Working Together, Forever? 
Project Evaluation, AI, and Managerial Redundancy

Vivek Choudhary 
Nanyang Technological University, vivek.choudhary@ntu.edu.sg

Arianna Marchetti 
London Business School, amarchetti@london.edu
Corresponding author

Yash Raj Shrestha 
University of Lausanne, yashraj.shrestha@unil.ch

Phanish Puranam 
INSEAD, phanish.puranam@insead.edu

As AI algorithms rapidly improve their performance at tackling managerial tasks, human-AI collaboration is often posed as a defense against managerial redundancy. We consider the case of project evaluation, an important component of managerial work, in which managers draw on their experience with past projects (e.g. investment, hiring, supplier selection) to evaluate the viability of new projects. Representing human and AI agents as supervised deep learning models, we consider how human redundancy probability increases with AI access to features data on past projects, across a variety of configurations for collaboration between human and machine. Our results from computational analysis indicate that division of labor with sequential ensembling – in which the human makes a prediction that is taken as input by the AI to then form its own prediction – is able to keep human redundancy probabilities very low even when data access to AI is very high. We find that it is superior in terms of lowering human redundancy probabilities to all other forms of division of labor between humans and AI or prediction tasks, including many currently being used.

Keywords: Human-AI Ensembling; Human-AI Collaboration; Project Evaluation; Automation; Organization Design

Electronic copy available at: https://ssrn.com/abstract=4657010
1. INTRODUCTION

Rapid advances in machine learning-based artificial intelligence (henceforth, AI) have raised the prospect of automation of knowledge work (Coombs et al. 2020; Sowa et al. 2021). Is managerial work, a form of knowledge work, also prone to automation through AI? Some have argued that while some aspects of managerial work may indeed be automated, managerial roles themselves are not at risk of redundancy given that humans and AI can complement one another and collaborate synergistically (e.g., Murray, Rhymer, & Sirmon, 2021; Shrestha, Ben-Menahem, & von Krogh, 2019). This is possible, scholars have argued, by exploiting the logic of “gains from trade” through specialization. This approach entails allocating distinct (sub-)tasks to the human (henceforth, H) and the AI to leverage their respective relative or absolute advantage in performing those (Canetti et al., 2019; Holzinger, 2016; Murray et al., 2020; Reuters, 2018), and then combining their outputs to generate the result.

Yet specialization can lead to a “creeping irrelevance” for H, as the areas in which they outperform algorithms and retain their skills become increasingly circumscribed as algorithms gain access to more data and computational power. To make this concrete, consider project evaluation which involves predicting and assessing a variety of alternatives based on their distinctive attributes, aiming to make informed decisions about resource allocation and selection of alternatives (e.g., Christensen & Knudsen, 2010; Csaszar & Ostler, 2020; Mintzberg 1976, Sah & Stiglitz, 1985, 1986, 1988). This decision making process occurs widely in managerial contexts, and scholars have used it to model decisions related to employee hiring, project investments, supplier and strategic partner selection, firm acquisitions, and new product launches (Csaszar 2012; Csaszar & Ostler, 2020). A logic of specialization suggests that it is possible to split the evaluation process such that managers specialize in some parts – for instance in analyzing qualitative data – leaving the analysis of quantitative data to algorithms.
While this division of labor can usefully exploit the agents’ relative advantages (i.e. H gives up the sub-task in which they are inferior to the AI), it also ensures over time that managers lose skills at the quantitative analysis that the AI performs, and offers no guarantees that algorithms will not eventually learn how to do qualitative analysis as well (or better) than managers, if only because managers themselves generate valuable data about the (sub-)tasks they specialize in (see also the problem of learning races in strategic alliances Khanna et al, 1994; Doz & Hamel, 2001). With the rise in data availability (e.g., Brynjolfsson & McAfee, 2017; Raisch and Krakowski, 2021) and advancements in model sophistication and computing power (Goodfellow et al. 2016), algorithmic accuracy can be expected to surpass managerial accuracy, making the advantage of managers at qualitative analysis, which is the basis for the logic of specialization, less tenable.

More recently, scholars have been interested in a different model of collaborative decision making between H and AI, which does not involve specialization. This is the logic of ensembling, which consists in aggregating predictions from multiple models applied to the same prediction problem (Sagi & Rokach, 2018). Unlike the traditional division of labor with specialization, where H and AI handle different (sub-)tasks, ensembling involves both H and AI working on the same prediction problem, whereby their respective predictions are then aggregated in a specific manner (e.g., averaging for estimation problems or using quorum, plurality, or unanimity for classification problems) to produce the final output. As Choudhary et al, 2023 note, ensembling in principle offers a role for humans even when their predictive accuracy is worse than that of AI, and it involves no loss of skills for H. The counter-factual—namely how AI and H would do if they were not collaborating, is also constantly observable, which is not the case with specialization.

What remains unknown is what happens to managerial redundancy under a division of labor with ensembling as data access to AI increases. Ensembling exploits “gains from
diversity” in prediction errors, and H contributes that through their access to unique data that is not available to AI. This contribution provided by H may also be threatened as algorithms gain access to more and more data over time. In the case of specialization, the consequences are easy to see: the more access to data that AI has for a given sub-task, the greater the probability of human redundancy at that sub-task. However, ensembling can occur in a variety of configurations— in parallel, with H and AI making predictions independently, or in sequence, where one of them (either AI first or H first) takes the predictions of the others as input. In this paper we investigate the following question: how do different configurations of ensembling impact the probability of H becoming redundant when H collaborates with AI for project evaluation tasks? Put differently, we investigate the elasticity of human redundancy risk to data availability to AI under different divisions of labor between humans and AI.

We draw on prior literature to model both H and AI as agents that learn from past data on projects, including their features and outcomes, to make predictions about outcomes of future projects based on their features (e.g. Hogarth and Karelia, 2007; Christensen and Knudsen, 2020; Csaszar and Ostler, 2020). We model both H and AI as supervised deep learning models. Two key differences exist between H and AI agents in our framework. First, the AI is endowed with a higher representational complexity than the human (i.e., we model AI with a deep learning algorithm that has a higher number of layers than the human’s). This assumption reflects the Universal Approximation Theorems (for details, see Kratsios, 2021), which demonstrate that neural networks can theoretically approximate any function of arbitrary complexity with a multi-layered structure, whereas humans engaged in prediction tasks are known to be well-represented by linear models (Hogart and Karelia, 2007). Second, we assume that H has access to all the available data for a given project evaluation task while AI cannot access the full data, as we assume that a portion (i.e. a subset of features) of it is non-codifiable and thus remains accessible to the H only. This could be for privacy or confidentiality reasons,
or simply because the data are multi-modal (e.g. image, sound, text) and the AI does not have access to all of these.

To evaluate projects and choose among alternatives, the predictions made by the H and AI are aggregated using two types of ensemble configurations. In sequential ensembling AI makes the final predictions using H’s prediction as an input to the model (or vice versa, which we call reverse sequential ensembling), while in parallel ensembling the independent predictions of H and AI are simply averaged. Performance for an agent or a configuration of agents is measured by out of sample predictive accuracy (i.e. accuracy on projects not part of the training data). In essence, we are comparing a complex model with incomplete data (AI) and a simple model with complete data (H), to their combinations.

This modeling setup allows us to examine how increased data access to the AI agent affects the probability of redundancy for the H agent. Our expectation is that while the risk of H’s redundancy increases in general with more data available to the AI, we should observe variations in the extent of such redundancy across the configurations we described above.

Consistent with intuition, we find that human redundancy probability (henceforth, HRP) at a project evaluation task does indeed increase as data access to AI increases. However, there are also three less obvious results. First, ensembling of any form (either parallel or sequential) lowers HRP for any level of data access to AI and even when H has no data advantage relative to the AI. This is because, besides data diversity, H still contributes model diversity to the ensemble: the simpler H’s model acts as a “cheap regularizer” of the more complex AI model which is prone to overfitting. Second, the sequential ensembling (with H making the first prediction) shows lower HRP than the parallel ensembling. This is the case because sequential ensembling provides a design that allows for a more flexible averaging between a less constrained functional form (AI) and a constrained one (H) than parallel ensembling. Third, sequential ensembling with H making the first prediction has lower HRP
and higher predictive accuracy than a sequential ensembling design where the AI makes the first prediction (which we label reverse sequential ensembling). This is because the aggregate representational complexity of the reverse sequential is lower than that of the sequential ensemble.

This paper makes several contributions to the human-AI collaboration literature. First, it tackles the issue of human managers replacement by AI in the context of project evaluation, which is not the only thing that managers do but is a significant part of it. In particular, our results about the configuration where H alone tackles project evaluation can be used to assess what tasks would be taken out of H’s hands under the logic of specialization. Second, it thoroughly examines an alternative to division of labor with specialization for H-AI collaboration, namely ensembling and its various configurations which may alleviate the risks of human redundancy in organizations. This sheds further light on the potential benefits of adopting ensembling as a collaborative approach to project evaluation tasks, providing valuable insights for managers and organizations seeking to harness the combined potential of H-AI collaboration in data-rich contexts. Third, it assesses how recent trends of increased data availability to AI may affect the stability of the human role in ensemble configurations. Lastly, unlike past research that compares H-AI collaboration designs based only on accuracy, it also introduces HRP as a new metric of evaluation to assess optimal designs.

2. LITERATURE REVIEW

2.1 Project evaluation as a canonical managerial task

A large part of what managers do daily consists of making decisions (e.g., Ghosh & Ray 1997, Shapira 2002) under uncertainty, which ultimately requires solving a prediction problem (Agarwal et al, 2018). An extensive body of literature has shown that AI algorithms can be extremely useful in tackling decision problems of the nature that managers face (e.g., Murray et al. 2020; Shrestha et al. 2021). The most impressive outcomes in predictive accuracy
to date have been produced by a type of AI that utilizes ML frameworks referred to as “deep neural networks” (LeCun, Bengio, & Hinton, 2015), i.e., complex, multi-layered structures of artificial neural networks.

Project evaluation represents a specific class of managerial predictive decision making that extant studies have focused on (e.g., Christensen & Knudsen, 2010). This is a form of predictive task that involves assessing various alternatives based on their characteristics, to make a final selection or decide how to allocate resources, using agreed-upon accuracy metrics (Christensen & Knudsen, 2010; Csaszar & Ostler, 2020; Sah & Stiglitz, 1985, 1986, 1988). Project evaluation encompasses a large part of the daily managerial work, including decision-making regarding hiring, project investments, supplier and strategic partner selection, company acquisitions, and new product launches (Csaszar & Ostler, 2020). Because of their predictive nature and due to the firms’ potential to generate large volumes of data in these domains, project evaluation in data-rich contexts is a type of managerial task that is well suited for H-AI collaboration (e.g., Choudhary et al, 2023), and represents the focus of this paper.

2.2 H-AI collaboration with specialization

Much prior literature on H-AI collaboration has focused on a form of design that depends on the principle of division of labor with specialization. This concept dictates that each entity carries out distinct, non-duplicated sub-tasks according to their respective capabilities and comparative advantage (Agrawal et al., 2018 section 6). This approach to division of labor leads to several economic benefits, such as cost efficiency, faster task completion, and scale and scope enlargement (Iansiti & Lakhani, 2020). This body of research has investigated two main design forms in the context of decision making: (1) H assigns (sub-)tasks to the AI, or (2) H trains the AI and approves its decisions.

Writing about the first class of designs (i.e., H assigns (sub-)tasks to AI), Dellermann et al. (2019) propose a conceptual understanding of hybrid or collective intelligence, where H
and AI collaborate, utilizing their individual strengths to enhance their combined performance beyond individual capabilities (Jarrahi, 2018; Murray et al., 2020; Seeber et al., 2020). Various applications demonstrate the benefits of this design (Daugherty & Wilson, 2018), from automated call centers to image-recognition algorithms in healthcare, where algorithms handle tasks more efficiently and cost-effectively than humans, while guaranteeing a similar level of quality as their human counterparts. This form of H-AI collaboration also works in scenarios requiring both routine and non-routine decision making. In these cases, AI can manage regular decision making tasks, while H step in during unusual circumstances, such as a significant change in data trends or a substantial drop in the quality of AI-based decisions (e.g., Attenberg, Ipeirotis, & Provost, 2015; Kamar, 2016; Garg, Shukla, Marla, & Somanchi, 2021). This logic instantiates specialisation at the level of (sub-)tasks between H and AI.

Studies such as Jain, Munukutla, & Held (2019) and Vellido (2020) exemplify the second stream of literature on H-AI collaboration (i.e., where H train and approve AI decisions). The aim in these designs is to enhance the predictive performance of AI algorithms by incorporating inputs from human decision makers, who are assumed to be superior to AI. This body of work mainly revolves around two modes of H-AI collaboration: “human as gatekeeper” and “human-in-the-loop” (henceforth, HITL). In the “human as gatekeeper” mode, H checks and approves AI outputs to mitigate potential errors. The HITL model, on the other hand, involves a more integrated role for H, primarily during the algorithm training phase, to increase its accuracy (Holzinger, 2016). It leverages the combined strengths of H and AI to create hybrid intelligence systems (Ostheimer et al., 2021), with H acting as trainers. They provide superior insights that can be used to rectify and improve the algorithm.

This existing body of literature on H-AI collaboration in decision making has not actively contemplated the idea of AI replacing human managers. This is largely due to two features of the design forms explored in this research stream: (1) many managerial tasks are
often considered complex and non-routinary, relying on tacit knowledge to be performed (e.g., Shamsie & Mannor 2013), and are thus difficult to automate by AI (e.g., Raisch and Krakowski 2021); and (2) the specialized division of labor that underlies these form of H-AI collaboration, by its nature, always gives H a role albeit of reduced or changed scope (i.e. H now takes on a sub-task, or even switches to a completely different task). Put differently, since they start with the assumption of human superiority in at least some sub-tasks, the question of human redundancy is moot.

However, such existing models of H-AI collaboration omit the dynamic implications of specialization. Assume H begins with a task T that can be decomposed into two sub-tasks T1 and T2 (though perfect decomposability itself is unlikely and there will need to be typically some cost of coordination). Even if H is superior to AI at subtask T1 (and the AI takes on T2 at which it is superior), this is unlikely to remain a stable state of affairs. With the rise in data availability (e.g., Brynjolfsson & McAfee, 2017; Raisch and Krakowski, 2021) and advancements in model sophistication (Goodfellow et al. 2016) for these tasks, algorithmic accuracy can surpass human accuracy even at T1.

In fact, as managers perform the (sub-)tasks they specialize in, they generate valuable data about those tasks, which becomes available to AI. This can lead to a possible gradual encroachment by the algorithm into T1, and potentially diminishing the scope of H’s involvement over time in this task. As data access to AI increases for a given task, AI then becomes more likely to outperform H at those, make them redundant, and ultimately replace them at T1 as well, though it is certainly possible, if not guaranteed, that H finds other new tasks to do (Acemoglu & Restrepo, 2019). Further this form of replacement is hard to reverse because as H relinquishes some (sub-)tasks to AI, their skills and competences over those (sub-) tasks decay (e.g., Balasubramanian, Ye & Xu 2022). Lastly, the counterfactual case where the H performs the entire task (T1 +T2) themselves is no longer observable, making it difficult
to assess the ongoing viability of this form of division of labor even if the task environment changes, and the dependencies between T1 and T2 change and require new forms of coordination. Specialization is not robust to architectural change.

2.3 Ensembling as an alternative to division of labor with specialization

More recent research (e.g., Choudhary et al, 2023; Csaszar and Steinberger 2022; Puranam 2020; Steyvers et al. 2022) has started exploring an alternative to the H-AI collaboration model based on specialized labor, namely ensembling. H-AI ensembling involves aggregating predictions from multiple models all tackling the same decision (e.g., Choudhary et al, 2023). Contrary to the standard division of labor where H and AI work on different (sub-)tasks, ensembling entails both entities addressing the same predictive problem, thus featuring a division of labor without specialization. The predictions from both parties are then aggregated to generate the final outcome.

Ensembling offers several advantages over specialization. First there is no need to imperfectly decompose a task T into sub-tasks T1 and T2, so that problems of coordination and response to architectural innovation are not as significant. Second, humans lose no skills since they continue to perform the entire task T. Third, the counterfactual of performance without collaboration (i.e. how H and AI would each do if acting alone on task T) is constantly observable. Fourth, and perhaps most importantly, ensembling can preserve a role for H even if H underperforms AI at both T1 and T2 (Choudhary et al, 2023). This is because ensembling is effective in improving the accuracy over that of its best component member’s. as long as it comprises (at least) weak learners exhibiting diversity in their prediction errors. This diversity can stem from diversity in the models the agents use to represent the decision problem and diversity in the access to different data (Csaszar & Ostler, 2020; also see Simons, Pelled, & Smith, 1999).
This implies however that the value added by H to a H-AI ensemble may also be diminished if algorithms gain access to more and more data over time. Ensembling can occur in a variety of configurations (Puranam, 2021)- in sequence, where one of them (either H or AI goes first) and takes the outputs of the other as input, or in parallel, with H and AI making predictions independently. We do not know how these different configurations differ in their HRP as AI gains more access to data. Specifically, we investigate how different configurations of ensembling impact the probability of H becoming redundant when H collaborates with AI for project evaluation tasks.

2.4 Summary

Prior literature on H-AI collaboration built on a division of labor with specialization has paid limited attention to the dynamics of human redundancy as data access to AI increases. Under specialization, it is not controversial that greater access to data for the task the human retains eventually risks eventual human redundancy at that task (though it is possible humans find other things to do). Under ensembling, a similar concern arises that greater data access to the AI diminishes the value of the diversity in prediction errors that humans bring to the ensemble. However, ensembling can take many configurations, and it is possible these differ in how sensitive they are in translating increasing data access to AI into HRP. This is what we seek to learn from our modeling exercise.

3. MODEL OVERVIEW

We test the extent to which human managers become redundant in an ensemble setting when AI has access to an increasing amount of data using a computational agent-based model. In doing so, we model two types of agents: a) a human, and b) a deep learning-based AI, each engaging in a project evaluation task, i.e., aimed at predicting the potential value generated by a given project by learning from past projects. We model both agents as neural networks (i.e., a form of supervised learning algorithm), whereby the data that agents have access to comprises
both the features (i.e., input variables to the decision making process) from past projects and the exact value (i.e., output) they generated. While H is characterized by a linear prediction model that has access to all features (and cases), AI is characterized by a model with higher representational complexity (greater nonlinearity and therefore flexibility in functional form) that has access only to a subset of the features.

Following Csaszar and Ostler (2020), we assume that agents make a prediction about the value that a given project generates according to their representation of the task environment generated by observing past features and outcomes of projects. We further assume that this task environment remains static over time. Besides the individual agents, each tackling a project evaluation task, we also model the H-AI ensemble, which performs the same prediction that the H and AI tackle separately. Both agents evaluate the same project and make the same predictions about the project’s value, although the agents might differ in the data they access or the representations of the task environment they develop to make those predictions.

We model two types of ensembles. In a parallel ensemble, both agents first produce independent predictions of the focal project’s potential value. The predictions of both agents are then averaged to form an ensemble prediction. In a sequential ensemble, on the other hand, one agent makes the final prediction using the other’s prediction as an input to its own model. Depending on the order of H and AI in the sequence, we model two types of sequential ensembles: 1) H-AI sequential ensembling (henceforth, sequential ensembling), where AI makes the final prediction using H’s prediction as input, and 2) AI-H sequential ensembling (henceforth, reverse sequential ensembling), where H makes the final prediction using the AI’s prediction as input.

With this model set-up, we study the differential rates at which H becomes redundant as access to data for the AI increases. In other words, our objective is to estimate the likelihood

---

1 In the supplementary analysis, we can also consider weighted averaging.
that a division of labor that includes the human agent, either alone or in ensemble, has lower than chance probability of being outperformed by the AI alone. We refer to this likelihood as human redundancy probability (HRP), which is defined as follows:

\[ \text{HRP} = \Pr \{\text{Accuracy}(\text{AI}) > \max\{\text{Accuracy}(\text{H}), \text{Accuracy}(\text{Ensemble H+AI})\}\} \text{ at task T} \] [1]

where Accuracy(AI), Accuracy(H), and Accuracy(Ensemble H+AI) denote the respective accuracies of prediction by AI alone, H alone, and the two agents in ensemble. The formula in [1] implies that, for high values of HRP, AI should outperform not only H alone, but also H and AI in ensemble, thus making H redundant in the project evaluation task.

### 3.1. The task environment

The task environment determines the type of projects evaluated by the agents and maps an outcome value to each project. Following the Brunswickian approach used in Csaszar and Ostler (2020), we describe each project by a fixed set of features (denoted by \(x_1, x_2, \ldots, x_M\)) and an outcome representing its value (denoted by \(y\)). For instance, consider the case of a manager tasked with deciding whether to hire a job candidate. The cues that the manager may use to make a prediction about the candidate’s future performance, based on which they make the hiring decision, could include, among others, the candidate’s experience, their educational qualification, their perceived friendliness during the interview process, as well as organizational aspects such as the size of the team that the candidate would join. Such cues form the feature class of the given prediction problem (i.e., \(x_1:x_M\)) and the actual performance of the candidate post-hiring decision represents the outcome variable (i.e., \(y\)).

Each environment corresponds to a data generating process (henceforth, DGP) that produces the outcome (i.e., \(y\)) as a function of the feature class (i.e., \(x_1:x_M\)), as in Csaszar and Ostler (2020). In our analysis, we simulate environments with a given size of the feature class
(i.e., M) and complexity as a polynomial of degree K. Even for a small M, the number of features in this data generating process can be of order as large as $M^C K$. Next, we set all the polynomial’s coefficients (i.e., its $\beta$ values) from a standard normal distribution (i.e., N(0,1)). We add gaussian noise $\varepsilon$ to the resulting polynomial, where $\varepsilon \sim$ normal distribution (0,U). The DGP can thus be summarized as follows:

$$y = \sum_i \beta_i x_i^k + \sum_i \sum_{j \neq i} \beta_{ij} x_j x_i^k + \varepsilon,$$

where $1 \leq i \leq j \leq M$ & $k \in [0,K]$ [2]

This setup allows us to examine how increased data access to the AI agent (from 1 to M features) affects the HRP. To ensure that our results are independent of a specific random draw, we report results averaged over 150 simulations. We only model a static environment where the agents do not face regime shifts in the DGP.

3.2. Differences in agents’ representational complexity and data access

We conceptualize each agent’s representation of the task environment as a function that takes as input the project features x and produces the predicted outcome $\hat{y}$ according to the characteristics of the task environment (Csaszar and Ostler, 2020) as per the following equation, where $f'$ is the agent’s approximation of the underlying reality $f$:

$$\hat{y} = f'(x_1: x_m) [3]$$

To model differences in representational complexity (i.e., the complexity of the function $f'$ for each agent) between H and AI, we adopt a neural network in the form of multi-layer perceptrons (henceforth, MLPs) to model each agent, since those allow for a large range of complexity, tunable by varying the number of hidden layers in the neural network. Following prior literature (e.g., Hogarth & Karelia 2007; Gigerenzer et al. 1999), we assume that AI has a higher representational complexity than H, i.e., AI can approximate more complex functional forms than H. We thus simulate H with a linear model, that is a neural network with

---

2 Results are robust to different numbers of simulations (i.e., 50, 100, 150, 200).
no hidden layers (similar to Csaszar and Ostler 2020). We simulate the AI as an MLP with at least one hidden layer.\textsuperscript{3} We denote the hidden layer size of H by $h_H$ and that of AI by $h_{AI}$.

Regarding data access, H and AI differ in the subset of problem features that they can access, and thus leverage, to make their prediction. We assume that H can always access all the available data, while the AI has access to a subset of it, (e.g., Choudhary et al, 2023; Lebovits et al, 2021). In other words, H has access to $m_H'$ cues and AI has access to $m_{AI}'$ cues, where $m = m_H' \geq m_{AI}'$. AI can access a subset of data available to H for a host of reasons: 1) the nature of H’s prior experience, which cannot be codified and made available to AI (e.g., Choudhary et al, 2023); 2) reasons of privacy or governance (e.g., Tucker 2018), as well as 3) strategic H behaviour that intentionally withholds data from AI in an attempt to stay relevant (e.g., Krishnaswamy et al. 2021; Ploug et al. 2023).

3.3. Agents’ predictive accuracy

The performance of each agent and their ensemble is measured as their ability to accurately predict the potential value of each new project in the hold-out sample. For a given project $i$, the prediction performance is given by the absolute residual of an agent’s prediction, i.e., $|y_i - \hat{y}_i|$. The overall performance over a given set of projects is then equal to the average absolute residual across projects, i.e., $\sum_i |y_i - \hat{y}_i|$. Each agent makes independent predictions on each project drawing on their representations of the task environment. Let us denote the predictions of H and AI on task $i$ by $y_i^H$ and $y_i^{AI}$, respectively.

We also evaluate the performance of three different ensembling approaches: parallel, sequential, and reverse ensembling. These three ensemble designs are like bagging and stacking in the extensive ML literature. In a parallel ensemble, H and AI make their\textsuperscript{3} For the AI’s problem representation, we could tune both the number of layers and the number of neurons per layer in the neural network (henceforth, NN). To maintain tractability, we only tune the AI’s number of hidden layers, which is known to be the hyperparameter that most directly models the agent’s representational complexity by allowing for higher order terms in their representation function, while keeping the number of neurons constant (at two, i.e., the minimum value). We refer the interested reader to Appendix A, where we provide further technical details on the use of MLPs and their underlying functional structures.


predictions independently and the ensemble prediction corresponds to the average of the outcome of the two i.e., \( \hat{y}_i = 0.5 (y_i^H + y_i^{AI}) \). In a sequential ensemble, H makes a first prediction, which is then used by AI as a new feature to generate the final prediction. In other words, AI uses the H’s prediction (i.e., the output) as an input feature to represent the environment and approximate the DGP. The prediction is thus equal to \( \hat{y}_i = f^{AI}(x, y, y_i^H) \).[4]

The sequential ensemble thus coincides with a type of weighted average, where \( \hat{y}_i = (Ax + By_i^H) \).

We can also reverse the order of H and AI and have a reverse sequential ensemble, where the H makes a prediction following the AI. In order words, H uses the AI’s prediction as an input and the ensemble prediction is given by \( \hat{y}_i = f^H(x, y, y_i^{AI}) \).[5]

We simulate the predictive performance of the agents and their ensemble using a computational agent-based model as follows: i) we generate a random task environment of complexity K and uncertainty U; ii) we draw N random projects from the environment; iii) we determine the agents’ data access and representational complexity; iv) the agents independently learn their representation of the task environment \( f' \) leveraging data from 80% of the projects, and then make their predictions based on the learned representation using the remaining 20% of the project data; and v) we compute the performance of H, AI and H-AI ensembles.

4. SIMULATION RESULTS

Our model examines the conditions under which H is redundant in project evaluation tasks. H can contribute to decision making in the presence of AI under either of the following conditions: a) the H’s predictive accuracy in decisions is superior to AI’s, or b) the ensemble’s predictive accuracy (which also comprises the H) is superior to AI’s. Therefore, H is redundant when both these conditions do not hold true. We express the HRP as a probability, as it is estimated based on frequency across a large number of simulation’s runs, and it corresponds to the number of occurrences where the HRP is strictly positive. Table 1 below describes the
simulation dataset and parameters, as well as the values we select for each parameter in the simulation.

— Insert Table 1 about here —

Figure 1 plots HRP for several configurations of agents tackling a project evaluation task (averaged across varying levels of problem complexity), namely: H alone, parallel ensemble, H-AI sequential ensemble, and AI-H sequential ensemble (or reverse sequential ensemble), for different levels of data access (i.e., problem features) to AI (either alone or in ensembling configurations). Figure 1 also reports a baseline model corresponding to a 50% probability. We use this baseline probability to assess the HRP of the different forms of division of labor: if a given configuration’s HRP falls below the baseline, then the AI dominates any configuration that involves a H (i.e., either H alone or an ensemble that includes H) at least half the time.

Consistent with intuition, we find that the HRP of a project evaluation task does indeed increase and cross the human redundancy threshold (i.e., the baseline model) as data access to AI increases. We also observe three less obvious findings, which can be better understood by combining the information contained in Figure 1 with that of Figure 2, where we show the underlying performance – i.e., accuracy (or average residuals in out-of-sample predictions) – across configurations (averaged across varying levels of problem complexity).

— Insert Figure 1 and 2 about here —

First, either type of ensembling (i.e., either parallel or two types of sequential) lowers HRP for any level of data access to AI. This has two noteworthy components. Ensembling always lowers HRP relative to specialization (which can be assessed by comparing the performance horse-race style between H and AI). In any form of ensembling, the HRP is lower than with a direct contest between H and AI, which arises when H specializes in a task that it is currently superior to AI on. Further, even when H has no data advantage relative to the AI
(i.e., for AI’s data access equal to five features), the HRP stays below the baseline probability for the sequential ensembling, and remains below this threshold even for parallel ensembling for lower levels of data access. This can be explained with the fact that H can still add value to the ensemble because of model diversity, despite not featuring any diversity in data availability compared to AI. AI models are prone to overfitting, and ensembling with a simpler human model acts as a correction to this, i.e., as a form of “cheap regularization.” We verify this by showing that, as representational complexity increases (for a given level of problem complexity), the AI becomes more prone to overfitting, and ensembling it with H brings increasing value.

Second, H-AI sequential ensembling (i.e., the configuration with H making the first prediction) outperforms the parallel ensembling in terms of HRP. This is because sequential ensembling acts as a form of averaging between the AI’s less constrained functional form and the H’s constrained functional form. Sequential ensembles thus correspond to the superset of all functional forms that can be built from parallel ensembles. To see this, let $f$ be the function estimated by AI and $g$ that by H, and $f$ is a higher order polynomial than $g$ and [8] below is a special case of [6]:

$$\text{Sequential ensemble } = f^{AI}(x, f^H) \quad [6]$$  
$$\text{Reverse sequential } = f^H(x, f^{AI}) \quad [7]$$  
$$\text{Parallel ensemble } = \frac{1}{2}[f^{AI}(x) + f^H(x)] \quad [8]$$

Third, sequential ensembling (i.e., the configuration where H makes the first prediction) outperforms reverse sequential ensembling (i.e., the configuration where AI makes the first prediction) in terms of both lower HRP as well as higher predictive accuracy. This is because the representational complexity of the reverse sequential ensembling, is lower than that of the sequential ensemble, as [7] is a lower order polynomial than [6].

5. ROBUSTNESS
We ran a battery of additional tests aimed at assessing the robustness of the pattern of results that we discussed.

5.1 How does problem complexity affect the ensemble performance?

First, we examine the effect of varying levels of problem complexity on our results. Figure 3 shows results on the variation of HRP across different forms of division of labor between H and AI for increasing values of problem complexity. To do so, we average the HRP across different levels of data access to AI. First, we find that, at a given level of problem complexity, project evaluation tackled by H alone (which proxies specialization) yields the highest HRP, compared to either form of ensembling (i.e., parallel or sequential). While the HRP is below the baseline probability for H’s redundancy for both very low and very high levels of problem complexity, for moderate problem complexity, the HRP increases and exceeds the baseline probability, indicating that AI outperforms not only H alone, but also the parallel ensemble.

Second, the HRP displays an inverted U-shape with increasing problem complexity for the cases of H alone and the parallel ensemble. Although it would seem natural to predict that AI outperforms both H alone and its ensemble for highly complex tasks, we observe, counterintuitively, that it is not the case. This can be explained by the fact that, at higher levels of environmental complexity, the AI is likely to incur overfitting issues due to its ability to model complex relationships. On the other hand, due to linearity constraints in the human representational complexity, H will likely underfit the data. Therefore, in an ensemble, the H’s model acts as a regularizer for the AI and helps improve the overall predictive accuracy. At high levels of environmental complexity, the HRP decreases below the baseline value, indicating that H, either alone or in ensemble has a chance to beat AI in prediction: While increasing the data accessible to AI worsens HRP for H alone and the parallel ensemble, that is not the case at high levels of complexity.
Third, the difference in HRP between H, AI, and their ensemble is lower at both very low and very high levels of environmental complexity, while it increases significantly for moderately complex tasks. This can be explained by noting that, at lower levels of complexity, both the simpler H model and the more complex AI model can perform satisfying predictions, and thus there is not much difference in HRP across the forms of division of labor. At a higher level of complexity, however, a complex model has low predictive accuracy, providing H with an opportunity to improve the performance of the ensemble. As both H and AI predict poorly in case of high complexity, their ensemble differences are also not very high in performance.

— Insert Figure 3 about here—

We further confirm that our pattern of results is robust to alternative values of simulation runs (i.e., in the range of [50, 100, 150, 200]). Results are also robust to higher levels of correlation between features.

6. DISCUSSION AND CONCLUSIONS

A significant part of what managers do in organizations consists of “project evaluation,” which involves predicting and assessing a variety of alternatives based on their distinctive attributes, to make informed decisions about resource allocation and selection of alternatives (e.g., Christensen & Knudsen, 2010; Csaszar & Ostler, 2020; Sah & Stiglitz, 1985). AI technologies based on machine learning are becoming increasingly competent at project evaluation tasks given the abundance of data that organizations produce on them. It is a truism that if AI has access to all data humans have and also have vastly more flexible functional forms to model the patterns in this data, then it will exceed humans in their predictive accuracy based on this data. It is true that the path towards complete data access is a gradual one for machines, and the end state may never be attained. However, human redundancy risk may increase with every step. We investigate the elasticity of this redundancy probability w.r.t data access to AI under different divisions of labor between H and AI.
We used a computational model of human-AI ensembling in the context of project evaluation where we represent both the human and AI agents as supervised (deep) learning models. H and AI differ along two dimensions: 1) AI has a higher representation complexity than H (i.e., we model AI as a neural network with a higher number of hidden layers than that representing H), and 2) H has access to more data than the AI. Assessing the probability of H’s redundancy at increasing levels of data availability to AI, we find that the rate of its increase with AI’s increase in data access varies widely by the chosen division of labor. Specialization involves humans alone taking on a distinct sub-task. Our results show that as data access for this sub-task increase for the AI, HRP increases the fastest. However, if H are ensembled with AI for this sub-task, either in sequential or parallel, the rise in HRP is retarded. Further, sequential ensembles with H first are the best at flattening the HRP curve. While not literally guaranteeing “working together forever”, this configuration does keep HRP very low even as AI access to data becomes very high. This is contrary to the design commonly adopted by organizations where H uses the AI’s outputs to make a final prediction.

These findings have important implications for the literature on H-AI collaboration. First, we explore the risk of human managers’ replacement by AI in data-rich project evaluation tasks, which has been overlooked in prior literature. Second, we advance research on H-AI ensembling by exploring the robustness of this design to changing conditions about data availability to AI. Third, we find that, counterintuitively, a design where H shares tasks and the underlying data necessary to make accurate predictions with the algorithm, instead of withholding information from the AI, is the most promising to keep the H relevant in decision making processes.

Our results are to be interpreted in light of several boundary conditions based on our modeling choices. First, organizations may face alternative goals when adopting AI for decision-making: while predictive accuracy might be crucial, they may also be concerned with
the prediction’s fairness and other attributes according to organizational and/or social norms (e.g., Conlon, Porter & Parkes 2004 JOM). However, in this paper we focus exclusively on predictive accuracy of a decision making task, in the narrow sense of smaller residuals. We do not distinguish errors of over-from under estimation, for instance. Given the nature of project evaluation, where objective metrics exist to assess the quality of the alternatives available to the organization, this seems a fair assumption for a first model on how ensembles can tackle these tasks.

Further, in studying the predictive accuracy of H-AI ensembles, we only consider static environments, i.e., environments where no significant change in data generating process occurs. Machine learning algorithms are known to have highest predictive accuracy in such regimes where both data shifts and covariate shifts are absent. Our results might thus be conservative in that they may indicate higher HRP as compared to cases where environmental changes do occur. Under regime shifts, the question then becomes one of comparing the ability of AI vs H to unlearn and relearn from sparse data (including through analogy from other known settings that resemble the new regime). We intend to investigate these issues in future work.

Finally, while our current framework already accounts for the intentional withholding of data by human agents, there remains a noteworthy aspect yet to be thoroughly examined: namely, the potential adversarial behavior of human actors manipulating data. Understanding how human entities may deliberately alter data introduces a new layer of complexity, shedding light on the resilience and adaptability of AI to such inputs. Investigating the interplay between human-driven alterations in data and the learning outcomes of AI systems can contribute valuable insights to enhance the robustness and security of artificial intelligence applications in evolving and dynamic environments. This exploration holds significance in advancing our
comprehension of the intricate relationships between human behavior, data integrity, and the effectiveness of AI learning processes.

5. REFERENCES


6. FIGURES AND TABLES

Figure 1. HRP across different configurations of H and AI, alone and in ensembling (averaged across varying levels of problem complexity).

Figure 2. Accuracy (average residuals in out of sample prediction) across configurations of H and AI, alone and in ensembling (averaged across varying levels of problem complexity).
Figure 3. Variation in HRP across configurations of H and AI, alone and in ensembling, at different levels of problem complexity (K).
Table 1. Description of simulation dataset and parameters’ values.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Example</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Features</strong></td>
<td>Five features, or variables, in the dataset. Each variable represents a specific feature of a project that is likely to influence the outcome and is thus of interest to the decision maker. These features encompass both codifiable tabular data (i.e., accessible to both H and AI) as well as non-codifiable experience data (i.e., only available to H). To model differences in data access to H and AI, we assume that H always has access to all available data, while AI can only access the subset of codifiable data. In other words, we model different regimes of data availability to AI by varying the number of codifiable tabular features that AI can leverage.</td>
<td>Feature 1 = job candidate’s education  Feature 2 = job candidate’s technical skills  Feature 3 = job candidate’s previous position  Feature 4 = job candidate’s seniority  Feature 5 = job candidate’s body posture during interview</td>
<td>5 features in total  H’s data access: 5 features  AI’s data access: [1, 2, 3, 4, 5]</td>
</tr>
<tr>
<td><strong>Outcome</strong></td>
<td>This represents the outcome variable in the past data.</td>
<td>Performance evaluation of past hired candidates.</td>
<td>1 variable in total</td>
</tr>
<tr>
<td><strong>Number of rows</strong></td>
<td>The simulation dataset contains 5,000 rows (or cases), i.e., observations about past projects that are available to both the H and the AI for training. We highlight that we model differences between H and AI in data access by varying the number of features each can access, while we assume that both can access all the cases available in the dataset. Indeed, we divide the data rows in 80%-20% split to train and test our H and AI models.</td>
<td>Total number of previous job candidates hired by the firm, for which data is available along the five features described above as well as their performance evaluations.</td>
<td>5,000</td>
</tr>
<tr>
<td><strong>Problem complexity (K)</strong></td>
<td>The DGP is categorized by different complexity levels (i.e., polynomial degree of the environment’s DGP), which represent different scenarios or conditions under which the data was collected or generated. For instance, when problem complexity $K = 1$ is 1, it denotes a DGP where the relationships between features and outcome are linear. On the other hand, when problem complexity $K = 6$, such a relationship could have at most degree six complexity.</td>
<td>Hiring low skilled workers in a manufacturing assembly lines could be considered a low-complexity problem, while hiring higher skilled candidates in consulting firms could be considered a high-complexity problem.</td>
<td>[1, 2, 3, 4, 5, 6]</td>
</tr>
<tr>
<td><strong>Noise type</strong></td>
<td>The noise in the dataset represents uncertainty. It follows a gaussian distribution. Noise can be either additive or multiplicative. Additive noise is added as a separate value to the data generating process. Multiplicative noise, on the other hand, is multiplied to the data points by random factors.</td>
<td>Noise can for instance be caused by the limited ability to accurately assess the job candidate’s technical skills (feature 2).</td>
<td>Multiplicative</td>
</tr>
<tr>
<td><strong>Noise mean</strong></td>
<td>The noise mean represents the average level of noise multiplied to the data at each complexity level (K).</td>
<td></td>
<td>[0, 0.1, 0.3, 0.5, 0.7, 0.9]</td>
</tr>
<tr>
<td><strong>Noise standard deviation</strong></td>
<td>Each complexity level corresponds to a given noise standard deviation, which indicates the variability of the noise around the mean.</td>
<td></td>
<td>[0, 0.05, 0.15, 0.25, 0.35, 0.45]</td>
</tr>
<tr>
<td><strong>Correlation among features</strong></td>
<td>Correlation means among the project features for each complexity level.</td>
<td>Job candidate’s education (feature 1) and job candidate’s technical skills (feature 2) are highly correlated, while job candidate’s previous position (feature 3) and job candidate’s body posture during interview (feature 5) are less likely to exhibit a high correlation.</td>
<td>[0, 0.25, 0.5, 0.75, 1]</td>
</tr>
<tr>
<td><strong>Mean standard deviation of features</strong></td>
<td>It represents the mean and variability of the project features.</td>
<td>Average and variability in the job candidate’s seniority (feature 4).</td>
<td>0</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>Value</td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td>-------</td>
<td></td>
</tr>
<tr>
<td><strong>Training iterations</strong></td>
<td>Number of iterations over which the model is trained for during the learning process. This indicates the number of times the algorithm updates its parameters to fit the data during the training phase.</td>
<td>n/a</td>
<td>100,000</td>
</tr>
<tr>
<td><strong>Early stopping rule</strong></td>
<td>Early stopping is a technique used to prevent overfitting by stopping the training process when the model’s performance on a validation dataset starts to degrade. This parameter can take two values: “True,” when the early stopping is enabled during training, and “False” otherwise.</td>
<td>n/a</td>
<td>True</td>
</tr>
</tbody>
</table>
| **Agent’s representational complexity: AI’s number of hidden layers** | Agents’ neural networks feature a varying number of hidden layers, which we tune to model different levels of representational complexity between H and AI. | n/a | AI: [0, 1, 2, 3, 4]  
H: 0 |
| **Agent’s representational complexity: number of hidden nodes per layer** | Each hidden layer in an agent’s neural network features a tunable number of nodes, i.e., neurons. Together with the number of hidden layers, this allows us to model the agent’s representational complexity | n/a | AI: 2  
H: 0 |
| **Simulation runs** | Number of runs the simulation is repeated over, to ensure robustness and reliability of the results. | n/a | 150 |
A Multi-Layer Perceptron (henceforth, MLP) is a type of artificial neural network that consists of multiple layers of nodes or neurons, organized in three main types of layers: an input layer, one or more hidden layers, and an output layer. It is a feedforward neural network, where information flows through the network in one direction, from the input layer to the output layer. Next, we provide a brief description of each layer in an MLP.

**Input Layer.** The input layer is responsible for receiving the initial input data. Each node in this layer represents a feature of the input, and the number of nodes is equal to the number of input features. There is no computation performed in the input layer; it simply passes the input to the next layer.

**Hidden Layers.** Between the input and output layers, there can be one or more hidden layers. Each hidden layer consists of nodes that process the input data using weights that are adjusted during the training process. The term “hidden” refers to the fact that the values in these layers are not directly observable as input or output.

**Output Layer.** The output layer produces the final result or prediction of the network. The number of nodes in the output layer depends on the nature of the problem, such as the number of classes in a classification task or the number of output values in a regression task.

The connections between nodes in adjacent layers are represented by weights. Each connection has an associated weight, and these weights are adjusted during the training process using optimization algorithms like gradient descent. Additionally, each node in the hidden layers and the output layer typically includes an activation function, which introduces non-linearity to the model and allows it to learn complex patterns in the data. The overall architecture of an MLP makes it capable of learning and approximating nonlinear relationships in the data, making it a powerful tool for tasks such as classification, regression, and pattern recognition.