AlphaManager: A Data-Driven-Robust-Control Approach to Corporate Finance*

Murillo Campello† Lin William Cong‡ Luofeng Zhou§

First draft: Feb 2023; this draft: June 2024.

Abstract

Corporate decision-making involves high-dimensional, non-linear stochastic control under managerial learning and dynamic interactions with the economic environment. We introduce an AI-assisted, data-driven-robust-control (DDRC) framework to complement theory, reduced-form models, and structural estimations in corporate finance research. We do so with an emphasis on explaining and predicting firm outcomes empirically, and offering policy recommendations for flexible business objectives. Specifically, we build a predictive environment module through supervised deep learning and add a policy module through deep reinforcement learning that goes beyond hypothesis testing on historical data or simulations. By incorporating model ambiguity and robust control techniques, our framework not only better explains and predicts corporate outcomes in- and out-of-sample, but also identifies important managerial decisions while offering effective policy recommendations adaptive to market evolution and feedback. We document rich heterogeneity in model prediction performance, ambiguity, and policy efficacy in the cross section of U.S. public firms and across time regimes. AlphaManager recommendations improve managerial performance by over 10% historically and have implications on management horizons. Critically, our DDRC approach informs where theory and causal analysis in corporate finance research should focus, and admits the incorporation of fragmented knowledge through ambiguity-guided transfer learning.

Keywords: AI/ML, Ambiguity, Big Data, Deep Learning, Managerial Decision, Offline Reinforcement Learning, Stochastic Control, Transfer Learning.

*We thank Wei Jiang and Tom Sargent for detailed feedback. We also thank Serdar Dinc, Jianqing Fan, Zhiguo He, Markus Pelger, Lea Stern, conference and seminar participants at AI and Big Data in Finance Research (ABFR) Webinar, the 10th Annual Workshop of Business Financing and Banking Research Group(USyd), China Meeting of Econometric Society (CMES 2021, Shanghai), Columbia Business School, 2024 EFMA Annual Meeting, 2023 GSU-RFS FinTech Conference, Harvest Fund 25th Anniversary Ceremony, INFORMS Annual Conference, 4th International FinTech Research Forum (RUC), Johns Hopkins University, NYU Stern, Peking University Guanghua, Rutgers Business School, 6th Shanghai Financial Forefront Symposium, Tsinghua University SEM, UBC Sauder, 4th Workshop on Big Data Econometric Theory and Application, and 2023 XJTLU AI and Big Data in Accounting and Finance Conference for helpful comments and discussions. We are grateful to Matthew Wong, Logan Kraver, and Atharwa Pandey for exceptional research assistance. Send correspondence to Cong at will.cong@cornell.edu.

†University of Florida Warrington College of Business and NBER.

‡Cornell University SC Johnson College of Business (Johnson), IC3, and NBER.

§New York University Stern School of Business

1 Introduction

Corporate finance research attempts to study firm decision-making and outcomes using theory and archival data. The literature is rooted in tractable representations of the corporate environment where rational agents optimize their utilities and interact with each other. However, researchers recognize the low predictability and explainability of corporate outcomes, and highlight a lack of systematic understanding of corporate decision-making (see, e.g., Graham, 2022, Presidential Address of the American Finance Association). Many corporate finance theory models are static, partial-equilibrium, or absent firm interactions with the environment or other firms, while empirical papers report evidence taken to be "consistent with" *ex-ante* theories without a realistic alternative to refute (Spiegel, 2023). Meanwhile, the real world managerial actions and firm outcomes are recorded with great granularity and timeliness in the digital era. Our paper shows that from a data-driven perspective, the abundance of data, algorithm advances, and powerful computation can inform scholars and business executives of novel empirical patterns and economic insights, and furthermore guiding them in the practice and research of corporate finance.

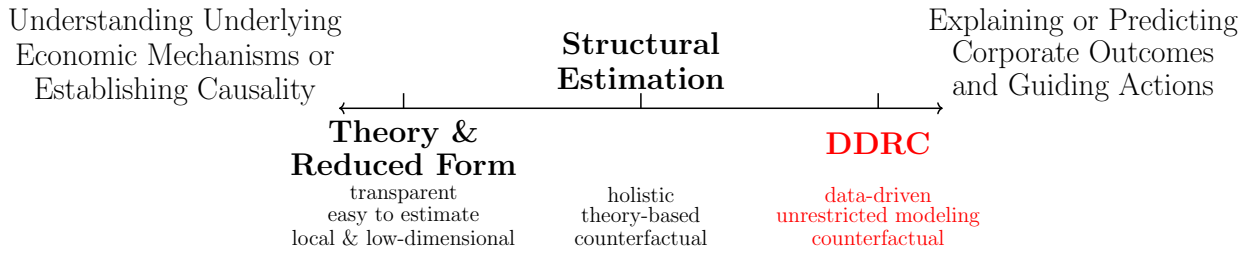
A data-driven approach to uncovering empirical patterns and advising corporate decision-making is challenging for several reasons. First, managers customarily consider a combination of complex decisions in a high-dimensional action space, often conditional on a non-trivial number state variables. Their actions interact and yield highly non-linear effects. All these call for approaches that go beyond conventional econometric models and low-dimensional causal inferences. Second, managerial decisions often interact with the economic environment and market reactions feed back into agents' decision-making (e.g., Bond et al., 2010, 2012; Edmans et al., 2012, 2015). Such interactions not only incur significant costs but are also difficult to incorporate into empirical studies.[1] Complicating matters, optimal corporate decisions are "unlabeled" in historical data, rendering supervised learning to be of limited use. Third, unlike physical laws or genetics, financial markets and systems are fast evolving, rendering concerns of potential data distributional shifts to be relevant and pressing.[2]

We argue that corporate finance and managerial decision-making can be viewed as robust

[1] More generally in social science research, real-time online interaction to generate new data is impractical, either because unfiltered, continuous data collection is expensive (e.g., in high-frequency trading), unethical (e.g., hiring and firing employees), or possibly dangerous (e.g., law enforcement). Even in domains where online interaction is feasible, we might still want to utilize previously collected data instead — for example, if the domain is complex and effective generalization requires large datasets. Therefore, as explained later, we pursue an offline reinforcement learning (RL) approach using historical data similar to Cong et al. (2020).

[2] Data shifts occur when joint distributions of inputs and outcomes differ across training and test samples.

Figure 1: Contrasting Corporate Finance Research Approaches



control problems subject to these particular challenges; namely, high-dimensionality, non-linearity, market feedback, and complexity that requires dynamic learning of the evolving environment, with potential shifts in the data-generating process. While extant models are useful, they are often inadequate for practical policy recommendations to corporate executives (Figure 1): To start, reduced-form theory and associated empirical models can help us understand the economic mechanisms underlying the impact of a particular policy on a carefully selected set of agents. Their focus or empirical strategies, nonetheless, are local and low-dimensional. Drawing key insights from rarely occurring settings (especially for causal identification) or partial-equilibrium theories that are meant to be verified in the data, they lack the power to explain larger sets of empirical outcomes, produce counterfactuals, or yield generalizable recommendations (beyond, e.g., the specification of the theoretical model and local causal effects).[3]

Structural estimations are more successful in handling model complexity to produce counterfactuals. They model environments more holistically, striving to retain interpretability and a clear understanding of the underlying economics. For tractability, however, they restrict the modeling space to analytical theory and conventional Markov-decision-processes (MDPs) with pre-specified (by researchers) transition probabilities — they do not model dynamic feedback and learning about the environment. Despite its advantages, structural modeling entails strong assumptions and heavy parameterizations. They still leave much to be desired when it comes to explaining true data outcomes, learning from data, making out-of-sample predictions, or ultimately, guiding managerial decision-making in practice.

It is useful to derive theoretical insights or document systematic or causal patterns in the data. But ultimately, and equally importantly, we need to put the pieces together to understand data patterns jointly and coherently, and insights have to be transformed into practical and implementable policy recommendations. To this end, we integrate deep

[3]Reduced-form models abstract away from describing full data counterfactuals. Counterfactuals are only used locally to assure the case for causality; à la "falsification tests."

learning, offline reinforcement learning, and robust control based on ambiguity to cast the concept and practice of corporate finance as a data-driven-robust-control (DDRC) problem. The framework we propose ("AlphaManager model") is essentially a data-driven counterpart to structural estimations. Critically, it allows us to search in a larger modeling space — model spaces that are consistent with theories as well as structures we learn from the data — for the best depiction of the economic system, map and explain corporate outcomes, and make high-dimensional and effective recommendations for enterprise decisions. AlphaManager uses deep neural networks to accommodate the features of financial big data such as high-dimensionality and non-linearity. It also utilizes offline deep RL to reduce the search cost and allows the incorporation of flexible managerial objective and dynamic feedback.[4] Finally, the robust control techniques such as the ambiguity aversion we introduce as a constraint helps AlphaManager to be more conservative when facing high model uncertainty such as potential overfitting and data shifts, mitigating their risks.

AlphaManager first constructs predictive environments to test counterfactuals, effectively explains and predicts observed variation in corporate outcomes, and ultimately allows us to recommend the optimal combination of managerial actions. It does so without the need for costly experimentation nor the need to establish causal links, while at the same time allowing the environment (e.g., financial markets) to influence observed results via feedback effects and other dynamics. In addition to its enhanced empirical performance — and despite its apparent complexity — DDRC modeling complements and advances upon reduced-form and structural approaches in a number of ways. First, it uncovers empirical patterns that lead to more theory research. Second, it informs us when one cannot effectively learn from historical data and theory so that reduced-form and structural models prove to be crucial for learning. Finally, it can incorporate and integrate both reduced-form models as well as structural models through transfer learning, another concept from AI and data science that can be tailored for finance research.[5]

AlphaManager also utilizes a RL module for managerial policy optimization, making our

[4]Because managers make decisions sequentially, accumulating knowledge about the economic environment while receiving feedback from it, RL is the ideal modeling approach for one to understand and advise managerial decisions: the process of RL involves iterative accumulation of experience for improving the action policy (Sutton and Barto, 1998). Unlike dynamic programming used in structural estimation, which solves a MDP problem with known transition probabilities and rewards taken from a particular theory construct, RL solves a more general MDP problem where transition probabilities and rewards are unknown. RL does it either using a model-based approach (e.g., AlphaManager which learns the a model of the environment from data), or a model-free approach (e.g., Q-learning). The AM approach is model-based.

[5]The pre-training in various large language models (LLMs) is a form of transfer learning.

paper the first to combine RL with robust control in corporate finance. At a high level, corporate finance problems involve a "system" (the firm and its environment) and a "controller" (the manager).[6] The controller applies a control strategy to control the system in an optimal fashion, subject to two realistic limitations: (1) its knowledge of the system state and the environment (internal states can be learned through "system sensors;" e.g., accounting, auditing, reporting) and (2) the system parameters that it can control. Simply put, (1) is about the information set and knowledge of managers, while (2) is about their action space and tools. Correspondingly, we operationalize these two building blocks of AlphaManager: First, we use deep learning to train a predictive environmental module (PEM) from historical data that allows effective counterfactual analysis and data generation. We then use RL to pin down the most effective combination of managerial actions for a given business objective. RL derives the optimal control/action trajectory for a given reward structure (e.g., appreciation in market capitalization) with unlabeled data and allows interactions with the environment. It provides normative suggestions on the optimal decisions that managers could make under a given condition. As we explain below, we put the robust control theory into practice and then guide the training of AlphaManager to retrieve optimal decisions for firms under ambiguity aversion (Hansen and Sargent, 2023).

Our AlphaManager application is trained on Compustat, which contains a long panel of fundamental variables of US-listed firms. Specifically, we study nearly 20 thousand distinct firms with over half-a-million unique firm-quarter observations, ranging from 1976 to 2023. We supplement our dataset with additional stock market data from CRSP, and incorporate macro-level data; e.g., the National Financial Conditions Index (NFCI), from the Chicago Fed. We define two sets of variables: state variables and managerial decision variables. State variables describe the state that a firm faces in a given period of time. Typical examples involve fundamentals of firms (internal states), and macroeconomic or market conditions (external states). Managerial decision variables mediate the future state dynamics and also

[6]Conventional control theory is typically targeted towards solving linear equations with well-defined objectives, law of motions for states, and constraints. However, real-world systems are non-linear. Techniques that convert non-linear equations into linear ones and then solve them using linear state space control theory are limited to specific classes of equations and cannot be generalized. Adding to the difficulty of solving non-linear equations, one needs to know how to model a system (its corresponding equations) in the first place. This is why the use of traditional Model-Driven Control strategies — structural models in corporate finance — requires the exclusion of any systems whose modeling is not fully known. Notably, the complexity of such systems is only increasing when one wants to solve hyper-scale problems; problems ranging from contagion in financial markets to responses to climate change. Unknown complex true models can be highly non-linear and are often coupled with unknown environmental states, these conditions pose a challenge to usual dynamic programming and simulation under high-dimensionality.

influence the utility functions of managers, which are made by managers based on the contemporaneous states in maximizing their utility.

In PEM, a deep neural network is trained to minimize the mean-squared error (MSE) between real future states and predicted ones, with the input of current states and current managerial decisions. Our PEM achieves high accuracy in predicting and explaining the evolution of firm outcomes (the state variables) with the help of information on managerial control, i.e., the managerial planning one period forward. For instance, AlphaManager produces a 64.72% cross-sectional R^2 for book asset growth and 3.23% for market cap growth, both out of sample. To make it comparable to other empirical research in corporate finance, we also calculate the predictability without managerial planning information. The predictability and explanability remain high for many variables but more interestingly, for outcomes such as book asset, market cap and enterprise value, the out-of-sample R^2s become negative. This finding is consistent with empirical asset pricing research where the market environment is simply too noisy. But more importantly, these results inform us which managerial decision variables actually matter for the firm dynamics going forward.

Based on PEM, we are able to analyze the heterogeneous outcomes and model ambiguity under counterfactual managerial decisions by state variable, sector, book-to-market decile, and macroeconomic regimes. PEM performs particularly well in trade and transportation, education and healthcare, manufacturing, among other sectors, and during expansions the predictability is also higher. Firms with higher firm-level states in the cross section are with lower MSE and ambiguity. The model ambiguity is higher for high book-to-market deciles.

In the RL module, with an exogenously specified utility function of managers, we obtain optimal policies that generate managerial decisions that maximize utilities based on current states. When optimizing the short-term market cap growth, the quarterly out-performance of optimal decisions compared to real managerial decisions is 10.07%, and when optimizing the long-term market cap growth, the quarterly out-performance becomes 8.73% on average. We also find similar patterns when the objective is set as enterprise value growth, with quarterly out-performance 4.43% and 2.71% correspondingly.

We contrast RL suggested managerial decisions under short-term versus long-term firm value growth, as well as the term structure of firm valuation growth under optimal decisions. We find that AlphaManager with a short-term objective tends to employ a higher leverage, while higher leverage is not beneficial in the long run. Short-term AlphaManager outperforms long-term RL in the long run but with higher ambiguity, and after ambiguity adjustment, the performance of short-term RL is decreasing in the evaluation time hori-

zon compared to the long-term RL. Furthermore, we discuss the long-term implication of firm valuation under managerial short-termism from the point of view of board members. Notably, this approach does not require any knowledge of real decisions, which is the key difference between RL and commonplace supervised machine learning algorithms.

Naturally, the action and information space of a manager is vast. One might wonder about the set of variables needed for a given application and about when should one stop collecting data and constructing new variables. Conveniently, our proposed approach is not hampered by these concerns. Among other things, most variables one uses in applied corporate finance research already convey information about many others. For example, the firm's capital structure already conveys information about its access to credit, risk, asset mix (collateral), and more. The same applies to firm size, or as to whether it is publicly traded, hence the amount of information available to investors. Data "confoundedness" is a plus — not a challenge — to our data-driven approach. The approach we propose is less prone to criticism regarding inconsistencies in variable selection made by researchers in existing studies (e.g., see Mitton, 2022)—inconsistencies leading to practices like P-hacking and *ex-post* theory-fitting.

Given the large counterfactual tests we do using the predictive environmental module, AlphaManager necessarily makes extrapolation and interpolation from historical data. Given that historical data are endogenously generated, potential model misspecification poses a particular challenge. The literature traditionally examines parameter stability using Chow-type tests (Andersen et al., 2015). One can also detect static misspecification's by testing moment conditions under a GMM framework (e.g., Pan, 2002). For dynamic misspecification tests, Jarrow and Kwok (2015) provide an intuitive exact calibration approach utilizing analytical theoretical models. Our use of deep neural networks goes beyond economic theory and is data-driven. While it explores a much larger modeling–functional space, model misspecification is generally dynamic and hard to detect with conventional econometric tools. Therefore, we adapt the new approach of utilizing ambiguity aversion and entropy-based measures to guard against and mitigate misspecifications to a data-driven setting. The model ambiguity measure also informs us of scenarios where additional knowledge from theories and causal identifications is important and the ones where predictive models trained on historical observations suffice. It not only guides corporate finance researchers on the dimensions to focus on, but also allows the integration of fragmented knowledge through ambiguity-guided transfer learning.

Our study contributes to the emerging literature on AI in finance. Machine learning and natural language processing have been widely applied in investment and asset pricing (e.g.

Cong et al., 2020, 2021; Gu et al., 2020; Feng et al., 2020). They have also seen applications in studies on corporate finance or financial market risk (e.g., Cong et al., 2018; Li et al., 2020; Bellstam et al., 2020; Hanley and Hoberg, 2019; Campello et al., 2024); but their applications beyond creating or improving the measure of some explanatory variables are rather limited. Exceptions to this line of work include Erel et al. (2018) on predicting board director performance, Cao et al. (2023) on how machines and managers interact in the context of conference calls and earnings announcements, Cao et al. (2021) on how an AI analyst can provide additional insights on stock market forecasts to human analysts, and Gulen et al. (2020) on applying causal random forests to analyze how financing affects investment. We note that these studies employ standard — often rudimentary — models designed for prediction or focus exclusively on supervised learning (learning through examples) without applying the core paradigm (i.e., deep RL) in AI innovations in the past two decades. In contrast, we present the first AI and robust control application in finance. In fact, we are among the first to apply model-based offline RL in economics to offer a data-driven alternative to reduced-form models and structural estimations.

Our paper also contributes to the literature on model uncertainty ("ambiguity") and robust control. Ambiguity represents a source of uncertainty where an economic agent is not confident in which prior models to choose for prediction tasks Hansen and Sargent (2023).[7] Its applications in finance are rare and mostly focus on asset pricing and investment (e.g., Wang, 2005; Mamaysky et al., 2007; Dicks and Fulghieri, 2021), with Dicks and Fulghieri (2019) (financial intermediation), Garlappi et al. (2017) (corporate investment), Izhakian et al. (2022) (capital structure), and Malenko and Tsoy (2020) (security design) as exceptions. Meanwhile, robust control has been put forth in Hansen and Sargent (2001) where a rational agent solves a stochastic control problem under ambiguity aversion. Studies about robust control are mostly theoretical, except for Barnett et al. (2020) which applies robust control to the context of climate change risks. Our work contributes to the literature on model uncertainty as the first empirical/methodological application of the ambiguity concept in corporate finance. We also add to robust control studies by estimating ambiguity with the help of deep learning and then approximate solutions to robust control problems using estimated ambiguity and offline RL.

Finally, our work can be placed in the context of the emerging computer science literature

[7]Hansen and Sargent (2023) defines three sources of uncertainty: risk (in-model innovation), ambiguity (model uncertainty), and misspecification (model class uncertainty). Campello and Kankanhalli (2024) provide a comprehensive review of the research of uncertainty in corporate finance.

on offline RL (a.k.a, batched RL, e.g., Fujimoto et al., 2019; Kidambi et al., 2020).[8] A number of papers have illustrated the power of such an approach in enabling data-driven learning of policies for dialogue (Jaques et al., 2019), robotic manipulation behaviors (Ebert et al., 2018; Kalashnikov et al., 2018), and robotic navigation skills (Kahn et al., 2021). While Cong et al. (2020) is the first finance paper that applies offline RL (with online updates) to portfolio management, the authors do not fully optimize the environment module to mimic the real environment. We instead follow Kidambi et al. (2020) to build and optimize an environment module to calculate the transition probability across states without requiring experimenting via a simulator or costly interactions with the actual corporate or market environment. Together with Chen et al. (2023), we are the first study utilizing transfer learning in finance. Our innovation lies in using ambiguity to guide the choice of transfer learning and the application in corporate finance. More broadly, our study adds to emerging studies utilizing AI for goal-oriented search, which involves both heuristic search using RL and greed search using panel trees (Cong et al., 2022, 2023).

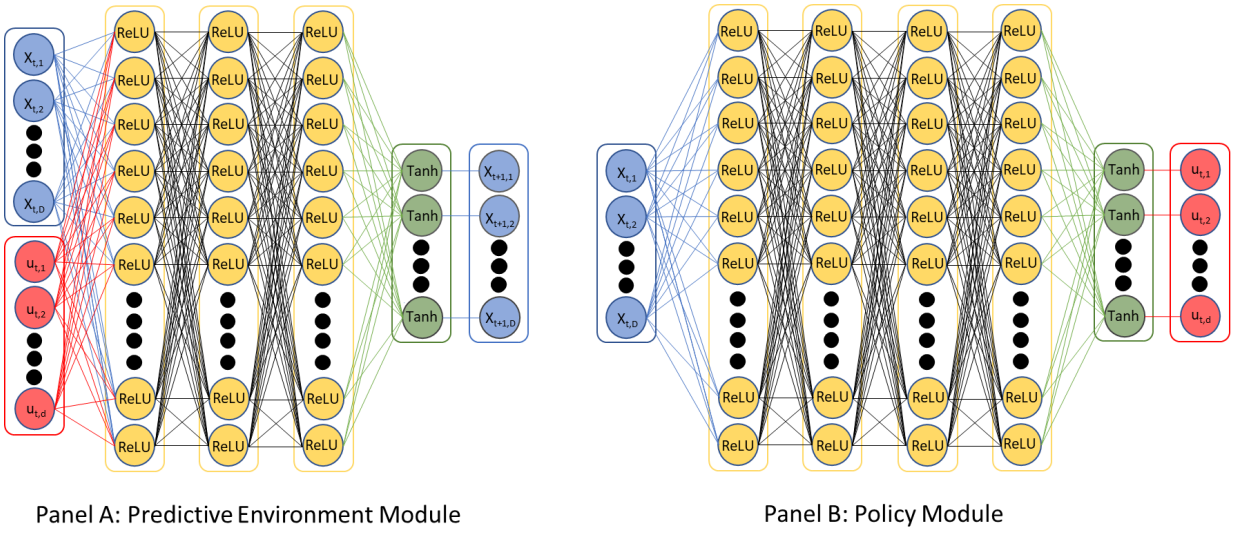
The rest of the paper is organized as follows. Section 2 describes the DDRC framework. Section 3 details data and model training. Section 4 reports the functionality and empirical results from PEM, whereas Section 5 investigates the optimal managerial actions AlphaManager recommends under various given objectives. Section 6 links DDRC to extant corporate finance research using ambiguity-guided transfer learning. Section 7 concludes.

2 The Data-Driven-Robust-Control Framework

The DDRC approach is comprised of two modules, one describes the market environment and how the outcomes of interest respond to managerial actions, and the other a reinforcement learner offering an optimal policy while dynamically interacting with the market environment and incorporating feedback. The resulting AlphaManager architecture is illustrated in Figure 2. The use of the rectifier or ReLU (rectified linear unit) and Tanh activation functions are standard in deep learning; the state variable vector X_t and managerial control vector u_t are introduced shortly.

[8]Offline RL often works only on historical data without any online interactions with the environment to generate additional data for model training (e.g., Fu et al., 2020). In a pure data-driven scheme, offline RL enables researchers to explore fields that are considered infeasible by classical online RL algorithms, especially those closely related to human behaviors where environment interaction is costly, infeasible, or dangerous (Levine et al., 2020).

Figure 2: AlphaManager (AM) Architecture



Note: The AlphaManager architecture. Panel A displays PEM with an input layer of a state vector $\mathbf{X_t}$ concatenated with a decision/control vector $\mathbf{u_t}$. $\mathbf{X_t}$ (in blue color) includes firm-specific state variables such as firm fundamentals, macroeconomic variables, etc. $\mathbf{u_t}$ (in red color) contains leverage, cash holdings, equity financing, etc. Appendix A contains a full list of firm-specific state variables and control variables. The second, third, and fourth layers are fully-connected layers with rectified linear unit (ReLU) activation functions. The fifth layer of Tanh function transforms the output spectrum to interval [-1, 1] and the last layer is the output layer with predictions of the system state evolution. Similarly, Panel B shows the structure of the policy module mapping current state vector to the optimal control for a given managerial objective. The policy module has one more hidden layer and the outputs are the controls. When implemented, AlphaManager uses the policy module to generate the optimal managerial actions and then predicts system states using PEM.

2.1 Predictive Environment Module (PEM)

To conduct counterfactual analyses in our context, we need to model the market environment. Conventional causal analyses (e.g., instrumental variables and regression discontinuity designs) do not provide a comprehensive solution because these identification strategies are hard to come by or costly to establish (e.g., through experiments). Even when we have them, causal links are only identified locally and typically when varying a single treatment dimension. We are interested in learning about the outcomes of not only large corporate events, such as mergers or bankruptcies, but also any marginal decisions, such as combinations of increasing dividend payouts and reducing investment spending — decisions that are far more frequent and high-dimensional in the practice of financial management. By building a market environment, we are taking a model-based RL approach that is similar to a structural estimation, except that the structure is data-driven rather than based on sparse theory.

In AlphaManager, we use a deep neural network to model how the internal state vari-

ables and external market states going forward react to potentially high-dimensional managerial actions. Neural networks are designed to capture non-linear dependencies in high-dimensional spaces, and they have the potential to approximate any functional form (referred to as the *universal approximation theorem* in computer science, see, e.g., Hornik et al., 1989). The large number of corporate finance variables and decisions available and the potential nonlinear interactions among them necessitate such a module, which we refer to as the Predictive Environment Module (PEM).

Our approach recognizes that corporate decisions involve the manager taking a sequence of high-dimensional actions to optimize some given economic objective, which makes them stochastic control problems. Suppose the dynamics of the state variables in the system follow:

$$\Delta X_{t+1} = f(X_t, u_t) + \varepsilon_{t+1},\ \varepsilon_{t+1} \sim N(0, \Sigma) \tag{1}$$

where X_t denotes a firm's internal state variables (e.g., accounting fundamental) and external state variables (e.g., inflation and unemployment) which are out of the manager's control at time t, u_t stands for the vector of managerial actions that are made by managers at time t, f describes how the expected value of state change, ΔX_{t+1}, corresponds to current state and managerial actions, and ε_{t+1} is the normally distributed risk term which represents the in-model uncertainty (i.e., the uncertainty that has been captured by the model). We can then use a fully connected neural network to approximate the function f.

We need PEM to make out-of-sample predictions and counterfactual statements, which necessarily involves extrapolation and interpolation. This is particularly dangerous, as historical data are generated by endogenous managerial decisions and some action/state space may not be covered. Just as in reduced-form models, two obstacles emerge: potential overfitting and data shift. Note that these concerns are less worrying in structural estimations because the theory model is taken to be the true model of the world. A similar argument here would be that the state variables we include fully describe the system's evolution.

There are statistical procedures to mitigate the concerns more generally without such strong assumptions. The first concern can be allayed in a number of ways such as adding L1-norm or L2-norm-based penalties in the loss function in training a supervised deep neural network (e.g., Kaniel et al., 2023). Given the virtue of complexity (e.g., Kelly et al., 2024), our over-parameterized PEMs in the empirical implementation also reduce overfitting and OOS performance. The second concern means that PEM necessarily introduces some model ambiguity (Hansen and Sargent, 2023) because even without overfitting, the data-generating

process could have changed so that it is uncertain a properly trained model would work on a new dataset. In other words, many neural network models with different parameters may perform similarly for a given training set. Yet they may give different predictions due to a combination of overfitting and potential data shifts. Thus, choosing the right model *ex ante* faces an additional source of uncertainty. In Section 2.3, we show how to empirically estimate the ambiguity for any given point in the variable space, and how we can address the issue by introducing ambiguity aversion to the objective function.[9]

2.2 The Policy Module and Offline Reinforcement Learning (RL)

Managers take a sequence of actions to solve the following stochastic control problem:

$$\max_{\{u_{t_0},\ldots,u_{t_0+T}\}} \mathbb{E}_{t_0} \sum_{t=t_0}^{t_0+T} r(X_t, u_t) \qquad s.t.\ \Delta X_{t+1} = f(X_t, u_t) + \varepsilon_{t+1},\ \varepsilon_{t+1} \sim N(0, \Sigma), \tag{2}$$

where $r(X_t, u_t)$ is the instantaneous reward function given the state X_t and the managerial decision u_t, and T is the length of the optimizing time period associated with the managerial objective. An example of instantaneous reward function is the enterprise value growth for the next period given current state and managerial decisions.

The canonical approach to such stochastic control is to derive Bellman/HJB equations and closed-form solutions. When analytical solutions are lacking, numerical algorithms can provide a reliable (low estimation variance) empirical solution for problems of low dimensionality. However, one has to make strong assumptions concerning the empirical specification of the environment. In our setting, there is no analytical characterization of the dynamics, and the stochastic evolution of the environment is high-dimensional and potentially non-linear, which motivates our use of RL.[10]

In order to functionally represent managerial decisions of interest, we follow the canonical policy gradient algorithm in RL (Sutton and Barto, 1998) and assume that decisions are made

[9]This essentially adds a penalty based on, e.g., relative entropy (also known as Kullback-Leibler divergence in computer science literature.

[10]Reinforcement learning is "learning what to do — how to map situations to actions— so as to maximize a numerical reward signal," constituting one of the three paradigms of modern machine learning together with supervised learning and unsupervised learning. In RL, an agent learns about states of its environment and takes actions that potentially affect states going forward as well as its objective function to maximize.

contingent on the state variable X. Specifically,

$$u_t = g(X_t), \tag{3}$$

where the functional representation g is called policy module, serving as a decision-making device contingent on the state.

Substituting (3) into (2), we can reformulate the stochastic control problem as:

$$\begin{aligned} \max_{g(\cdot)} \quad & \mathbb{E}_{t_0} \sum_{t=t_0}^{t_0+T} r(X_t, u_t) \\ s.t. \quad & \Delta X_{t+1} = f(X_t, u_t) + \varepsilon_{t+1}, \quad \varepsilon_{t+1} \sim N(0, \Sigma) \\ & u_t = g(X_t). \end{aligned}$$

An AlphaManager learns about the environment through interactions and updates her understanding of the world through PEM, which tells about $f(X_t, u_t)$. At the same time, she is maximizing over a large space of dynamic policy $g(X_t)$ the expected cumulative payoff.

2.3 Robust Control and Ambiguity Aversion

Neural networks usually have more free parameters than training data, making the estimation of neural network parameters under-identified. Even with less parameters, potential endogeneity problems in the training dataset undermine a model's OOS predictive power, because data-driven training has no ability to identify any causal link or to separate shocks from supply side from demand side. As a result, through the same training process, different initialization of parameters or variations of model architecture may result in multiple models with similar in-sample performance, and thus potential different OOS predictions. Traditionally, the model selection focuses on the performance on a validation set. But if the prediction performance is similar across models, examining the first moment of prediction is no longer sufficient. Data shifts exacerbates the issue. The dispersion in the predictions on the test set therefore not only reveal potential overfitting, but also inform us about model uncertainty due to data shifts. This "across-model" uncertainty is referred to as *ambiguity* in robust control literature (Hansen and Sargent, 2023).

The literature typically handles ambiguity problems in two steps. First, researchers estimate ambiguity through some metrics, e.g., relative entropy, to summarize the discrepancy among multiple models defined as statistical distributions indexed by different parameters.

If this discrepancy for a given test sample is large, the ambiguity for this specific sample is high. Second, when modeling a decision-maker's problem, a punishment for ambiguity is added to the original objective function, making the agent ambiguity-averse.

To estimate ambiguity, we apply the idea of boosting to train a set of neural networks with similar structures for PEM, under the same loss function but with different initial coefficients or slight variations in the model specification.[11] In the supervised learning scheme, these networks perform the same functionality in-sample, as errors would be punished during the training process consistently. In contrast, out-of-sample, especially for unprecedented scenarios or unusual events that are not covered in the training sample, the performances of these models might diverge. Accordingly, the disagreement level of these models is a natural proxy of whether a given input is likely within the training set span or not. When the disagreement level of these models is high, it is either because the current states are "unusual" or the current managerial decisions are "uncommon." Formally, we trained a set of environment modules $i = 1, 2, ..., I$ mapping (X_t, u_t) to $\hat{X}^i_{t+1}$. The boosting error, motivated by the relative entropy which is a theoretical metric of ambiguity, for (X_t, u_t) is calculated as:

$$\text{BoostingError}(X_t, u_t) = \frac{1}{D} \sum_{d=1}^{D} \left(\max_{i=1,2,...,I} \hat{X}^i_{t+1,d} - \min_{i=1,2,...,I} \hat{X}^i_{t+1,d} \right)^2, \tag{4}$$

where D is the number of dimensions for state vector X and u_t is an arbitrary managerial control attempted. Note that there are alternative specifications of boosting errors one can use, which still capture the concept of ambiguity.

In addition to punishing ambiguity directly, a common practice in economics and computer science is to consider an agent solving a max-min problem.[12] Under model uncertainty, the agent tries to maximize the lowest reward generated by a set of models. This pessimism effectively serves as an additional device, together with ambiguity aversion, to control for model uncertainty in suggesting decisions.

To train the policy module and solve the RL problem, we use fully connected neural networks to parameterize the decision-making device g, the mapping from state to control. We also incorporate the idea of Kidambi et al. (2020) and Yu et al. (2020) on how to incor-

[11] The choice of ambiguity metrics could vary, as shown in Hansen and Sargent (2023). We choose to use boosting error in our context given its simplicity of calculation and its intuition of maximum dispersion of predictions for models around our main PEM in the sense of model structure and parameter initialization under the same training process.

[12] See Barnett et al. (2020) in economics and Jin et al. (2021) in computer science for modelling pessimistic agents solving max-min problems to mitigate ambiguity.

porate pessimistic reward and punishment on ambiguity in the training process. Specifically, in training the policy module, for any given state X_{t_0}, the empirical objective function is calculated as:

$$J(X_{t_0}) := \sum_{t=t_0}^{t_0+T} \min_{i=1,2,\ldots,I} r^i\left(X_t, g(X_t)\right) - \delta \cdot \max\left\{0, \max_{t=t_0,\ldots,t_0+T} \text{BoostingError}\left(X_t, g(X_t)\right) - \theta(X_{t_0})\right\} \tag{5}$$

In Eq (5), $r^i\left(X_t, g(X_t)\right)$ denotes the instantaneous reward function by environment module i, given state PEM-predicted time-t state, X_t, and the decision-making device g, and we take the lowest instantaneous reward across I different environment modules. The ambiguity punishment parameter, δ, is chosen to be very large so that we guarantee that the boosting error is always small enough. Empirically we pick $\delta = 10^7$. $\text{BoostingError}(X_t, g(X_t))$ is the boosting error at time t, given PEM-predicted time-t state, X_t, and the decision-making device g, calculated as Eq (4). The benchmark error, $\theta(X_{t_0})$, is calculated as a constant α plus β times the boosting error of X_{t_0} and the lagged real managerial decisions $\bar{u}_{t_0-1}$ which is known at time t:

$$\theta(X_{t_0}) = \alpha + \beta \cdot \text{BoostingError}(X_{t_0}, \bar{u}_{t_0-1}), \tag{6}$$

where $\beta > 0$ and $\bar{u}_{t_0-1}$ is the actual managerial decision in the previous period just a simple benchmark for how ambiguous the states correspond to. What we do here is essentially prioritizing the reduction in ambiguity over the baseline rewards, if the ambiguity exceeds some threshold. When considering different optimizing period T, given the monotonicity of $\max_{t=t_0,\ldots,t_0+T} \text{BoostingError}\left(X_t, g(X_t)\right)$, we choose higher values of α and β when optimizing under a higher T. The empirical choices of α and β under different optimizing period T are detailed in Appendix C.

3 Data and Training

3.1 Data and Variables

Our panel data is based on quarterly CRSP–Compustat merged database, where we are able to include 20,485 different firms ranging from 1976:Q1 to 2023:Q4, with 784,460 firm-quarter observations. The variables in this paper fall into two classes: state variables (X_t) and managers' decision variables (u_t). In each period, managers make their decisions

and their decisions will impact state variable dynamics for the next period. Examples of state variables include fundamental variables that are not entirely determined by managers, such as the firm's market capitalization, interest expenses, sales growth, earnings forecasts, and Tobin's Q. They also include past managerial decisions. Decision variables u_t reflect managerial decisions directly (or at least largely so). Examples include corporate investment spending, cash savings, and capital structure.

In order to train our supervised learning-based predictive environment module, we require the existence of observations for two consecutive fiscal quarters in standard Compustat dataset (with "datafmt" code to be "STD"). We only focus on domestic (with "popsrc" code to be "D") industrial companies (with "indfmt" code to be "INDL"), and we exclude financial and utility firms (with "naics2" code not to be 22 nor 52) from our sample. We also require book assets to be positive. AlphaManager has a rolling training for RL. We divide our dataset by fiscal quarter to have roughly 30% initial training set, which implies that the initial burn-in is until 1991. We winsorize each variable at 1% and 99%, and subsequently normalize them to be bounded inside $[-1, 1]$ for training convenience. We also consider macroeconomic states for the same time period, using the Chicago Fed National Financial Condition Index (NFCI) subindices (risk, credit, financial leverage, and non-financial leverage) as macro covariates. The detailed variable selection is listed in Appendix A.

3.2 Summary Statistics and Stylized Patterns

Summary statistics are reported in Table 1. We observe some stylized patterns either linking managerial actions to the evolution of system state variables, especially firm fundamentals, or about conventions of managerial decision-making. For these patterns, we first want to compare the counterfactual outcomes predicted by stylized patterns and our PEM. If PEM provides different counterfactual outcomes than some widely believed stylized patterns, it is likely due to the lack of training data with unrealistic managerial decisions. To take these patterns into account, a transfer learning approach can be applied to augment our training sample by adding synthetic data generated by these stylized patterns to the training sample of PEM. We detail this process in Section 6.2.

We then want to compare the counterfactual decisions made by the stylized pattern and our RL module. They can be different because (1) RL is for optimal decisions (under certain objectives) while stylized patterns are based on real decisions; or (2) our RL module misses out on important information such as whether a firm is financially constrained or distressed;

Table 1: Summary Statistics

variable	count	mean	median	std
lev_market	784,460	0.2449	0.2020	0.2333
aqc_fund	784,460	0.0047	0.0000	0.0227
inv_ratio2	784,460	0.0144	0.0068	0.0227
cash_hold	784,460	0.2041	0.0932	0.2651
payout_div	784,460	0.0019	0.0000	0.0051
detb_issue_log	784,460	0.9122	0.0000	1.8047
eq_issue_log	784,460	0.5231	0.0010	1.0551
rd_exp	784,460	0.0130	0.0000	0.0309
payout_rep	784,460	0.0026	0.0000	0.0095
log_at	784,460	5.2788	5.0987	2.2292
log_act	784,460	4.2719	4.2478	2.1785
log_sale	784,460	3.7052	3.6178	2.2490
log_ap	784,460	2.7043	2.3038	2.0113
log_cogs	784,460	3.3015	3.1079	2.1523
log_intpn	784,460	0.5467	0.0000	1.1486
log_invt	784,460	2.4818	2.1510	2.2778
log_lct	784,460	3.5870	3.3725	2.1935
log_rect	784,460	3.1032	2.9350	2.1419
log_revt	784,460	3.6999	3.6127	2.2525
log_market_cap	784,460	5.2181	5.0620	2.2858
log_enterprise_val	784,460	5.7726	5.6021	2.2511
log_VOL	784,460	10.3049	10.4611	2.6685
log_RET	784,460	-0.0140	0.0000	0.2872

or (3) real managers have preferences that differ from what is specified in our policy module.

3.3 Training and Computation

We describe the training of the AlphaManager model here and list the hyperparameter choices in Appendix B. PEM has 268 inputs, including 12 firm fundamental variables (Compustat state variables in Appendix A), 12 variables capturing their growth (first difference in variable values). We further incorporate the corresponding variables lagged 1 through 4 quarters to capture patterns in the time series. Involving lagged variables provides additional information on time dependence, and incorporating the changes in values helps neural networks better extract more information than typically learned from the levels of state variables alone. We additionally include lagged decisions and their growth, which are publicly observable at the current time point, as well as their va1-4 quarter lagged versions. We also have 2 stock market states and the current growth, together with their 1-4 quarter lagged values. For the 4 macro states, we only consider their current levels and their levels lagged 1 through 4 quarters. For controls, we have 9 action variables and correspondingly 9 changes (growth) from their previous values. The output of PEM has 14 dimensions, namely 12 next-quarter changes in fundamental and 2 next-quarter changes in return and volume.

For training PEM, we first initialize a neural network and train the model in mini-batch mode (with batch size to be 2048 observations) using data until 1991:Q4, which is ready to predict system states in 1992:Q1 "out-of-sample." In this stage, we train our neural network for 40 epochs. Next, we include 1992:Q1 data into our training set to update the environment module and use the updated model to predict the system states in 1992:Q2 for additional 5 epochs, and so on. We also train 10 auxiliary models with different realizations in the initialization phase and with slightly different model designs at the same time for ambiguity estimation. We use three hidden layers in the baseline PEM, with each layer having 300 neurons. In total, we have 265,514 parameters.

Our policy module has 250 inputs, namely the inputs of PEM excluding current decisions and their growth versions ($9 + 9 = 18$), and has 9 outputs, which are changes in current decisions compared to the last-quarter ones. For training policy module, we initialize a neural network and train the RL model using data until 1991 to recommend managerial decisions in 1991. We train the model for 64 epochs first, and then when the loss function (after ambiguity punishment) turns positive, we continue to train the model for 5 epochs and move on to the next stage. Then, we focus on 1992 data to recommend 1992 managerial decisions, and follow

the same training scheme going forward in time. As baseline specifications, we consider two objectives, market capitalization growth and enterprise value growth. We do this for two horizons: one quarter ahead ("short horizon") and eight quarters ahead ("medium/long horizon"). We use a four-hidden layer network in training our policy module, with each layer having 256 neurons. In total, we have 265,993 parameters.

We train our neural networks on the Red Cloud platform, with 16 CPUs, 55G memory, and an A100 GPU.[13] The training process takes 30-35 hours for PEM (including the auxiliary models), 8 hours for the policy module with short-term objectives, and 75-80 hours for the policy module under long-term objectives. We use Adam optimizer (Kingma and Ba, 2014) in training neural networks.

4 The Economic Environment for U.S. Public Firms

4.1 Supervised Learning of PEM

When implemented on U.S. equities, PEM predicts next period system reactions to current period managerial actions. In other words, environment module predicts X_{t+1} given X_t and u_t. As discussed earlier, a deep neural network representation could easily fit the training sample, but is not guaranteed to produce sensible out-of-sample counterfactuals, especially in the presence of distributional shifts.

Suppose there is a rarely observed (or an unseen) event in the training sample, which is the general focus of event studies and causal identification strategies. One can use use a dummy variable to denote whether this event occurs or not. In the training sample, this variable is almost always labeled 0 and only rarely 1, which makes it a phenomenon too "weak" to pick up. In this case, the ambiguity for our test sample is infinite, i.e., any models trained on our training set could provide unconstrained estimations on the treatment effect. To address this concern of ambiguity, we follow the idea of boosting and train a set of auxiliary neural networks to estimate the ambiguity by their dispersion in predictions.[14]

Acting as a metric of ambiguity, boosting error differentiates scenarios where all models

[13]Red Cloud is a subscription-based Infrastructure as a Service cloud that provides root access to virtual servers and storage on-demand under Cornell University Center for Advanced Computing (link).

[14]The main difference between our boosting method and commonly used boosting algorithms such as AdaBoost lies on how to use the auxiliary predictors. These auxiliary networks share a very similar structure with our neural network but with different initialization. Traditional applications combine all the auxiliary predictors to be a stronger predictor, while we use auxiliary predictors to jointly form a natural gauge of model ambiguity.

coincide in predictions versus ones where model predictions are highly dispersed. By punishing the boosting error in the training of our policy module later on, the ambiguity averse agent avoids high-ambiguity actions which we have little reference from prior empirical data, and only explores the action domain with low ambiguity. Similar ideas are also seen in computer science literature related to offline reinforcement learning, such as Kidambi et al. (2020) and Yu et al. (2020).[15]

4.2 Predicting Firm Fundamentals and Market Reactions

To train PEM, we use standard mean-squared error (MSE) as our loss function. We standardize input variables to be in the range $[-1, 1]$ to fit the $tanh(\cdot)$ function in the neural networks, rather than normalizing them to be with zero mean and unit variance.

To better understand the economic implication of training loss, we convert the technical loss to pseudo R^2 by dependent variables and plot the results in Figure 3. The expression of pseudo R^2 for the d^{th} state for a given sample S_d with corresponding predictions $\hat{S}_d$ is:

$$\text{Pseudo } R^2(S_d, \hat{S}_d) = 1 - \frac{\sum_{X_d \in S_d}(X_d - \hat{X}_d)^2}{\sum_{X_d \in S_d}(X_d - \bar{X}_d)^2}, \tag{7}$$

where $X_d \in S_d$ is a sample of the d^{th} dimension of state vector, and $\hat{X}_d \in \hat{S}_d$ is its corresponding prediction from PEM; $\bar{X}_d$ is the initial training set (i.e., sample before 1991:Q4) average of X_d. It is not surprising that our environment module gains very high pseudo R^2 for most variables, as decision variables are used as network inputs. The pseudo R^2 for stock return growth is not as high as other variables, since it is driven by investors' demand — which incorporates factors like sentiment and liquidity — that is out of managers' control, at least before the control decisions are revealed to the investors.

Table 2 shows the pseudo R^2 of future fundamentals ignoring or with control information. In the table, we notice that for outcomes such as Gross Revenue Growth or Net Income Growth, whether we consider the managerial action does not matter much. However, managerial actions seem to have a significant impact on state variables such as Book Asset Growth and Market Cap Growth. This reveals that some corporate outcomes are influenced by managerial actions while others are not, at least not immediately. We consider both the case with and without controls because from outside investors' perspective, managerial

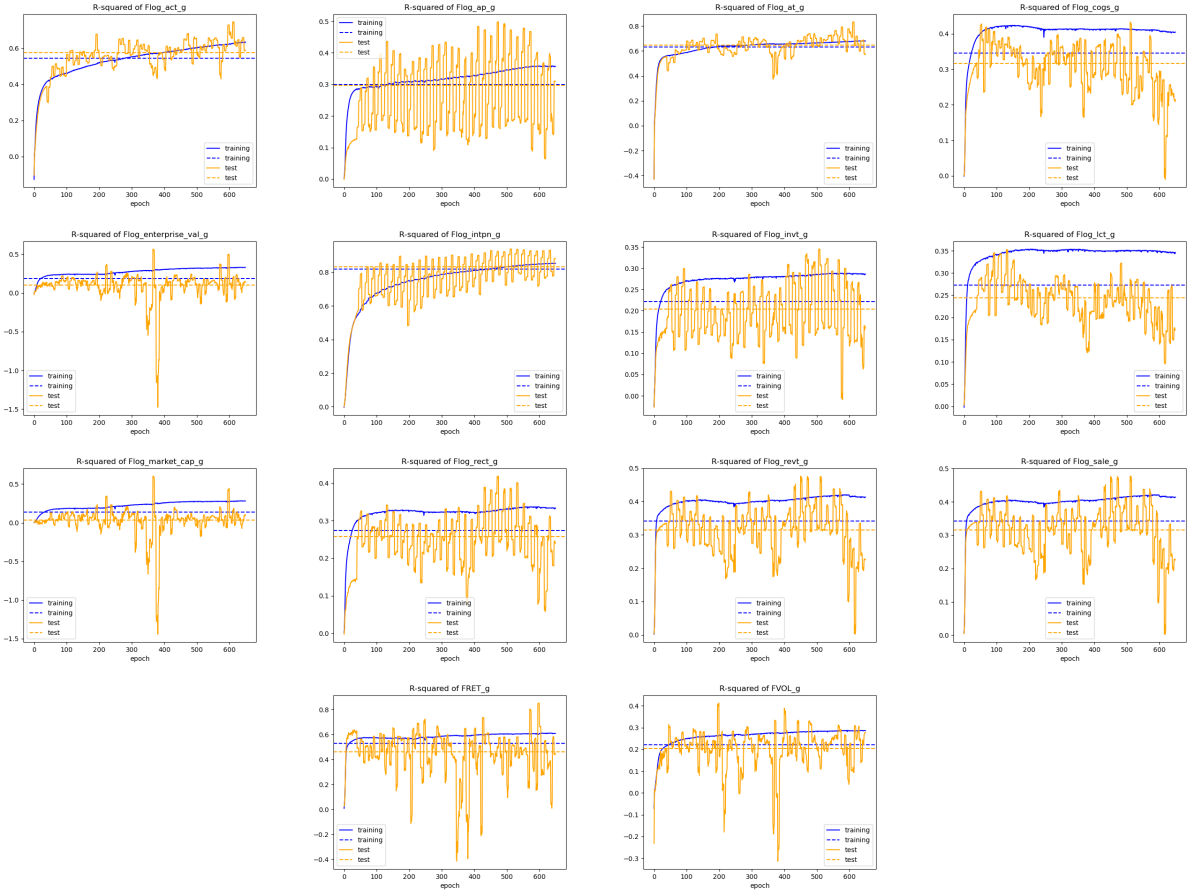
[15] In Kidambi et al. (2020) and Yu et al. (2020), punishment is based on the reliability of action (u_t in our paper) rather than predicted states (X_{t+1}), which is more conservative than our setting.

Table 2: Pseudo R^2 for the Growth of State Variables With and Without Control

	Ignoring Control		With Control	
State Variable	Training R^2	Test R^2	Training R^2	Test R^2
log_at	-12.14%	-15.68%	63.23%	64.72%
log_act	-7.02%	-10.43%	54.41%	57.53%
log_sale	31.78%	28.71%	34.19%	31.47%
log_ap	26.03%	25.61%	29.92%	29.61%
log_cogs	32.84%	29.52%	34.54%	31.61%
log_intpn	81.89%	83.32%	82.03%	83.46%
log_invt	15.07%	13.13%	22.13%	20.38%
log_lct	9.10%	5.80%	27.23%	24.41%
log_rect	22.12%	20.51%	27.42%	25.82%
log_revt	31.77%	28.72%	34.16%	31.44%
log_market_cap	2.47%	-10.12%	13.49%	3.23%
log_enterprise_val	-2.05%	-12.25%	18.80%	10.17%
log_vol	18.64%	15.88%	22.20%	20.48%
log_ret	50.10%	42.85%	52.78%	46.31%

Note: This table documents the pseudo R^2 for the growth of state variables with and without the current control (u_t). PEM intends to predict future firm-specific state variables given current state and control. Since the current control vector is private information (for instance the manager's planning on leverage for the next period), it is not observable by outsiders such as investors or econometricians. In predicting the future, we first ignore all controls, by setting them to be 0, calculate PEM predictions, and then the pseudo R^2 (ignoring control) for each firm-specific state variable. Then, we incorporate control variables as inputs, calculate PEM predictions, and then the pseudo R^2 (with control) for each firm-specific state variable.

Figure 3: PEM: In-Sample and Out-of-Sample Pseudo R^2 for System States



Note: This figure shows how the R^2 evolves during the training process for training (blue) and test (orange) samples. The x-axis is the training epochs, and the y-axis represents pseudo R^2. The dashed lines represent average values of pseudo R^2s. The predicted variables are future growth of current assets (Flog_act_g), accounts payable (Flog_ap_g), book assets (Flog_at_g), cost of good sold (Flog_cogs_g), enterprise value (Flog_enterprise_val_g), net interest paid (Flog_intpn_g), inventory (Flog_invt_g), current book liability (Flog_lct_g), market cap (Flog_market_cap_g), receivables (Flog_rect_g), revenue (Flog_revt_g), sales income (Flog_sale_g), equity return (FRET_g), and trading volume (FVOL_g), with formal definitions detailed in Appendix A.

controls are only disclosed with a delay and constitute "insider" information that outsiders should not use for trading. Another way to interpret this "controllability" result is to look at the outcome variables less affected by the manager. For example, Trading Volume Growth is likely driven by traders in the secondary market rather than directly by the manager of the firm.

Figure 3 displays how PEM performs on both the training and test sets. The x-axis is the training epoch, in which the whole training set has been used to update model parameters once. Because for every 5 epochs (after the first 100 epochs) we use a new data set, it is

possible to have a low R^2 in years where we have big unexpected shocks.

Table 3 reports PEM's heterogeneous performance (measured by OOS MSE) for each state variable by its cross section ranking. We report results for three subsets of cross sections: pre dot com bubble, between dot com bubble and great financial crisis (GFC), and post GFC, where cutoff points of these two events are based on the NBER recession end months. For each state variable, we rank firms in each cross section as "low" and "high" by comparing the state variable level to its cross-sectional median, and calculate average and standard deviation of PEM MSE within each subsample. For most firm-level state variables, the lower half generally has higher prediction error, with the exceptions of book value of current asset, interest paid and book value of current liability. For macro state variables, the dot com to GFC period is the hardest to predict; the same applies to our two objectives (market capitalization and enterprise value). Table 4 reports PEM's heterogeneous performance by sector, defined as the first digit of North American Industry Classification System (NAICS) code. Trade and Transportation sector ("naics1" = 4), Education and Healthcare ("naics1" = 6), and Other Services sector ("naics1" = 8) always have the lowest prediction error across three episodes. Table 5 reports PEM's heterogeneous performance by book-to-market decile, where 1 indicates the lowest decile and 10 indicates the highest decile. The best performers with regard to prediction error of PEM oscillate among middle deciles and the distribution across deciles exhibits a U-shape MSE, while the bottom decile always has the worst performance.

Table 6 results reveal heterogeneous ambiguity for each state variable (measured by boosting error defined in the previous section), following the same logic. For most firm-level state variables, the lower half generally has lower ambiguity. The dot com to GFC period usually has the highest ambiguity, and the pre dot com period has the lowest in general. In Table 7, we do the same exercise for sector. Manufacturing ("naics1" = 3), Education and Healthcare ("naics1" = 6) and Other Services ("naics1" = 8) achieve the lowest ambiguity across three episodes. Mining and Construction ("naics1" = 2) has low ambiguity in the pre dot com period, while the ambiguity for Education and Healthcare sector ("naics1" = 6) gradually goes up. Table 8 shows that the highest three deciles of book-to-market has the lowest ambiguity over the three episodes.

Table 3: Heterogeneous PEM Performance for System States

variable		full sample		pre-dotcom		dotcom-GFC		post-GFC	
		mean	std	mean	std	mean	std	mean	std
log_at	low	2.95%	9.34%	3.33%	9.89%	2.96%	9.07%	2.66%	9.04%
	high	1.60%	6.28%	1.82%	6.58%	1.79%	6.70%	1.32%	5.75%
log_act	low	4.04%	11.33%	4.31%	11.33%	3.78%	10.83%	3.99%	11.62%
	high	2.76%	8.20%	2.68%	7.62%	2.88%	8.36%	2.76%	8.54%
log_sale	low	6.92%	16.94%	6.56%	15.42%	6.30%	15.35%	7.56%	18.80%
	high	3.17%	10.00%	3.33%	10.01%	3.10%	9.94%	3.09%	10.03%
log_ap	low	6.39%	13.21%	5.22%	11.08%	6.14%	12.65%	7.42%	14.82%
	high	5.43%	12.57%	6.25%	13.54%	5.74%	13.06%	4.61%	11.40%
log_cogs	low	4.12%	12.40%	4.18%	12.10%	4.01%	12.25%	4.14%	12.70%
	high	2.75%	10.55%	3.09%	11.64%	2.66%	10.25%	2.53%	9.82%
log_intpn	low	1.84%	7.63%	1.45%	6.62%	1.89%	8.18%	2.11%	8.00%
	high	1.72%	5.59%	1.25%	4.69%	1.79%	6.05%	2.03%	5.90%
log_invt	low	4.85%	14.89%	4.82%	14.28%	4.58%	14.40%	5.03%	15.61%
	high	5.06%	12.85%	5.48%	13.68%	5.39%	13.40%	4.55%	11.82%
log_lct	low	6.55%	15.11%	6.52%	14.43%	6.40%	14.87%	6.66%	15.74%
	high	5.23%	13.24%	5.65%	13.76%	5.46%	13.75%	4.78%	12.50%
log_rect	low	6.87%	16.00%	5.92%	14.21%	6.60%	15.24%	7.76%	17.58%
	high	3.29%	9.37%	3.47%	9.61%	3.41%	9.43%	3.07%	9.15%
log_revt	low	6.90%	16.93%	6.56%	15.46%	6.30%	15.39%	7.52%	18.74%
	high	3.16%	9.99%	3.34%	10.13%	3.07%	9.89%	3.07%	9.94%
log_market_cap	low	12.53%	21.42%	12.99%	21.63%	13.62%	22.77%	11.55%	20.38%
	high	7.68%	15.70%	9.92%	18.70%	8.30%	16.43%	5.55%	11.96%
log_enterprise_val	low	10.34%	18.96%	11.53%	20.69%	11.55%	20.10%	8.73%	16.65%
	high	6.06%	13.67%	8.59%	17.98%	6.14%	12.80%	4.05%	9.24%
VOL	low	7.74%	14.11%	8.83%	14.05%	8.21%	14.96%	6.65%	13.57%
	high	4.01%	8.49%	4.89%	8.76%	3.68%	8.21%	3.52%	8.38%
RET	low	6.77%	14.92%	7.48%	15.94%	7.51%	15.95%	5.79%	13.37%
	high	5.05%	10.92%	5.88%	12.14%	5.43%	11.49%	4.19%	9.41%
macro1	low	4.91%	5.78%	4.62%	5.33%	4.63%	5.37%	4.97%	6.11%
	high	6.00%	6.71%	6.42%	6.80%	5.99%	6.54%	4.84%	5.88%
macro2	low	4.70%	5.53%	4.74%	5.47%	4.44%	5.18%	4.81%	5.89%
	high	5.86%	6.64%	6.59%	6.90%	6.26%	6.70%	5.01%	6.10%
macro3	low	5.05%	5.91%	5.64%	6.28%	5.14%	5.81%	4.55%	5.74%
	high	5.43%	6.28%	5.43%	6.10%	5.53%	6.26%	5.26%	6.22%
macro4	low	4.82%	5.83%	4.58%	5.33%	5.47%	6.09%	4.77%	5.75%
	high	5.75%	6.36%	6.35%	6.74%	5.16%	5.96%	5.06%	6.26%

Note: This table shows the heterogeneous PEM performance for each state variable. For each cross section, we divide firms in low and high groups for each state variable, and we calculate mean and standard deviation of PEM MSE for the focal state variable within three subsamples: pre dot com, between dot com and GFC, and post GFC.

Table 4: Heterogeneous PEM Performance for Sectors

MSE	full sample		pre-dotcom		dotcom-GFC		post-GFC	
sector	mean	std	mean	std	mean	std	mean	std
agriculture	7.00%	7.03%	7.40%	7.59%	7.04%	6.58%	6.62%	6.74%
mining & construction	6.84%	7.00%	6.20%	6.32%	7.23%	7.20%	7.10%	7.30%
manufacturing	4.95%	5.93%	4.98%	5.77%	5.17%	6.00%	4.78%	6.02%
trade & transportation	4.35%	5.17%	4.60%	5.36%	4.33%	5.09%	4.15%	5.05%
information & professional services	5.14%	5.92%	6.28%	6.76%	5.18%	5.71%	4.24%	5.15%
education & healthcare	4.37%	5.22%	4.99%	5.57%	4.05%	4.76%	4.02%	5.11%
recreation & accommodation	4.76%	5.44%	5.09%	5.22%	4.66%	5.03%	4.52%	5.85%
other services	3.89%	4.66%	4.27%	5.05%	3.52%	4.07%	3.60%	4.39%
public administration	7.00%	7.66%	7.52%	7.99%	7.09%	7.63%	5.05%	6.15%

Note: This table shows the heterogeneous PEM performance for each sector, defined by the first digit of North American Industry Classification System (NAICS) code. For each firm-quarter observation, we calculate average MSE of PEM for state variables to get the average MSE of PEM for each firm-quarter observation, and then for each cross section, we calculate mean and standard deviation of PEM MSE within three subsamples: pre dot com, between dot com and GFC, and post GFC, and we take the average across state variables. The top three sectors (with the lowest MSEs) within each time period is marked in red.

Table 5: Heterogeneous PEM Performance for "Value" Deciles

MSE	full sample		pre-dotcom		dotcom-GFC		post-GFC	
book-to-market decile	mean	std	mean	std	mean	std	mean	std
1	6.45%	6.98%	7.40%	7.34%	6.41%	6.77%	5.72%	6.73%
2	5.68%	6.45%	6.53%	6.94%	5.55%	6.07%	5.08%	6.19%
3	5.19%	6.10%	5.89%	6.51%	5.20%	5.95%	4.64%	5.78%
4	4.75%	5.72%	5.41%	6.13%	4.88%	5.63%	4.16%	5.38%
5	4.68%	5.65%	5.14%	5.91%	4.77%	5.56%	4.27%	5.48%
6	4.68%	5.60%	4.96%	5.72%	4.95%	5.69%	4.31%	5.43%
7	4.72%	5.66%	4.86%	5.52%	4.96%	5.78%	4.47%	5.69%
8	4.82%	5.70%	4.78%	5.37%	5.13%	5.93%	4.67%	5.80%
9	5.13%	5.85%	4.97%	5.63%	5.29%	5.99%	5.15%	5.92%
10	6.03%	6.55%	5.37%	5.98%	6.06%	6.59%	6.50%	6.88%

Note: This table shows the heterogeneous PEM performance for each book-to-market decile. For each firm-quarter observation, we calculate average MSE of PEM for state variables to get the average MSE of PEM for each firm-quarter observation, and then for each cross section, we calculate mean and standard deviation of PEM MSE within three subsamples: pre dot com, between dot com and GFC, and post GFC, and we take the average across state variables. The top three deciles (with the lowest MSEs) within each time period is marked in red.

Table 6: PEM: Heterogeneous Ambiguity for System States

variable		full sample		pre-dotcom		dotcom-GFC		post-GFC	
		mean	std	mean	std	mean	std	mean	std
log_at	low	7.96%	3.84%	7.04%	3.19%	7.78%	3.57%	8.76%	4.24%
	high	9.35%	4.19%	8.45%	3.85%	9.56%	4.17%	9.93%	4.33%
log_act	low	9.56%	4.59%	8.07%	3.62%	9.57%	4.31%	10.68%	5.07%
	high	10.68%	4.74%	8.99%	3.95%	10.85%	4.68%	11.90%	4.93%
log_sale	low	8.81%	4.85%	7.13%	3.25%	8.79%	4.46%	10.10%	5.63%
	high	9.15%	4.35%	7.66%	3.48%	9.44%	4.31%	10.13%	4.66%
log_ap	low	8.65%	4.38%	7.01%	3.23%	8.33%	3.96%	10.07%	4.88%
	high	9.51%	4.34%	7.84%	3.52%	9.60%	4.19%	10.74%	4.58%
log_cogs	low	9.07%	4.25%	8.07%	3.61%	9.10%	4.02%	9.82%	4.64%
	high	9.52%	4.30%	8.50%	3.92%	9.90%	4.36%	10.09%	4.40%
log_intpn	low	6.96%	3.47%	6.18%	2.85%	7.26%	3.53%	7.40%	3.77%
	high	7.43%	3.64%	6.11%	2.82%	7.58%	3.65%	8.33%	3.88%
log_invt	low	9.10%	4.32%	7.82%	3.57%	9.21%	4.12%	10.00%	4.71%
	high	8.87%	4.02%	7.51%	3.35%	9.39%	4.11%	9.60%	4.18%
log_lct	low	10.31%	5.00%	8.77%	4.03%	10.22%	4.99%	11.52%	5.34%
	high	11.21%	4.93%	9.49%	4.26%	11.51%	5.06%	12.37%	4.96%
log_rect	low	9.08%	4.34%	7.69%	3.39%	9.12%	3.91%	10.10%	4.89%
	high	9.24%	3.96%	7.99%	3.44%	9.64%	3.95%	9.96%	4.11%
log_revt	low	8.69%	4.87%	7.03%	3.30%	8.71%	4.44%	9.95%	5.68%
	high	9.00%	4.36%	7.52%	3.48%	9.24%	4.31%	10.00%	4.67%
log_market_cap	low	12.96%	8.25%	9.56%	4.60%	15.10%	9.74%	14.24%	8.63%
	high	13.33%	6.97%	11.10%	5.06%	15.98%	8.97%	13.50%	6.31%
log_enterprise_val	low	11.81%	7.66%	8.57%	4.13%	13.69%	8.70%	13.15%	8.26%
	high	12.39%	6.32%	10.08%	4.63%	14.63%	7.83%	12.86%	5.87%
VOL	low	9.50%	4.46%	7.98%	3.48%	10.40%	5.21%	10.10%	4.33%
	high	11.07%	4.64%	9.77%	4.07%	11.98%	5.18%	11.55%	4.51%
RET	low	11.62%	6.57%	8.92%	4.23%	13.57%	8.11%	12.53%	6.36%
	high	11.13%	5.86%	8.93%	4.24%	12.94%	7.11%	11.74%	5.60%
macro1	low	9.47%	3.72%	7.53%	2.60%	9.44%	3.51%	10.82%	4.04%
	high	10.82%	4.17%	8.85%	3.28%	11.45%	4.33%	10.68%	3.93%
macro2	low	9.29%	3.66%	7.64%	2.68%	9.79%	3.59%	10.70%	3.93%
	high	10.55%	4.07%	8.95%	3.31%	11.18%	4.43%	10.81%	4.03%
macro3	low	9.37%	3.65%	8.26%	3.08%	9.69%	3.49%	10.62%	3.94%
	high	10.46%	4.12%	8.14%	2.99%	11.33%	4.50%	10.88%	4.01%
macro4	low	9.91%	3.94%	7.48%	2.56%	9.56%	3.75%	10.89%	3.93%
	high	9.78%	3.85%	8.81%	3.27%	11.49%	4.19%	10.59%	4.04%

Note: This table shows the heterogeneous ambiguity for each state variable measured by the greatest different in predictions among PEM and its auxiliary models for that specific state variable. For each cross section, we divide firms in low and high groups for each state variable, and we calculate mean and standard deviation of ambiguity for the focal state variable within three subsamples: pre dot com, between dot com and GFC, and post GFC.

Table 7: PEM: Heterogeneous Ambiguity for Sectors

Ambiguity	full sample		pre-dotcom		dotcom-GFC		post-GFC	
sector	mean	std	mean	std	mean	std	mean	std
agriculture	10.77%	4.04%	9.59%	3.48%	11.55%	4.33%	11.40%	4.08%
mining & construction	10.55%	4.02%	8.45%	2.76%	11.72%	4.39%	11.44%	4.02%
manufacturing	9.42%	3.94%	7.70%	2.90%	10.10%	4.04%	10.49%	4.15%
trade & transportation	9.63%	3.76%	7.73%	2.65%	10.35%	4.04%	10.84%	3.77%
information & professional services	10.07%	3.70%	8.82%	3.28%	10.53%	3.89%	10.73%	3.65%
education & healthcare	9.21%	3.35%	7.88%	2.70%	9.81%	3.49%	9.99%	3.42%
recreation & accommodation	9.81%	3.65%	8.35%	2.78%	10.41%	3.81%	10.79%	3.82%
other services	8.62%	3.22%	7.22%	2.55%	9.42%	3.22%	10.19%	3.24%
public administration	9.81%	3.83%	8.76%	3.24%	10.66%	4.26%	11.84%	3.67%

Note: This table shows the heterogeneous ambiguity for each sector, defined by the first digit of North American Industry Classification System (NAICS) code. The ambiguity is measured by the greatest different in predictions among PEM and its auxiliary models for each state variable. For each firm-quarter observation, we calculate average ambiguity of PEM for state variables to get the average ambiguity of PEM for each firm-quarter observation, and then for each cross section, we calculate mean and standard deviation of PEM ambiguity within three subsamples: pre dot com, between dot com and GFC, and post GFC, and we take the average across state variables. The three sectors with the lowest ambiguity within each time period is marked in red.

Table 8: PEM: Heterogeneous Ambiguity for "Value" Deciles

Ambiguity	full sample		pre-dotcom		dotcom-GFC		post-GFC	
book-to-market decile	mean	std	mean	std	mean	std	mean	std
1	11.50%	4.30%	9.86%	3.40%	11.99%	4.38%	12.51%	4.50%
2	10.56%	3.98%	8.97%	3.09%	11.11%	4.10%	11.50%	4.14%
3	10.17%	3.92%	8.55%	3.08%	10.78%	4.13%	11.07%	3.98%
4	9.90%	3.78%	8.43%	3.05%	10.37%	3.94%	10.79%	3.86%
5	9.79%	3.74%	8.21%	2.99%	10.48%	3.96%	10.59%	3.74%
6	9.55%	3.69%	8.01%	2.92%	10.29%	3.98%	10.30%	3.67%
7	9.40%	3.66%	7.83%	2.86%	10.18%	3.93%	10.11%	3.67%
8	9.36%	3.74%	7.69%	2.81%	10.07%	3.95%	10.20%	3.81%
9	9.27%	3.73%	7.40%	2.59%	9.85%	3.95%	10.29%	3.80%
10	8.96%	3.75%	6.94%	2.43%	9.38%	3.81%	10.16%	3.89%

Note: This table shows the heterogeneous ambiguity for each book-to-market decile. The ambiguity is measured by the greatest different in predictions among PEM and its auxiliary models for each state variable. For each firm-quarter observation, we calculate average ambiguity of PEM for state variables to get the average ambiguity of PEM for each firm-quarter observation, and then for each cross section, we calculate mean and standard deviation of PEM ambiguity within three subsamples: pre dot com, between dot com and GFC, and post GFC, and we take the average across state variables. The three deciles with the lowest ambiguity within each time period is marked in red.

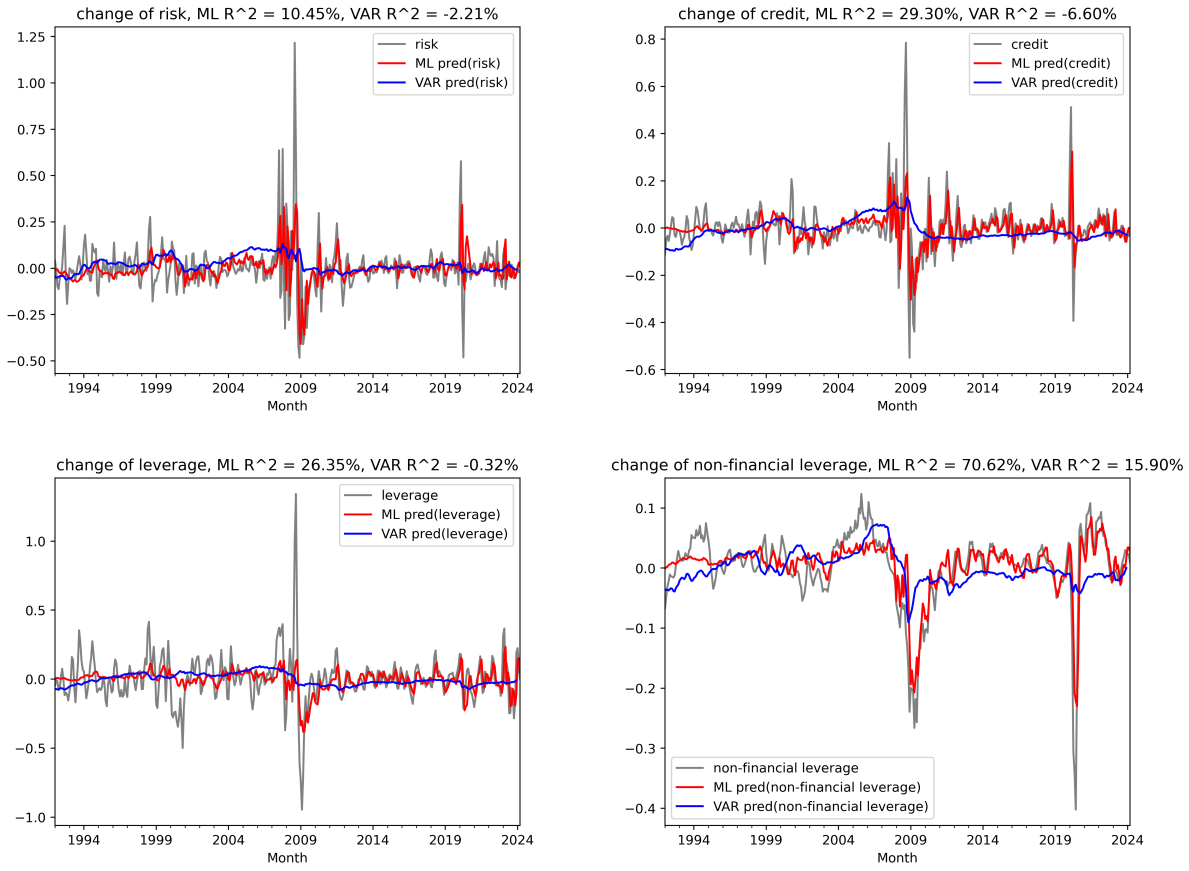


Figure 4: Predicting macroeconomic variables using neural network and VAR

4.3 Predicting the Evolution of Macroeconomic States

Macroeconomic variables are treated differently from fundamentals and stock market reactions because these variables are not likely influenced by individual firms' state or decisions. For instance, if a representative firm picks a different leverage by issuing more equity or debt, this counterfactual is not likely to have an impact on macroeconomic non-financial leverage or macroeconomic credit risk. Thus, we eliminate the connection in PEM linking firm-specific fundamentals or stock market variables to macroeconomic states. Given that macroeconomic variables do not have cross-sectional variation and are co-integrated, they are hard to predict precisely. Also, their information content is overlapping. As such, we choose NFCI subindices, including risk, credit, financial leverage and non-financial leverage, to summarize macroeconomic conditions in our study.

Figure 4 shows the predicted change in next month for macroeconomic state variables using current and one-month lagged macroeconomic states generated by a neural network

model and a VAR model. When using neural network, current growth rates of macroeconomic state variables are also added as additional inputs, given the capability of neural network models handling colinear variables. VAR models are generally used in time series modelling, which are widely used to model macroeconomic variables. In our setting, the neural network model is comparable to the VAR model because they are using the same information set as well as the same training paradigm when predicting the macroeconomic growth. Consistent with PEM, we update the VAR model every year using all available observations till that year. We find that using simple VAR models to predict these four time series generates much lower (or even negative) out-of-sample R^2 than PEM.

4.4 An Illustration of Firm Recapitalization

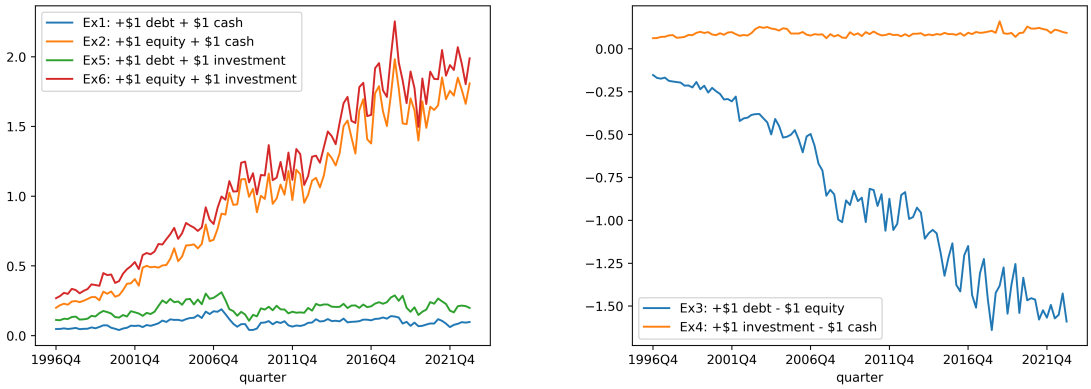
Even though PEM is high-dimensional, we can analyze low dimensional action combinations, which are the focus of conventional reduced-form models. Firm recapitalization that shows up in every corporate finance textbook serves as an excellent illustration. Specifically, we ask how the enterprise value changes when a manager (firm):

(1) raises \$1 in debt and put that \$1 into its cash savings
(2) raises \$1 in equity and put that \$1 into its cash savings
(3) raises \$1 in debt and \$1 less equity
(4) puts \$1 cash into investment
(5) raises \$1 in debt and put that \$1 into investment
(6) raises \$1 in equity and put that \$1 into investment

We observe several patterns. First, Exercise 4 simply corresponds to the return on the asset investment and is only marginally above zero. Second, in Exercise 3: changing capital structure while keeping the same enterprise value reduces enterprise value, because in general leverage is believed to be too high (firms should decrease their debt proportion). Third, Exercises 1, 2, 5, and 6 reveal that debt financing is not very helpful for increasing enterprise value as equity financing is.

Exercises 1-3 are more related to capital structure whereas 4-6 are about financing investments. Overall, the best recapitalization policy for increasing enterprise value is Exercise 6. Notably, the dispersion in the performances of the six recapitalization policies grew bigger over time.

Figure 5: Recapitalization Exercises: Capital Structure and Financing Investments



Note: This figure shows the recapitalization exercises where counterfactuals are generated by PEM. The x-axis is the calendar year, and the y-axis is dollar change in enterprise value. Solid lines plot the median value of the change in enterprise value for each cross section, and their legends explain the content of the recapitalization exercises.

5 Optimal Policy Recommendations to Firm Managers

5.1 Reinforcement Learning and the Policy Module

The policy module, built on the predictive environment module, intends to seek optimal managerial decision-making processes given a certain utility function (i.e., objective) of the manager. With the assumption that managers make decisions based on the current state of the firm, a neural network could parameterize the decision-making process as a function of current states: $u_t = g(X_t)$. To mitigate the uncertainty of predictive environment module under unusual decisions, we punish directly the boosting error defined in Equation (4) and train a model based on the discrete version of utility function in Equation (5). Moreover, to make the proposed algorithmic decisions meaningful for managers to learn from, we initialize our policy module by supervised learning to get started with the real managerial decision-making process.[16]

In our initialization phase, we start from a randomly initialized neural networks whose inputs are current states and outputs are current decisions to be made. We optimize an exogenously specified objective function using RL. The value of the objective function is

[16]RL need not rely on supervised labeling. It has a different functionality than supervised learning, which intends to minimize the loss between label and prediction. Here, as there may be lots of learnable optima and RL may get stuck at any of them, the initialization is very important to determine which optima to arrive at. We choose to use supervised learning to make the policy module approximate the real decisions first, and then let RL marginally improve the policy module to get higher utility till convergence.

given by the counterfactual statements from environment module. For each epoch, the policy module first gives out a set of decisions to be made, then together with state variables, these decisions are fed into the environment module to generate counterfactual market reactions in the next quarter. Finally, the difference between counterfactual market capitalization (alt., enterprise value) and current market capitalization (enterprise value) is treated as the objective function value for this trait. After the feed-forward process is complete, backward propagation takes place and the parameters of the policy module are updated, given the loss function to be negative objective value.

We consider a set of reasonable objective functions and also a spectrum of horizons for managers' objectives. Specifically, we consider both short-term (1 quarter) and long-term (2 years) market capitalization increases and enterprise value increases as our objectives. We ignore agency problems and take managers as maximizing shareholders' value or overall investors' value. The DDRC framework is flexible to allow other objectives and these two are useful benchmarks to consider.

5.2 Performance Under Various Managerial Objectives

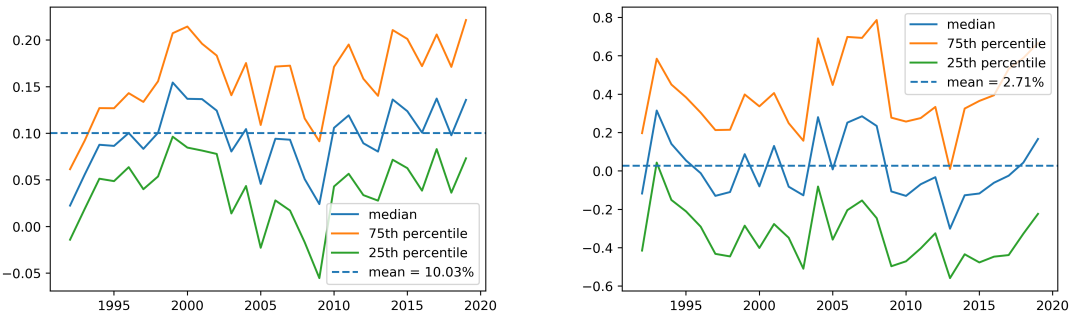
We report the performance of AlphaManager under the various objectives (and horizons).

Long-term objectives maximizing valuation growth in 2 years. To outperform in the longer run, AlphaManager needs to have a longer horizon in its objective. We therefore examine how AlphaManager performs if the object is to maximize increases in the market capitalization or enterprise value over the subsequent 8 quarters instead of just over the next quarter. For long-term model with capitalization growth as the objective function, the average quarterly return is 8.73% while the first quarter performance is on average only 3.09%. When using enterprise value growth as the objective function, these two numbers become 4.43% and 1.28%, respectively.

Short-term objectives centering on next quarter valuation growth. We then consider a somewhat myopic objective of increasing market equity valuation over the next quarter. We estimate the corresponding short-term ambiguity constraint parameters under optimal long-term AM agents to make the results comparable. The overall average out-performance is 10.03% quarterly in the short-term when using market capitalization growth as the objective. When the enterprise value growth serves as the objective, the out-

performance becomes 2.71%. The cross-sectional dispersion is non-trivial, as seen in the 75th and 25th percentile firms in Figure 6.

Figure 6: Out-performance of AlphaManager with Short-term Objectives



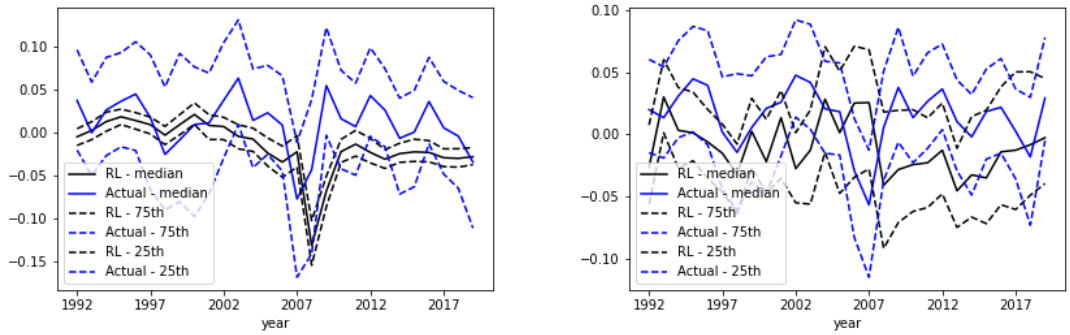
Note: This figure shows the out-performance of AlphaManager trained with short-term market cap growth (left) and enterprise value growth (right) as objective functions. The solid blue line shows the median performance, the orange line and the green line show the 75th and 25th percentiles respectively, and the dotted blue line shows the median for each year. The y-axis is in percentage of the change in objective for the next quarter predicted by the PEM.

Figure 7 depicts what myopic behavior (a focus on next quarter outcomes) implies for the long-term performance of the firms. This relationship is tracked over 30 years of data. *Ex-ante*, several outcomes could appear. On the one hand, one could argue that managers historically may not be as myopic, and as such, historical actions may generate better long-term performance if short-termism hurts firm fundamentals in the intermediate or long run. On the other hand, even with potentially longer-term goals, managers could have taken very suboptimal actions historically. The 1-quarter-market-cap-focused myopic AlphaManager can end up perform much better. In the data, we observe that though AlphaManager outperforms under the specified objective, over the intermediate term (i.e., for a different objective with longer horizon) it underperforms. Algorithmic predictions don't capture cross-sectional "risk" (in-model volatility), so the algorithmic predictions (black lines) have much smaller variations.

Figure 8 shows the term structure for short-term and long-term AM under the objective of market cap growth. In the left panel, the short-term AM is out performing the long-term AM in longer terms' rewards, but this out performance is at the cost of high ambiguity. In the right panel, we adjust reward (i.e., market cap growth) using the formula:

$$\text{Adj. Reward}_t = \frac{1 + \text{Reward}_t}{\sqrt{\max\{1, \frac{\text{BoostingError}_t}{\text{BoostingError}_{t_0}}\}}} - 1, \tag{8}$$

Figure 7: Long-Term Performance Under Short-Termism



Note: This figure shows the long-term performance of AlphaManager trained with short-term market cap growth (left) and enterprise value growth (right) as objective functions. The solid blue line shows the median average performance of actual managerial decisions, while the solid black line shows the median performance of actions suggested by the RL module. Dashed lines show the 25th and 75th percentiles.

where Reward_t is the reward that AM achieves under the PEM at time t, BoostingError_t is the boosting error for the AM-suggested managerial decisions at time t, and $\text{BoostingError}_{t_0}$ is the benchmark boosting error at time t_0. When the ambiguity is as low as the benchmark ambiguity, there is no punishment on the reward; the adjusted reward is decreasing in ambiguity of AM. After the adjustment, even though the short-term AM outperforms in the short-term, the long-run performance is worse than the long-term AM, and further into the future, the gap between adjusted rewards from short-term AM and long-term AM is also wider.

Table 9 reports heterogeneous predicted performance under the objective of 2-year enterprise value growth for the RL-recommended decisions by each state's cross-sectional ranking. The AM's impressive performance is mainly driven by small and illiquid firms during high-risk episodes. The pre dot com period enjoys the highest performance of AM. Table 10 documents the heterogeneous performance of RL. Manufacturing sector has high performance across three episodes. Table 8 shows the heterogeneity by book-to-market deciles. The best performers are gradually shifting down as time goes by.

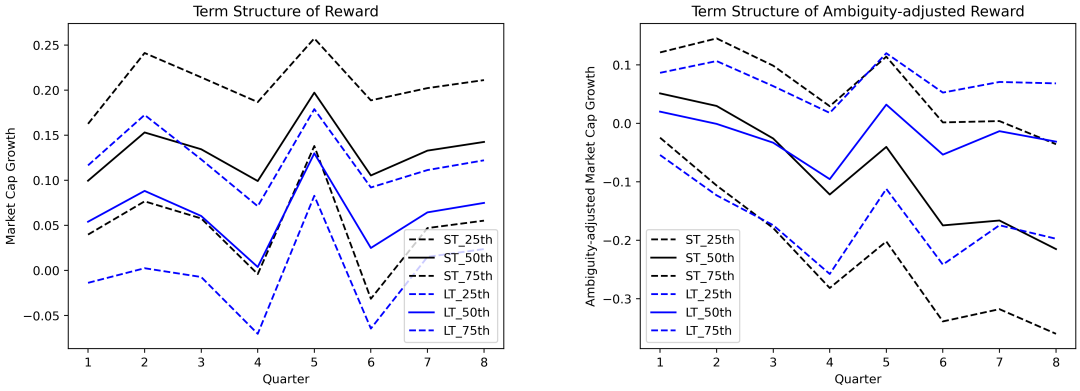
The analysis shows that AlphaManager again outperforms historical actions and there is heterogeneity in the cross-section. The average out-performance is much lower, reflecting that over the longer horizon, we just cannot predict much. Notably, these patterns also emerge in papers based on CEO surveys (e.g., Graham, 2022).

Table 9: Heterogeneous Performance of AlphaManager Across Firms

variable		full sample		pre-dotcom		dotcom-GFC		post-GFC	
		mean	std	mean	std	mean	std	mean	std
log_at	low	8.11%	3.73%	8.13%	4.00%	6.48%	4.74%	6.67%	4.92%
	high	1.99%	4.62%	4.54%	4.02%	0.74%	5.07%	0.17%	3.50%
log_act	low	10.13%	4.73%	6.75%	4.35%	5.14%	5.63%	4.96%	6.41%
	high	3.16%	4.68%	5.30%	4.29%	2.18%	5.38%	2.04%	3.74%
log_sale	low	8.80%	4.24%	7.98%	3.87%	6.18%	4.88%	6.08%	5.48%
	high	2.25%	4.68%	4.37%	4.11%	1.15%	5.31%	1.00%	3.93%
log_ap	low	8.40%	3.96%	7.66%	4.30%	5.69%	5.29%	5.97%	5.45%
	high	2.52%	4.69%	4.83%	4.05%	1.43%	5.27%	0.82%	3.81%
log_cogs	low	8.64%	4.06%	7.81%	3.89%	6.02%	4.90%	5.82%	5.56%
	high	2.38%	4.77%	4.45%	4.19%	1.27%	5.42%	1.22%	4.09%
log_intpn	low	7.11%	3.63%	6.10%	4.09%	4.58%	5.33%	4.65%	5.70%
	high	3.29%	5.34%	5.91%	4.82%	2.31%	5.87%	2.04%	4.60%
log_invt	low	8.22%	3.84%	6.81%	4.29%	3.89%	5.64%	4.40%	5.89%
	high	3.68%	5.01%	5.32%	4.35%	3.25%	5.72%	2.46%	4.62%
log_lct	low	10.78%	4.75%	6.79%	4.35%	5.37%	5.51%	4.95%	6.35%
	high	3.03%	4.67%	5.21%	4.27%	1.89%	5.34%	1.99%	3.76%
log_rect	low	8.12%	3.77%	7.70%	4.20%	5.82%	5.22%	6.08%	5.35%
	high	2.43%	4.71%	4.83%	4.11%	1.25%	5.20%	0.65%	3.73%
log_revt	low	8.97%	4.21%	7.15%	4.08%	6.00%	4.88%	5.91%	5.61%
	high	2.46%	4.88%	4.87%	4.39%	1.35%	5.47%	1.19%	3.99%
log_market_cap	low	14.14%	9.11%	8.35%	3.65%	6.35%	4.97%	6.27%	5.37%
	high	2.15%	4.57%	4.18%	4.02%	1.16%	5.15%	0.88%	3.89%
log_enterprise_val	low	12.30%	7.92%	8.42%	3.80%	6.63%	4.70%	6.68%	4.98%
	high	1.93%	4.53%	4.31%	3.95%	0.74%	5.02%	0.30%	3.57%
VOL	low	10.77%	4.68%	8.02%	3.63%	5.59%	5.16%	5.23%	5.30%
	high	2.42%	4.92%	3.88%	4.11%	1.71%	5.52%	1.66%	4.81%
RET	low	11.16%	7.26%	6.35%	4.40%	4.09%	5.81%	3.93%	5.57%
	high	3.94%	5.15%	5.74%	4.35%	3.07%	5.53%	2.89%	5.10%
macro1	low	4.36%	5.24%	7.04%	4.31%	4.12%	5.13%	3.46%	5.43%
	high	4.37%	5.34%	4.99%	4.22%	3.05%	6.12%	3.35%	5.31%
macro2	low	4.35%	5.26%	6.37%	4.48%	4.20%	5.18%	3.00%	5.14%
	high	4.38%	5.31%	5.59%	4.21%	2.93%	6.08%	3.96%	5.62%
macro3	low	4.35%	5.20%	5.74%	4.35%	3.98%	4.73%	3.11%	5.36%
	high	4.38%	5.38%	6.41%	4.40%	3.10%	6.56%	3.71%	5.35%
macro4	low	4.26%	5.31%	7.26%	4.27%	4.38%	4.83%	2.88%	4.78%
	high	4.51%	5.24%	4.89%	4.17%	2.61%	6.42%	3.80%	5.74%

Note: This table shows the heterogeneous performance under RL-recommended decisions by each state variable. For each cross section, we divide firms in low and high groups for each state variable, and we calculate mean and standard deviation of the performance of RL-recommended decisions within three subsamples: pre dot com, between dot com and GFC, and post GFC.

Figure 8: Term Structure of Reward and Ambiguity-adjusted Reward



Note: This figure shows the term structure for AlphaManager trained with short-term market cap growth with reward (left) and ambiguity-adjusted reward (right). The ambiguity adjustment follows Eq.(8). The solid blue lines show the median performance of long-term AM, while the solid black lines show the median performance of short-term AM. Dashed lines show the 25th and 75th percentiles.

Table 10: Heterogeneous Performance of AlphaManager Across Industries

RL performance (quarterly avg)	full sample		pre-dotcom		dotcom-GFC		post-GFC	
sector	mean	std	mean	std	mean	std	mean	std
agriculture	3.16%	4.62%	4.97%	4.71%	1.86%	4.44%	2.74%	4.28%
mining & construction	3.53%	5.07%	6.06%	4.10%	1.13%	5.50%	2.17%	4.53%
manufacturing	5.67%	5.19%	6.62%	4.63%	5.30%	5.48%	5.15%	5.32%
trade & transportation	3.55%	4.87%	5.68%	4.47%	2.31%	5.49%	2.70%	4.21%
information & professional services	3.56%	5.28%	5.61%	4.13%	2.72%	5.51%	2.00%	5.50%
education & healthcare	4.36%	5.23%	6.62%	4.72%	2.46%	5.83%	4.14%	4.57%
recreation & accommodation	3.80%	5.11%	5.86%	4.15%	1.74%	5.50%	3.71%	4.86%
other services	4.35%	5.14%	7.55%	4.35%	1.61%	5.17%	1.85%	2.89%
public administration	4.43%	5.71%	5.22%	5.40%	4.49%	5.24%	1.74%	6.70%

Note: This table shows the heterogeneous performance under RL-recommended decisions by each sector, defined by the first digit of North American Industry Classification System (NAICS) code. For each cross section, we calculate mean and standard deviation of the performance of RL-recommended decisions within three subsamples: pre dot com, between dot com and GFC, and post GFC. The top three sectors (with the highest average rewards) within each time period is marked in red.

Table 11: Heterogeneous Performance of AlphaManager Across Value Deciles

RL performance (quarterly avg)	full sample		pre-dotcom		dotcom-GFC		post-GFC	
book-to-market decile	mean	std	mean	std	mean	std	mean	std
1	3.48%	4.92%	3.85%	4.25%	2.58%	5.25%	3.84%	5.08%
2	4.65%	4.82%	5.53%	4.25%	4.05%	5.13%	4.40%	4.91%
3	4.99%	5.02%	6.31%	4.31%	4.48%	5.30%	4.33%	5.12%
4	5.09%	5.14%	6.65%	4.44%	4.35%	5.52%	4.38%	5.10%
5	5.08%	5.38%	6.82%	4.72%	4.44%	5.80%	4.21%	5.23%
6	4.57%	5.82%	6.95%	4.66%	4.07%	6.16%	3.05%	5.81%
7	4.48%	5.72%	6.84%	4.54%	3.24%	6.28%	3.17%	5.62%
8	4.06%	5.62%	6.22%	4.57%	2.72%	6.08%	2.88%	5.57%
9	3.88%	5.39%	6.16%	4.06%	2.73%	5.78%	1.58%	5.45%
10	3.52%	4.64%	5.50%	3.48%	2.60%	5.14%	0.85%	4.40%

Note: This table shows the heterogeneous performance under RL-recommended decisions by each book-to-market decile. For each cross section, we calculate mean and standard deviation of the performance of RL-recommended decisions within three subsamples: pre dot com, between dot com and GFC, and post GFC. The top three deciles (with the highest average rewards) within each time period is marked in red.

5.3 AI-Assisted Optimal Managerial Actions

Let us examine the managerial actions AlphaManager recommends under the four objectives discussed earlier. We plot the actual decisions (e.g. leverage) while the direct output of our model is the change of them (e.g. growth rate of leverage). If model-implied decision goes out of the bound (e.g., negative leverage), we set it to be the bound (e.g., 0).

Starting with the objective of maximizing next quarter's market capitalization, AlphaManager recommends (relative to what was historically done): (1) doing more acquisitions; (2) increasing cash holdings; (3) issuing less debt but more equity (reduce leverage); (4) paying out more dividend; and (5) increasing investment, especially in R&D. The recommendation also includes allowing for more variation in investments. The figure also suggests that a manager should carry out more repurchases in bad times.

Next, for an objective maximizing short-term enterprise value, AlphaManager makes similar recommendation, except that a manager does not need to reduce debt issuance; if anything, debt issuance should be slightly higher. Leverage seems beneficial in general, especially before the great financial crisis (GFC).

Moving to an objective of long-term maximization of market capitalization increases (8 quarters), a manager should do more acquisitions and hold more cash, especially for small firms. In general, it is helpful to issue slightly more debt. Issuing more equity helps, especially for small firms. Other recommendations include investing more, lowering leverage, increasing

Figure 9: Empirical and Optimal Decisions under 1-QTR Market Cap Growth

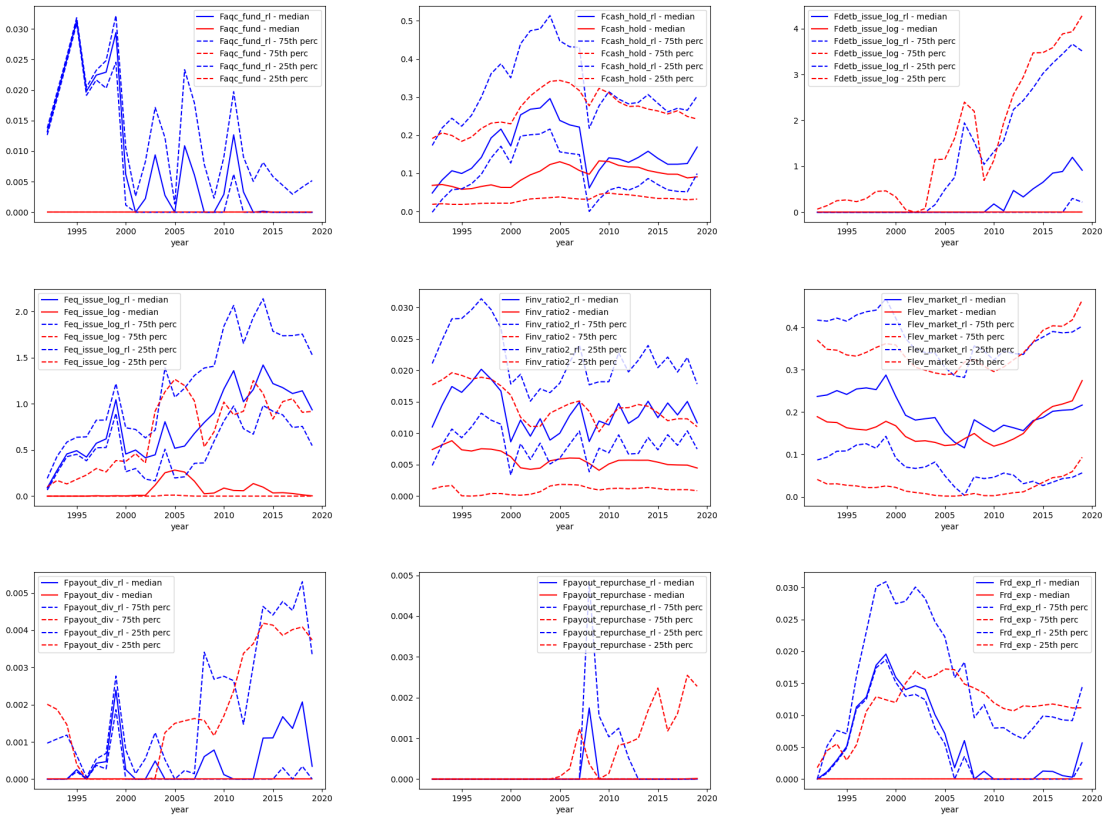


Note: This figure shows the empirical (red) and optimal decisions (blue) suggested by AM under 1-quarter market cap growth as the objective to maximize. We plot mean (in solid lines) along with 25th and 75th percentiles (in dashed lines) to represent cross-sectional variation. The decisions are acquisitions (Faqc_fund), cash holdings (Fcash_hold), debt issuance (Fdebt_issue_log), equity issuance (Fequity_issue_log), investment ratio (Finv_ratio2), leverage (Flev_market), dividend (Fpayout_div), repurchase (Fpayout_repurchase), and R&D expenses (Frd_exp), with formal definitions detailed in Appendix A.

dividend payouts, do more repurchases but less R&D.

Finally, for the objective of maximizing long-term (8 quarters) enterprise value, AlphaManager recommends doing more acquisitions, holding slightly more cash and issuing slightly more debt, issuing more equity and investing more, having slightly higer leverage during normal times but having lower leverage after the onset of GFC, paying slightly less dividends, making more repurchases but lowering R&D expenses.

Figure 10: Empirical and Optimal Decisions under 1-QTR Enterprise Value Growth



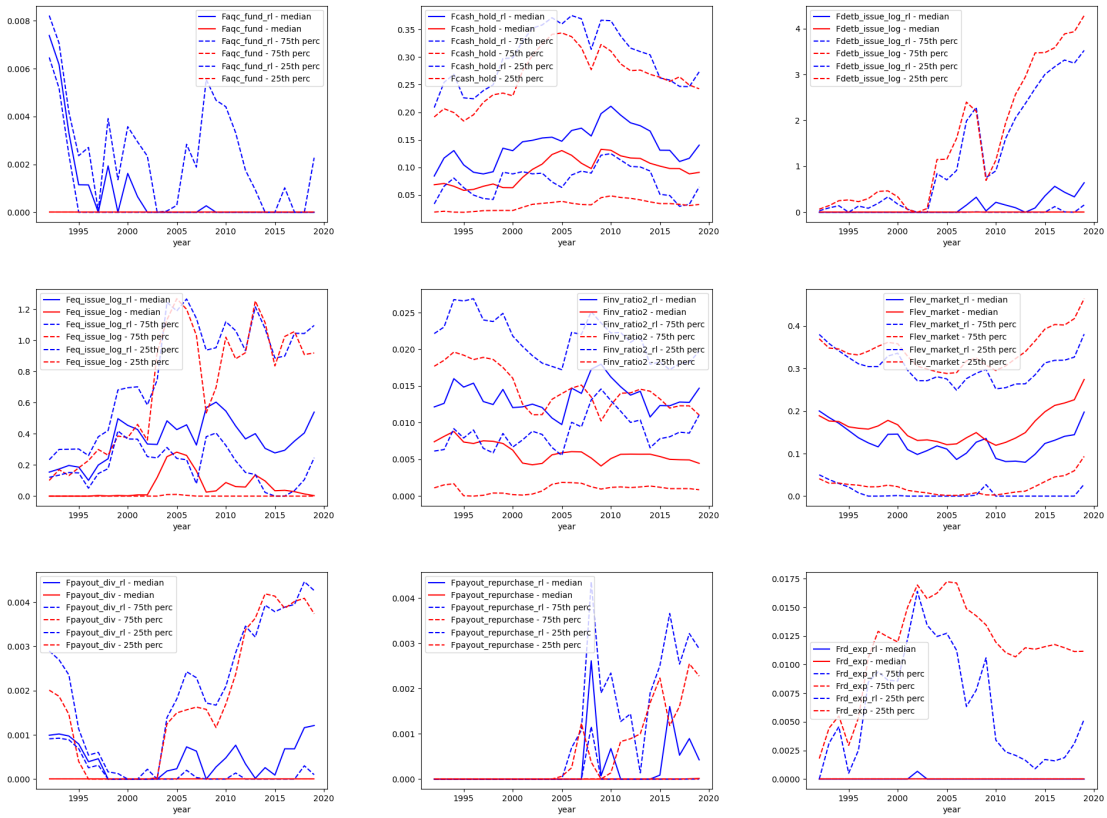
Note: This figure shows the empirical (red) and optimal decisions (blue) suggested by AM under 1-quarter enterprise value growth as the objective to maximize. We plot mean (in solid lines) along with 25th and 75th percentiles (in dashed lines) to represent cross-sectional variation. The decisions are acquisitions (Faqc_fund), cash holdings (Fcash_hold), debt issuance (Fdebt_issue_log), equity issuance (Fequity_issue_log), investment ratio (Finv_ratio2), leverage (Flev_market), dividend (Fpayout_div), repurchase (Fpayout_repurchase), and R&D expenses (Frd_exp), with formal definitions detailed in Appendix A.

6 Piecing Together Corporate Finance Research

As explained in the introduction, our DDRC approach focuses more on producing reliable empirical predictions and policy recommendatios. It does not serve as a substitute for theory, reduced-form models, or structural estimations, because it does not offer insights on the underlying economic mechanisms. The associated cost of the many connections AlphaManager establishes in the data is that they are sometimes difficult to rationalize — they lack the usual notion of *ex-ante* theory predictions or causality statements — and they can be computationally expensive.

Notably, however, both problems can be mitigated by incorporating theoretical guidance (multiple concurrent models). DDRC is still a useful approach for understanding the

Figure 11: Empirical and Optimal Decisions under 8-QTR Market Cap Growth

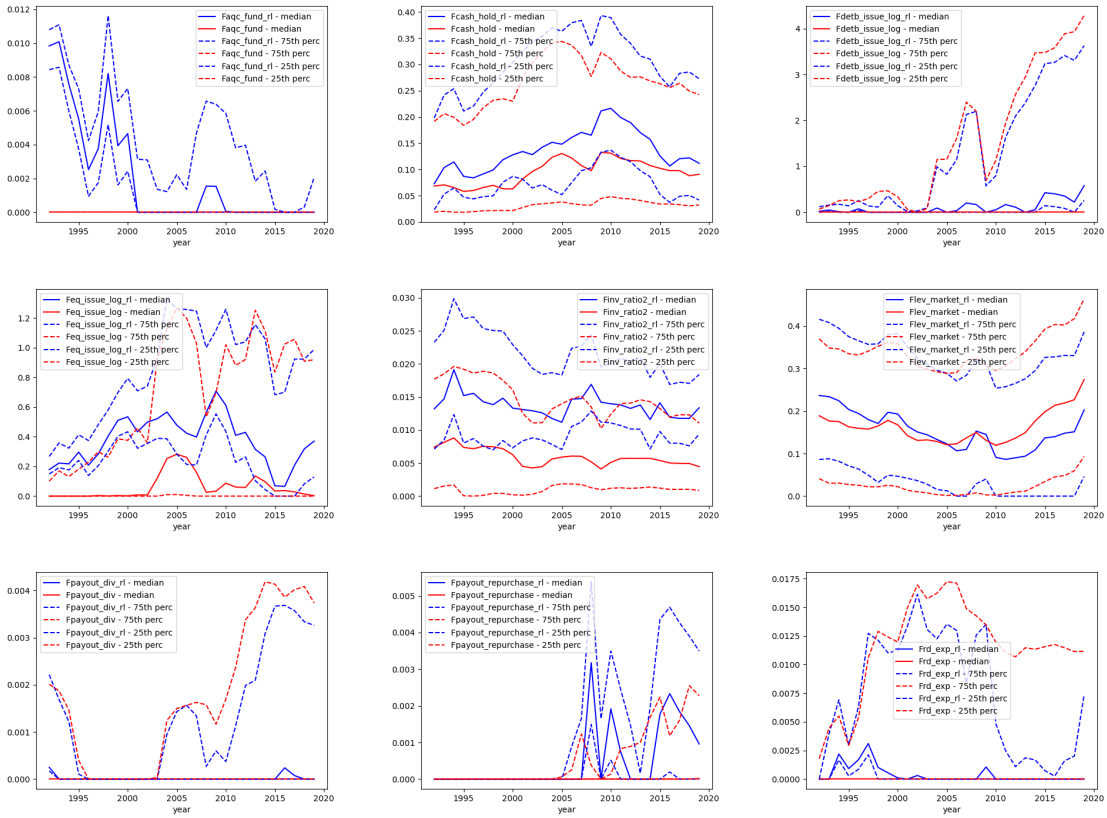


Note: This figure shows the empirical (red) and optimal decisions (blue) suggested by AM under 8-quarter market cap growth as the objective to maximize. We plot mean (in solid lines) along with 25th and 75th percentiles (in dashed lines) to represent cross-sectional variation. The decisions are acquisitions (Faqc_fund), cash holdings (Fcash_hold), debt issuance (Fdebt_issue_log), equity issuance (Fequity_issue_log), investment ratio (Finv_ratio2), leverage (Flev_market), dividend (Fpayout_div), repurchase (Fpayout_repurchase), and R&D expenses (Frd_exp), with formal definitions detailed in Appendix A.

underlying economic mechanisms by complementing other paradigms in corporate finance research. In a way, DDRC is a data-driven version of the structural approach that allows the incorporation of knowledge from theory and reduced-form models, as well as from financial big data. To wit, DDRC is not at odds — nor compete — with reduced causal exercises and theory-driven structural estimations. Instead, they give us guidance as to when we should use those other approaches to data analysis.

We next discuss how the measure of model ambiguity in AlphaManager reveals the types of corporate decisions, firms, and market environment that would benefit more from theoretical guidance or causal identifications, as well as the types for which big data and powerful algorithms suffice. We then discuss how to incorporate the insights from other approaches

Figure 12: Empirical and Optimal Decisions under 8-QTR Enterprise Value Growth



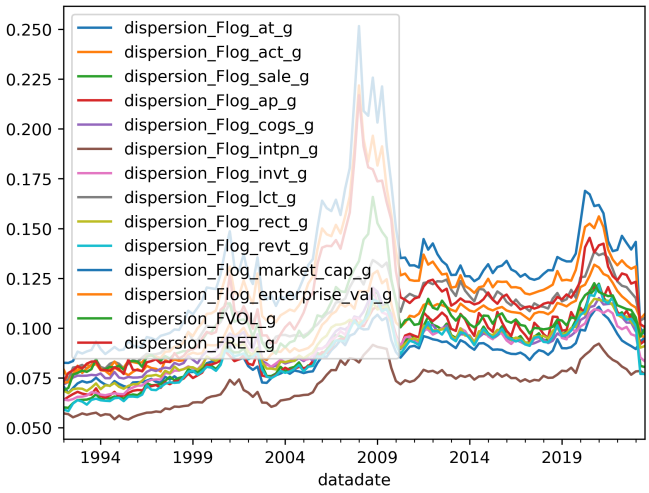
Note: This figure shows the empirical (red) and optimal decisions (blue) suggested by AM under 8-quarter enterprise value growth as the objective to maximize. We plot mean (in solid lines) along with 25th and 75th percentiles (in dashed lines) to represent cross-sectional variation. The decisions are acquisitions (Faqc_fund), cash holdings (Fcash_hold), debt issuance (Fdebt_issue_log), equity issuance (Fequity_issue_log), investment ratio (Finv_ratio2), leverage (Flev_market), dividend (Fpayout_div), repurchase (Fpayout_repurchase), and R&D expenses (Frd_exp), with formal definitions detailed in Appendix A.

into DDRC through ambiguity-guided transfer learning.

6.1 Ambiguity and the Need for Theory/Reduced-Form Models

Our measure of ambiguity or relative entropy among the boosted PEMs informs how much we can rely on the data-driven approach. First, when the dispersion in predictions is high, one has to go back to theory or reduced-form models (with or without causal identification) in order to fully explain or predict firm outcomes, or to make counterfactual recommendations. Second, our policy module will avoid exploring the policy space that causes more dispersed responses from PEMs. This causes the AlphaManager to be conservative in the offline learning at the cost of missing out exploring more profitable policy actions, the knowledge of

Figure 13: Average Ambiguity Time Series by State Variables



Note: This figure shows average estimated ambiguity time series for each state variable. We empirically estimate ambiguity using boosting error for each observation, and then aggregate at state-year level by taking averages.

which has to be learned through other approaches to corporate finance research. Figure 13 plots the estimated average ambiguity time series for each one of our state variables. During macroeconomic turning points, the ambiguity is estimated to be high.

The corporate finance literature has documented and studied a large collection of phenomena. Yet findings are typically dispersed. The DDRC framework, in addition to guiding managers in practice, also provides insights about what areas in corporate finance research deserve more emphasis and attention going forward. Along the dimensions that AlphaManager has greater ambiguity, theoretical insights and causal identification would be particularly informative and thus deserve more attention.

6.2 Ambiguity-Guided Transfer Learning from Other Models

Even after identifying the areas firms, and macroeconomic conditions where theory, reduced-form models, or structural estimations can be particularly informative, the question remains how one can incorporate the insights gained into a DDRC framework. A modified transfer learning technical comes to the rescue.

Transfer learning is a powerful machine learning technique that leverages the knowledge gained while solving one problem to solve a different but related problem. This approach is

particularly useful when dealing with scenarios where the data for the new task is scarce or when training a model from scratch would be computationally expensive. By transferring and adapting pre-trained models to new tasks, researchers and practitioners can achieve significant improvements in model performance with less data and in shorter training times. Transfer learning has found widespread applications across various domains, including natural language processing, computer vision, and speech recognition, demonstrating its versatility and efficiency in enhancing learning processes and outcomes in diverse settings.

In our setting, if one wants to incorporate any theoretical predictions (e.g., Hennessy and Whited, 2005; Bolton et al., 2011), or relationships identified in reduced-form models, or counterfactual predictions from structural estimations, one can use them to generate data to be added to the training sample for PEM. Note that using the historical data in reduced-form empirical models is unlikely to work. For example, a causal identification may concern only one particular firm and the effect would be too "weak" to be picked up in AlphaManager. Therefore, one has to necessarily generate more such cases by extending the reduced-form results to the cross section and broader ranges of the treatments, in order to generate sufficient observations.

Transfer learning is traditionally applied to one related dataset. Yet as we discuss earlier, we have many insights drawn from extant corporate finance research. How to incorporate them in a unified framework? Where to draw the line? That is where the ambiguity metric can be used to guide the weights we put on the data for the transfer learning. If a pattern is rarer or more ambiguously described by PEM, one should increase the weights of data concerning that pattern in the transfer learning.

7 Conclusion

For a typical managerial objective, corporate decision-making is a high-dimensional, nonlinear stochastic control problem in which the managers interact with the economic environment, learning about it while formulating an optimal dynamic policy. Existing approaches in corporate finance and stochastic control often prove inadequate for solving this problem, falling short of explaining or predicting empirical firm outcomes or getting adopted by managers. We introduce an AI-assisted, data-driven framework to complement them, with an emphasis on empirical performance and producing useful counterfactuals, so as to offer effective policy recommendations for a wide range of business objectives.

We first build a predictive environment module through supervised neural networks, and

then add a policy module through deep reinforcement learning that goes beyond hypothesis testing on historical data or simulations. By incorporating model ambiguity and robust control techniques, our framework not only better explains and predicts corporate outcomes in- and out-of-sample, but also identifies important managerial decisions while offering effective dynamic policies adaptive to market evolution and feedback. Critically, our proposed framework helps to distinguish scenarios where theory and causal identifications are crucial from situations where predictive models trained on historical observations suffice. It can also incorporate insights and predictions from theory, reduced-form model, and structural estimations through ambiguity-guided transfer learning. These flexibilities make DDRC a plausible candidate to focus researchers' effort and study corporate finance in a unified framework instead of fragmented attempts.

We believe that the DDRC approach to corporate finance opens the door for analyzing a number of important yet hitherto unexplored topics. For example, we have taken the managerial objective in this paper as given, and there are techniques for learning about the historical managerial objectives. The framework also provides insights on the heterogeneity in the effects of managerial actions, firm controlability, and their interactions with the macroeconomic environment. These constitute important topics for future research.

References

Andersen, Torben G, Nicola Fusari, and Viktor Todorov, "Parametric inference and dynamic state recovery from option panels," *Econometrica*, 2015, *83* (3), 1081–1145.

Barnett, Michael, William Brock, and Lars Peter Hansen, "Pricing uncertainty induced by climate change," *The Review of Financial Studies*, 2020, *33* (3), 1024–1066.

Bellstam, Gustaf, Sanjai Bhagat, and J Anthony Cookson, "A text-based analysis of corporate innovation," *Management Science*, 2020.

Bolton, Patrick, Hui Chen, and Neng Wang, "A unified theory of Tobin's q, corporate investment, financing, and risk management," *The journal of Finance*, 2011, *66* (5), 1545–1578.

Bond, Philip, Alex Edmans, and Itay Goldstein, "The real effects of financial markets," *Annu. Rev. Financ. Econ.*, 2012, *4* (1), 339–360.

_ , Itay Goldstein, and Edward Simpson Prescott, "Market-based corrective actions," *The Review of Financial Studies*, 2010, *23* (2), 781–820.

Campello, Murillo and Gaurav Kankanhalli, "Corporate decision-making under uncertainty: review and future research directions," *Handbook of Corporate Finance*, 2024, pp. 548–590.

_ , _ , and Pradeep Muthukrishnan, "Corporate hiring under Covid-19: Financial constraints and the nature of new jobs," *Journal of Financial and Quantitative Analysis*, 2024, *Forthcoming.*

Cao, Sean, Wei Jiang, Baozhong Yang, and Alan L Zhang, "How to talk when a machine is listening: Corporate disclosure in the age of AI," *The Review of Financial Studies*, 2023, *36* (9), 3603–3642.

_ , _ , Junbo L Wang, and Baozhong Yang, "From man vs. machine to man+ machine: The art and AI of stock analyses," *Columbia Business School Research Paper*, 2021.

Chen, Hui, Yuhan Cheng, Yanchu Liu, and Ke Tang, "Teaching Economics to the Machines," *Available at SSRN 4642167*, 2023.

Cong, Lin William, Guanhao Feng, Jingyu He, and Junye Li, "Sparse Modeling Under Grouped Heterogeneity with an Application to Asset Pricing," 2023.

_ , _ , _ , and Xin He, "Growing the Efficient Frontier on Panel Trees," *NBER Working Paper*, 2022, (w30805).

_ , Ke Tang, Jingyuan Wang, and Yang Zhang, "AlphaPortfolio: Direct Construction Through Reinforcement Learning and Interpretable AI," *Working Paper*, 2020.

_ , _ , _ , and _ , "Deep Sequence Modeling: Development and Applications in Asset Pricing," *The Journal of Financial Data Science*, 2021, *3* (1), 28–42.

_ , Tengyuan Liang, and Xiao Zhang, "Textual Factors: A Scalable, Interpretable, and Data-driven Approach to Analyzing Unstructured Information," Technical Report, resubmission requested. 2018.

Dicks, David and Paolo Fulghieri, "Uncertainty, investor sentiment, and innovation," *The Review of Financial Studies*, 2021, *34* (3), 1236–1279.

Dicks, David L and Paolo Fulghieri, "Uncertainty aversion and systemic risk," *Journal of Political Economy*, 2019, *127* (3), 1118–1155.

Ebert, Frederik, Chelsea Finn, Sudeep Dasari, Annie Xie, Alex Lee, and Sergey Levine, "Visual foresight: Model-based deep reinforcement learning for vision-based robotic control," *arXiv preprint arXiv:1812.00568*, 2018.

Edmans, Alex, Itay Goldstein, and Wei Jiang, "The real effects of financial markets: The impact of prices on takeovers," *The Journal of Finance*, 2012, *67* (3), 933–971.

_ , _ , and _ , "Feedback effects, asymmetric trading, and the limits to arbitrage," *American Economic Review*, 2015, *105* (12), 3766–3797.

Erel, Isil, Léa H Stern, Chenhao Tan, and Michael S Weisbach, "Selecting directors using machine learning," Technical Report, National Bureau of Economic Research 2018.

Feng, Guanhao, Stefano Giglio, and Dacheng Xiu, "Taming the Factor Zoo: A Test of New Factors," *The Journal of Finance*, 2020, *75* (3), 1327–1370.

Fu, Justin, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine, "D4rl: Datasets for deep data-driven reinforcement learning," *arXiv preprint arXiv:2004.07219*, 2020.

Fujimoto, Scott, David Meger, and Doina Precup, "Off-policy deep reinforcement learning without exploration," in "International Conference on Machine Learning" PMLR 2019, pp. 2052–2062.

Garlappi, Lorenzo, Ron Giammarino, and Ali Lazrak, "Ambiguity and the corporation: Group disagreement and underinvestment," *Journal of Financial Economics*, 2017, *125* (3), 417–433.

Graham, John R, "Presidential address: Corporate finance and reality," *The Journal of Finance*, 2022, *77* (4), 1975–2049.

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

Gulen, Huseyin, Candace Jens, and T Beau Page, "An application of causal forest in corporate finance: How does financing affect investment?," *Working Paper*, 2020.

Hanley, Kathleen Weiss and Gerard Hoberg, "Dynamic interpretation of emerging risks in the financial sector," *The Review of Financial Studies*, 2019, *32* (12), 4543–4603.

Hansen, Lars Peter and Thomas J Sargent, "Robust control and model uncertainty," *American Economic Review*, 2001, *91* (2), 60–66.

_ **and Thomas J. Sargent**, "Risk, ambiguity, and misspecification: Decision theory, robust control, and statistics," *Journal of Applied Econometrics*, 2023, *n/a* (n/a).

Hennessy, Christopher A and Toni M Whited, "Debt dynamics," *The journal of finance*, 2005, *60* (3), 1129–1165.

Hornik, Kurt, Maxwell Stinchcombe, and Halbert White, "Multilayer feedforward networks are universal approximators," *Neural networks*, 1989, *2* (5), 359–366.

Izhakian, Yehuda, David Yermack, and Jaime F Zender, "Ambiguity and the tradeoff theory of capital structure," *Management Science*, 2022, *68* (6), 4090–4111.

Jaques, Natasha, Asma Ghandeharioun, Judy Hanwen Shen, Craig Ferguson, Agata Lapedriza, Noah Jones, Shixiang Gu, and Rosalind Picard, "Way off-policy batch deep reinforcement learning of implicit human preferences in dialog," *arXiv preprint arXiv:1907.00456*, 2019.

Jarrow, Robert and Simon Sai Man Kwok, "Specification tests of calibrated option pricing models," *Journal of Econometrics*, 2015, *189* (2), 397–414.

Jin, Ying, Zhuoran Yang, and Zhaoran Wang, "Is pessimism provably efficient for offline rl?," in "International Conference on Machine Learning" PMLR 2021, pp. 5084–5096.

Kahn, Gregory, Pieter Abbeel, and Sergey Levine, "Badgr: An autonomous self-supervised learning-based navigation system," *IEEE Robotics and Automation Letters*, 2021, *6* (2), 1312–1319.

Kalashnikov, Dmitry, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke et al., "Scalable deep reinforcement learning for vision-based robotic manipulation," in "Conference on Robot Learning" PMLR 2018, pp. 651–673.

Kaniel, Ron, Zihan Lin, Markus Pelger, and Stijn Van Nieuwerburgh, "Machine-learning the skill of mutual fund managers," *Journal of Financial Economics*, 2023, *150* (1), 94–138.

Kelly, Bryan, Semyon Malamud, and Kangying Zhou, "The virtue of complexity in return prediction," *The Journal of Finance*, 2024, *79* (1), 459–503.

Kidambi, Rahul, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims, "MOReL: Model-based offline reinforcement learning," in "NeuIPS 2020" 2020.

Kingma, Diederik P and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

Levine, Sergey, Aviral Kumar, George Tucker, and Justin Fu, "Offline reinforcement learning: Tutorial, review, and perspectives on open problems," *arXiv preprint arXiv:2005.01643*, 2020.

Li, Kai, Feng Mai, Rui Shen, and Xinyan Yan, "Measuring corporate culture using machine learning," *The Review of Financial Studies*, 2020.

Malenko, Andrey and Anton Tsoy, "Asymmetric information and security design under Knightian uncertainty," *Available at SSRN 3100285*, 2020.

Mamaysky, Harry, Matthew Spiegel, and Hong Zhang, "Improved forecasting of mutual fund alphas and betas," *Review of Finance*, 2007, *11* (3), 359–400.

Mitton, Todd, "Methodological variation in empirical corporate finance," *The Review of Financial Studies*, 2022, *35* (2), 527–575.

Pan, Jun, "The jump-risk premia implicit in options: Evidence from an integrated time-series study," *Journal of financial economics*, 2002, *63* (1), 3–50.

Spiegel, Matthew, "For corporate finance to truly advance we need more genuinely testable models," *Financial Review*, 2023.

Sutton, Richard S and Andrew G Barto, *Introduction to reinforcement learning*, Vol. 135, MIT press Cambridge, 1998.

Wang, Zhenyu, "A shrinkage approach to model uncertainty and asset allocation," *Review of Financial Studies*, 2005, pp. 673–705.

Yu, Tianhe, Garrett Thomas, Lantao Yu, Stefano Ermon, James Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma, "Mopo: Model-based offline policy optimization," *arXiv preprint arXiv:2005.13239*, 2020.

Appendix A - Variable Selection and Definitions

Table 12 shows how we construct our firm-specific state and decision variables using Compustat and CRSP.

Table 12: Variable Selection and Definitions

variable	explanation	category	formula	source
lev	leverage	Decision	(dtllq + dlcq) / L.atq	Compustat
aqc	acquisition	Decision	aqcq / L.atq	Compustat
inv_ratio	investment	Decision	capxq / L.atq	Compustat
cash_hold	cash holding	Decision	cheq / L.atq	Compustat
div_payout	dividend	Decision	devq / L.atq	Compustat
log_debt_issue	debt issuance	Decision	log(1 + dltisq)	Compustat
log_eq_issue	equity issuance	Decision	log(1 + sstkq)	Compustat
rd_exp	R&D expenses	Decision	xrdq / L.atq	Compustat
repurchase	repurchases	Decision	prstkcq / L.atq	Compustat
log_at	book asset	State	log(1 + atq)	Compustat
log_act	current asset	State	log(1 + actq)	Compustat
log_sale	gross revenue	State	log(1 + saleq)	Compustat
log_ap	accounts payable	State	log(1 + apq)	Compustat
log_cogs	cost of good sold	State	log(1 + cogsq)	Compustat
log_intpn	net interest paid	State	log(1 + intpnq)	Compustat
log_invt	inventories	State	log(1 + invtq)	Compustat
log_lct	current liabilities	State	log(1 + lctq)	Compustat
log_rect	receivables	State	log(1 + rectq)	Compustat
log_revt	net revenue	State	log(1 + revtq)	Compustat
log_market_cap	market cap	State	log(1 + csho * prccq)	Compustat
log_enterprise_val	enterprise value	State	log(1 + atq + csho * prccq - ceqq - txdbq)	Compustat
log_vol	trading volume	State	log(1 + VOL)	CRSP
log_ret	equity return	State	log(1 + RET)	CRSP
macro1	risk	Macro	risk	ChicagoFed
macro2	credit	Macro	credit	ChicagoFed
macro3	leverage	Macro	leverage	ChicagoFed
macro4	non-financial leverage	Macro	nonfinancial_leverage	ChicagoFed

Appendix B - Technical Details

B.1 Predictive Environment Module Hyper-Parameters

Table 13: Predictive Environment Module: Neural Network Hyperparameters

hyperparameter	value (main)	value (macro)
optimizer	Adam	Adam
learning rate	0.00002	0.00025
batch size	2048	full
epoch (first)	40	100
epoch (rolling)	5	10
depth (main & aux)	3	3
neurons (main)	(300, 300, 300)	(30, 30, 30)
aux network num	10	-
neurons (min aux)	(280, 280, 280)	-
neurons (max aux)	(350, 350, 350)	-

B.2 Policy Module Hyperparameters

Table 14: Policy Module: Neural Network Hyperparameters

hyperparameter	value
optimizer	Adam
periods in obj. func.	{1, 8}
learning rate	0.00003
batch size	full
epoch	64
depth	4
neurons	(256, 256, 256, 256)

Appendix C - Ambiguity Constraint Parameters

We first estimate parameters in the ambiguity constraints, namely α and β in Eq.(6), for the long-term models. We follow an iterative process to search for proper constraints that could give out positive and reasonable long-term performance. For each cross section, we start from a tight lower-bound for these parameters, with 0 intercept and 1 slope. Essentially, this constraint requires AM managerial decisions as certain as the last set of observable decisions (the benchmark). Instead of search for optimal decisions which maximize long-term firm values, AM in this stage tries to minimize ambiguity of suggested managerial decisions. After 10 training epochs, we use a dynamic programming algorithm to estimate α and β so that the constraint never binds and the area under the curve is minimized. In this way, the ambiguity punishment for the current AM equals to 0, and the ambiguity constraint is tight enough that AM performance is not likely to be irrationally high. After training for 5 more epochs, we examine whether the reward after ambiguity punishment is positive on average. If yes, we continue to train AM for 5 epochs and move on to the next cross section; if not, the current ambiguity constraint is likely too tight, so that AM is too ambiguity-averse and does not have sufficient incentive to explore the environment and reach for higher firm value. In this case, we re-estimate α and β to loose the ambiguity constraint again and try to train the model for 5 more epochs, until it has a positive reward after ambiguity punishment.

We then estimate α and β for the short-term models. To make short-term models comparable to the corresponding long-term models in each cross section, we first calculate the long-term AM's short-term boosting error, and then we use the dynamic programming algorithm to estimate α and β. Throughout the training of the short-term AM for that cross section, we stick with this ambiguity constraint and try to optimize short-term objective. The short-term AM is expected to behave better than the long-term AM, because under the long-term AM, the suggestion managerial decisions fall inside the constraint, while short-term objective values are suboptimal.