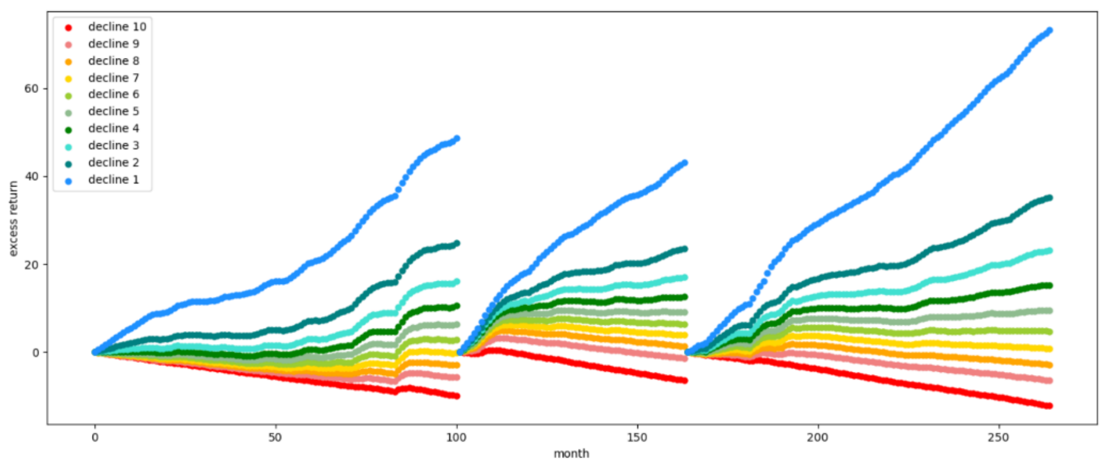

FIG. 9. Predictive Accumulative Excess Return in Model 2

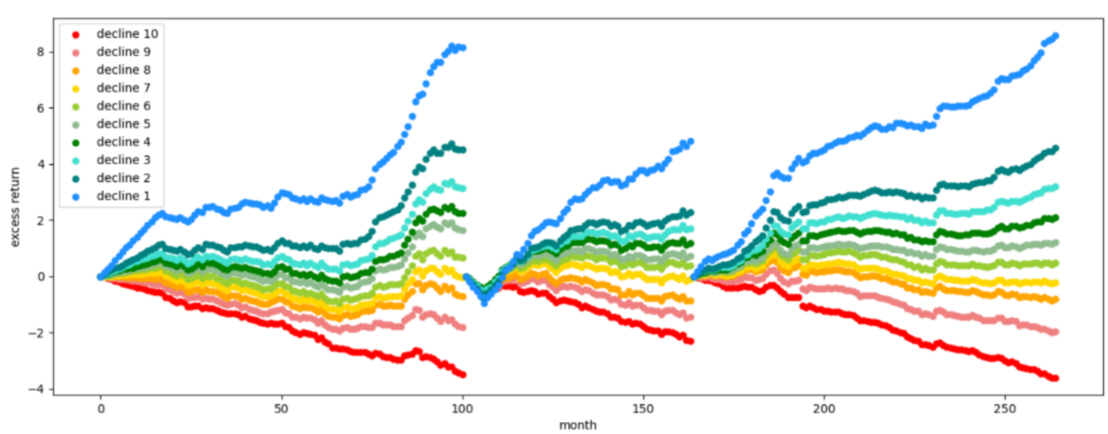

FIG. 10. Real Accumulative Excess Return in Model 1

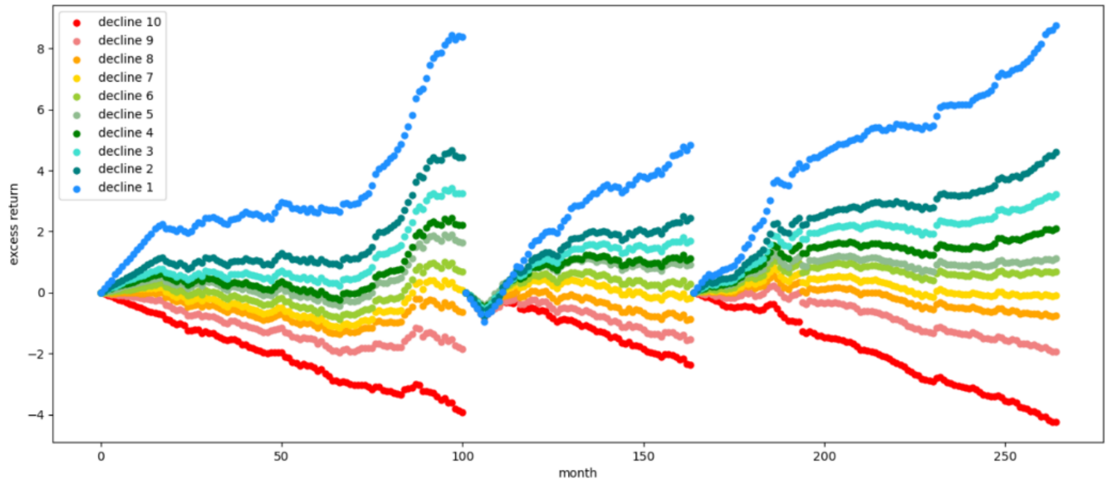

FIG. 11. Real Accumulative Excess Return in Model 2

TABLE II. Generative Adversarial Network with Different Optimization Conditions[a].

	Model 1			Model 2		
	Train	Valid	Test	Train	Valid	Test
R-Squared	14.60%	0.54%	24.79%	77.70%	6.12%	28.33%
Explained Variation	32.85%	27.20%	23.53%	45.28%	31.73%	30.02%
Sharpe Ratio	0.767	0.438	0.490	1.13	1.53	1.22

[a] *Model 1 is trained with optimization condition that minimizes the maximum pricing error. Model 2 is trained with optimization condition that minimizes the pricing error generated by the moment condition that minimizes return volatility.*

with traditional machine learning methods, generative adversarial network has a better performance. Out-of-sample R^2 can reach 24.79% and 28.33%, which are far more than the results in table I.

More importantly, compared model 1 with model 2, we can find that a GAN model with moment condition that minimizes return volatility has higher R^2 and Sharpe Ratio. This means the model has better ability for pricing China's stock market when taking government's intervene into consideration.

We rank the importance of firm-specific for the pricing kernel based on the sensitivity of the SDF weight with respect to these variables. The sensitivity is defined as:

$$S(x_j) = \frac{1}{C}\sum_{i=1}^{N}\sum_{t=1}^{T} \mid \frac{\partial w(I_t, I_{i,t})}{\partial x_j} \mid, \tag{50}$$

figure rank the variable importance of characteristics in the two models. The sum of all sensitivities is normalized to one. Results in tow models are similar, which show the robustness of our pricing strategy.

So far we have analyzed predictability of individual returns. Then we compare forecasting performance of machine learning methods for aggregate portfolio returns. There are a number of benefits to make the analysis on portfolio level. First, portfolio forecasts provide an additional indirect evaluation of my model and its robustness. Second, aggregate portfolios are more interesting because they represent the risky-asset savings vehicles most commonly held by investors. Third, the distribution of portfolio returns is sensitive to dependence among stocks, which implies that a good stock-level prediction is not guaranteed to produce accurate portfolio-level forecasts. Bottom-up portfolio forecasts allow us to evaluate a model's ability to transport its asset predictions. Finally, the portfolio results can be seen as another out-of-sample performance since the optimization procedure doer not directly account for portfolio's performance.

We build bottom-up forecasts by aggregating individual stock return predictions into portfolios. Given the weight of stock i in portfolio p, denoted $w_{i,t}^p$ and given a model-based out-of-sample forecast for stock i, denoted $r_{i,\hat{t+1}}$, I construct the portfolio return forecast as

$$r_{\hat{t+1}}^{p} = \sum_{i=1}^{n} w_{i,t}^{p} \times r_{i,\hat{t+1}}. \tag{51}$$

We divide all stocks into ten percentiles according to their predictive monthly returns and construct portfolios. Then We calculate the cumulative excess return of each portfolio. It is clear that the highest and lowest deciles separate in both models. Then we plot the real cumulative excess returns of this investment strategy compare them with the predicted results. We find the prediction is consistent with the reality, which implies both models have good performance.

V. CONCLUSION

This paper uses machine learning and deep learning methods to price China's A-share stock market. Traditional methods including OLS, Lasso and Random Forest, have poor performance in predicting the excess returns and suffer from over-fitting. This phenomenon motivates me to find the correct pricing kernel for the stochastic discount factor. Taking China's stock market's speciality into consideration, we need a pricing model that can describe 'Chinese Characteristics', which means that the market is full of myopic investors and noise traders. We construct a generative adversarial network and use minimum volatility as moment conditions. This optimization condition can reflect Chinese government's active intervene in capital market to maintain financial stability. Compared with traditional models and the GAN model that minimizes the largest pricing error, our model has higher R^2 and can obtain higher Sharpe Ratio. This means a correct pricing model for Chinese stock market should include government's behaviour that has effects on stock returns both directly and indirectly. Additionally, We construct deciled portfolios and predict the long-term excess returns and I find the predictive results are consistent with the reality.

Gu, S., B. T. Kelly, and D. Xiu (2020): "Empirical Asset Pricing via Machine Learning," *Review of Financial Studies*, 33(5), 2223–2273.

Bryzgalova, S., M.Pelger, and J.Zhu (2020): "Forest through the Trees: Building Cross-Sections if Stock Returns," *Working Paper.*

Freyberger, J., A.Neuhierl, and M.Weber(2020): *Dissecting Characteristics Noparametrically. Cambridge, U.K.: Cambridge University Press.*

Freyberger, J., A.Neuhierl, and M.Weber(2020): "Dissecting Characteristics Noparametrically," *Review of Financial Studies*, forth coming, 33(5), 2326–2377.

Rossi, A. G. (2018): "Predicting Stock Market Returns with Machine Learning," *Working paper.*

Gu, S., B. Kelly, and D. Xiu(2019): "Autoencoder Asset Pricing Models," *Journal of Econometrics*, forth coming.

Avramov, D., S. Cheng, and L.Metzker(2020): "Machine Learning versus Economic Restrictions: Evidence form Stock Return Predictability," *Working Paper.*

Cong, L. W., K. Tang, J. Wang, and Zhang(2020): "AlphaPortfolio for Investment and Economically Interpretable AI," *Working Paper.*

Fama, E. F., and K. R. French(1993): "Common Risk Factors in the Returns on Stocks and Bonds," *Journal of Financial Economics*, 33(1), 3-56.

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

Lattau, M.and M. Pelger(2020): "Factors that Fit the Time-Series and Cross-Section of Stock Returns," *Review of Financial Studies*, 33(5), 2274-2325.

Kozak S., S. Nagel, and S. Santosh(2020): "Shrinking the Cross Section," *Journal of Financial Economics*, 135(2), 271-292.

Kelly, B., S. Pruitt, and Y. Su(2019): "Characteristics are Covariances: A Unified Model of Risk and Return," *Journal of Financial Economics*, 134(3), 501-524.

Pelger, M.(2020): "Understanding Systematic Risk:A High-Frequency Approach," *Working Paper.*

Pelger, M., and R. Xiong(2019): "Interpretable Sparse Proximate Factors for Large Dimensions," *Working Paper.*

Bansal, R., and S. Viswanathan(1993): "No-Arbitrage and Arbitrage Pricing: A New Approach," *The Journal of Finance*, 48(5), 1719-1747.

Chen X., and S. Ludvigson(2009): "Land of Addicts? An Empirical Investigation of Habit-Based Asset Pricing Models," *Journal of Applied Econometrics*, 24, 1057-1093.

Lewis, G., and V. Syrgkanis(2018): "Adversarial Generalized Method of Moments," *Working Paper.*

Markus K. B., Michael S. , and Xiong W.(2018): "China's Model of Managing the Financial System," *Working Paper.*

Packer, F., and M, Spiegel(2016): "China's IPO Activity and equity market volatility," *FRBSF Economic Letter*, 18, 1-5.

Xing H., Pan J., and Wang J.(2016): "Chinese Capital Markt: An Empirical Overview," *Working Paper*, MIT Sloan.

Appendix A: Firm-Specific Characteristics

- **Total Assets(AT)** This variable measures the size of a firm.
- **Book-to-Market Ratio(BEME)** Firms with higher book-to-market ratio are likely to have higher returns. Usually they have some troubles and investors feel pessimistic about their perspective.
- **Asset Turnover(ATO)** This variable reflects efficiency of the use of assets.
- **Beta(β)** Beta measures the amount of risk. Under the setting of CAPM, I use time-series regression of each stock to calculate value of Beta.
- **Current-to-Asset Ratio(C)** This variable measures liquidity.
- **Capital Density(D2A)** This variable is the sum of depreciation and amortization per period

over total assets. It is a measure for cost. A lower D2A means higher returns.

- **Unused Rate(UR)** This variable is the sum of the difference of fixed assets and the difference of inventory over total assets. It measures the extent to which facilities are unused.
- **Price-to-Earnings Ratio(P2E)** This variable measures whether the price of a stock is overestimated. If the ratio is high then there is a large possibility that there exists a bubble.
- **Fixed Cost-Revenue Ratio** This variable measures a firm's ability for profit.
- **Cash Flow-Face Value Ratio(CF)** This variable measure a firm's ability to bring returns to shareholders.
- **Idiosyncratic Rate** Idiosyncratic is defined as the residual of regression.
- **Inventory** Investment is the growth rate of capital.
- **Leverage Ratio** Leverage is total liabilities over total assets. It is an index measuring a firm's ability to pay for debts.
- **Lag Market Equity** This variable is a measure for history information.
- **Turnover** This measure stock market activity.
- **Net Operating Assets** This variable reflects the fraction of assets using for main business.
- **Growth of Operating Assets** This measures potential for profit.
- **Operating Leverage** Its definition is the cost of main business over total assets. It is a fraction contrast to NOA.
- **Marginal Cost of Price** Its definition is operating revenue minus cost over operating revenue.
- **Marginal Price** Its definition is operating revenue minus depreciation over operating revenue.
- **Profit Rate** A firm's total profit over book value. This variable measures a firm's ability of earning profits for its shareholders.
- **Return of Total Assets(ROA)** Its definition is revenue over last year's total assets.
- **Momentum** Its definition is cumulative returns from last 12 months to last 2 months.
- **Long-Term Turnover** Its definition is cumulative returns from last 36 months to last 12 months.

Appendix B: Macroeconomics Variables

- **Purchase Price Index of Industrial Products** This variable measure the relative cost of industry.
- **Producer Price Index(PPI)** This index reflesscts the potential trend of price change in a certain area.
- **Consumer Price Index(CPI)** This index reflects price of common goods and inflation.
- **Capital Price Index** This variable measures the price of means of production.
- **Growth Rate of Investment for Real Estate Development** This variable reflects prosperity of real estate market. Real estate market is highly correlated to economic cycle and it is seen as an indicator for macroeconomic.
- **Loan from Financial Institution** This variable reflects information of financial market.
- **Short-Term Loan from Financial Institution** This variable indicates economic status.