Journal of Evolutionary Economics (2023) 33:1473–1517
https://doi.org/10.1007/s00191-023-00845-3

REGULAR ARTICLE

Superhuman science: How artificial intelligence may impact innovation

Ajay Agrawal[1,2] · **John McHale**[3] · **Alexander Oettl**[2,4]

Accepted: 24 October 2023 / Published online: 21 November 2023

Abstract
New product innovation in fields like drug discovery and material science can be characterized as combinatorial search over a vast range of possibilities. Modeling innovation as a costly multi-stage search process, we explore how improvements in artificial intelligence (AI) could affect the productivity of the discovery pipeline in allowing improved prioritization of innovations that flow through that pipeline. We show how AI-aided prediction can increase the expected value of innovation and can increase or decrease the demand for downstream testing, depending on the type of innovation, and examine how AI can reduce costs associated with well-defined bottlenecks in the discovery pipeline.

Keywords Artificial intelligence · Innovation · R&D prioritization

JEL classification 033 · 031 · D20

1 Introduction

In November 2020, Google DeepMind's AlphaFold won the 14th round of the CASP[1] protein folding competition. They used a type of AI, machine learning using deep neural networks, to help predict the 3D structure of target proteins

[1] Community Wide Experiment on the Critical Assessment of Techniques for Protein Structure Prediction (CASP).

✉ Alexander Oettl
OETTL@gatech.edu

[1] University of Toronto, Toronto, Canada

[2] National Bureau of Economic Research (NBER), Cambridge, MA, USA

[3] J.E. Caines School of Business and Economics, University of Galway, Galway, Ireland

[4] Scheller College of Business, Georgia Institute of Technology, 800 West Peachtree St., NW, Atlanta, GA 30308, USA

based on their amino acids sequences. The "protein folding problem" is challenging because of the vast number of potential shapes that a protein could take for a particular amino acid sequence. Knowing the shape of proteins is, among other uses, critical for identifying targets for drugs to bind to in order to produce therapeutic effects.

> *"This will change medicine. It will change research. It will change bioengineering. It will change everything," says Andrei Lupas, an evolutionary biologist at the Max Planck Institute for Developmental Biology in Tübingen, Germany, who assessed the performance of different teams in CASP. AlphaFold has already helped him find the structure of a protein [in half an hour] that has vexed his lab for a decade, and he expects it will alter how he works and the questions he tackles…* (from *Nature*, November 2020)[2]

This is just one example of the technological evolution of AI for scientific discovery and innovation (Dosi 1982). Machine-learning-based AI tools are increasingly being applied where innovators must search over large and complex search spaces, promising to improve the productivity of the innovation process in various domains that have proved challenging for researchers, such as materials science, proteomics, genomics, and drug discovery.

Although there is extensive debate over the prospects for artificial *general* intelligence (AGI), the AI we consider in this paper is *narrow*, helping the innovator on the specific task of predicting which innovations are most likely to succeed. On this narrow task, the AI can be *superhuman* in the sense of surpassing human-level performance. Notwithstanding this superhuman capacity at specific tasks, we do not assume that the AI replaces the human scientist, but rather explore how the combination of the AI and scientists can increase discovery through improved prioritization.

We develop a model of innovation in domains that require search over vast combinatorial search spaces that includes two stages: prediction and testing.[3] Prediction by human scientists is characterized as formulating hypotheses based on theory and prior empirical evidence. Prediction by AIs is characterized as pattern recognition based on prior experimental data. In our model, testing remains fully in the domain of human scientists and involves running experiments to test the predictions from the first stage. In other words, AI only impacts the first stage: prediction.[4]

The model is motivated by five main ideas. First, innovation can be viewed as search over a (potentially vast) combinatorial search space. Second, innovation results from a costly multi-stage process. Third, the uncertainty innovators face as to the location of valuable combinations in this space can be partly reduced by having a prediction model – or map – of the underlying search landscape. Fourth,

[2] https://www.nature.com/articles/d41586-020-03348-4

[3] In the Appendix, we extend this to multiple stages by including intermediate screening stages between prediction and testing.

[4] We underline that limiting the use of AI to the prediction stage is a simplifying assumption – although one that we think captures the main current use of AI in discovery processes. Over time, AI is likely to be increasingly used in other stages (e.g., in the selection of control groups in clinical trials).

the output of the prediction model can usefully be summarized in the form of a ranking function that allows prioritization in the context of the costly multi-stage search process. Fifth, breakthroughs in AI are a potential source of improvement in the performance of these maps.

The main contribution of this paper is to develop a framework for linking an AI-based improvement in prediction technology to the economic effects on innovation in combinatorial-type discovery problems. Using the device of the ranking function, we show the effects of the improved technology on the expected marginal values and costs of potential innovations. However, the model also shows how the impact on innovation depends in subtle ways on features of the innovation search process such as whether the search is parallel or sequential, whether the potential innovations are independent or are substitutes, and the extent of bottlenecks in the discovery pipeline.

Scholars have long modeled innovation as a process of combining existing knowledge to produce new knowledge (Usher 1929; Schumpeter 1939; Nelson and Winter 1982; Weitzman 1998; Fleming and Sorenson 2004; Arthur 2009). The idea of new knowledge as new combinations has received particular attention where the mapping from existing knowledge inputs to new knowledge is highly complex. Examples of such domains of discovery include biotechnology, molecular and materials science, and particle physics. Recent advances in machine learning – and in particular advances in artificial neural networks such as deep learning – have led to optimism that these advances provide a new general purpose technology (GPT) for discovery (Agrawal et al. 2019a; Cockburn et al. 2019). These tools are already altering innovation practice in fields such as genomics, proteomics, small molecule drug discovery, and materials science, even if there is skepticism about their ultimate impact on R&D productivity.

To help understand how AI might affect innovation in domains requiring search over complex combinatorial search spaces, we build on the classic innovation function approach to economic growth to develop a model of AI-aided innovation. This approach models innovation as a function of research effort and existing knowledge stocks (Romer 1990; Grossman and Helpman 1991; Aghion and Howitt 1992; Jones 1995). In our model, the innovator must search over a vast space of potential combinations. Innovators use knowledge in the form of data on past successes and failures to develop a prediction model for the "fitness landscape" that maps combinations to the probability of success of those combinations. In essence, the model of the fitness landscape aids their search in the context of a multi-stage search process by helping to predict which combinations out of possibly billions have the greatest likelihood of success. We thus treat a key part of the discovery process as a prediction problem that can benefit from recent advances in AI. The model can therefore be seen as contributing new candidate microfoundations for the innovation production function.

In the model, a summary measure of the output of the prediction model is a logistic ranking function that shows how the probability of success declines as we move from better- to worse-ranked combinations. This function plays a key role in the prioritization process for a multi-stage discovery pipeline where the later stages

– screening and testing – are costly, and access to improved technologies for prediction can better prioritize the use of costly R&D resources.[5]

We draw on the literature on optimal search where information is imperfect and search is costly. This literature originated with Stigler (1961). In particular we draw on and extend a special case of the "Pandora's box" model developed in Weitzman (1979). The sequential search problem examined by Weitzman involves boxes that vary in the distribution of potential outcomes, search costs, and the time that elapses before the value of the box is revealed. He shows that a "reservation price" can be attached to each box. Ranking the boxes in descending order of reservation prices, the optimal search rule – "Pandora's rule" – is to continue down the ranking until the maximum value obtained is greater than the reservation prices of all remaining unopened boxes.

We develop a special case of the Weitzman model where there are just two outcomes that can be revealed in the test of a given combination – success or failure – and there is an unbiased estimate of the probability of success available to a risk-neutral innovator from the prediction model. We also assume that costs are the same for all tests and that there is no time discounting. This allows the innovator to prioritize solely based on the ranking of combinations given by the prediction model, and we make a strong assumption about the functional form of the ranking function that is the output of the prediction stage of the discovery process.

Using this tractable search setup, we examine three cases. In Case 1, the value of each success is independent of whatever other successes are found and the search can proceed either in parallel or sequentially. The objective of the innovator is then to discover all combinations with an expected net value greater than or equal to zero. In Cases 2 and 3, successes are assumed to be perfect substitutes so that having achieved a success the marginal values of any further successes are equal to zero. In Case 2, the testing of combinations is assumed to take place in parallel, and the innovator must decide on the portfolio of combinations to send for parallel testing. In Case 3, which most closely follows the Weitzman model, the innovator is assumed to search sequentially, with the search stopping when a success is found.[6]

We organize the remainder of the paper as follows. Section 2 positions the paper in the context of a number of related literatures. Section 3 briefly illustrates how AI is altering the discovery process in genomics, proteomics, drug discovery, and materials science. Section 4 sets out the basic conceptual building blocks of search-based innovation model over a vast combinatorial space and introduces the idea of a ranking function. Section 5 then develops a simple two-stage example of "search with a

[5] In the context of a multi-stage discovery pipeline with potentially multiple intermediate screens between initial prediction and final (determinative) testing, we examine in the Appendix how bottlenecks in the pipeline in the form of high costs and ineffective screens can reduce the number of potential innovations that enter the pipeline. We also explore how such bottlenecks can reduce the return to the adoption of AI-based prediction and the possibilities for AI improving – or even substituting for – later costly stages of the pipeline.

[6] Drawing on the real options literature, in the Appendix we also extend our two-stage search process to a more general multi-stage search process where there is an option to abandon an advancing combination at the end of each screening stage (Roberts and Weitzman 1981; Dixit and Pindyck 1994) and the probability of success is sequentially revised using Bayesian updating.

map" in which the first stage is the development of a prediction model (the output of which is captured by a logistic ranking function) combined with identification of a probability threshold for determinative testing, and the second stage is costly testing of all combinations with a probability of success at or above the threshold. Section 6 reflects on AI as a method of generating improved prediction models (or maps) of the fitness landscape that has undergone rapid recent development. Section 7 concludes with a recap of the main ideas and possible directions for future research.[7]

2 Related literature

The paper is related to a number of literatures. First, our paper draws on literature in economics that applies the ideas of fitness landscapes to the study of innovation.[8] As used in economics and management science, this work is typically situated in evolutionary economics and builds on the work of Nelson and Winter (1982). Important contributions include Levinthal (1997), Gavetti and Levinthal (2000), Kauffman et al. (2000), Rivkin (2000), Fleming (2001), Fleming and Sorenson (2004). introduced the idea of science as a map to aid technological search, an idea that is central to our approach.[9] In our model, innovators use (imperfect) knowledge of the fitness landscape (the prediction model) to identify a promising subset of potential combinations followed by screening/testing of that subset.

Second, our paper is inspired by growing literature describing the use of machine learning in scientific discovery and innovation. Rapid advances in hardware, software, and data availability have driven this growth, with applications of deep neural networks showing notably rapid growth across a number of domains. For a sampling of reviews see Chen et al. (2018); Vamathevan et al. (2019) [drug discovery], Angermueller et al. (2016); Tang et al. (2019) [computational biology], Wainberg et al. (2018); Zou et al. (2019) [genomics], Goh et al. (2017), Nature Communications (2020) [computational chemistry], and Butler et al. (2018); Keith et al. (2021) [materials science]. The apparent success of deep learning in providing predictive models of highly complex combinatorial spaces explains the rising interest in it as a GPT for discovery. This potential – together with the power of viewing innovation as a prediction-model-aided combinatorial search process – motivates the paper.

Third, we draw on and extend the innovation production function that has been at the core of developments in endogenous growth theory (see, e.g., Jones 2005; Trammell and Korinek 2023). As already noted, central to the innovation

[7] Another For a discussion of the policy implications of the model, see the working paper version of this paper (Agrawal et al. 2022).

[8] Sewall Wright (1932) first introduced the fitness landscape concept in evolutionary biology, and Stuart Kauffman (see Kauffman 1993) extensively developed it.

[9] Drawing on the evolutionary approach, these papers model innovation (or imitation) as a "walk" or "hill climb" towards a local optimum on the fitness landscape. Innovators typically search one-mutant neighbors and adopt fitter variants until a local optimum is reached, although "long-jumps," in which innovators jump longer distances across the landscape, are also studied as a process of exploratory search. The evolutionary approach has proved extremely fruitful and provides rich dynamics for the search process.

production function approach is the idea that existing knowledge is an input into the production of new knowledge – the "standing on the shoulders of giants" or spillover effect. Typically, papers in this literature do not explicitly adopt a combinatorial view of the process through which existing knowledge is turned into new knowledge (although see Romer 1992; Weitzman 1998; Jones 2021) for important exceptions.[10]

Finally, our paper draws on contributions to the emerging literature on the economics of artificial intelligence. A key breakthrough in AI has followed from the shift from rules-based systems to a statistical approach that emphasizes prediction (see, e.g., Athey 2017; Mullainathan and Spiess 2017; Agrawal et al. 2018; Taddy 2019). As emphasized by Arrow (1962), uncertainty is a pervasive feature of the innovation process and hence the value of prediction technologies that help reduce it. We can view AI – and machine learning in particular – as a GPT for prediction and make extensive use of this idea in our paper.[11]

3 Innovation as search over complex spaces: Some motivating examples

To help motivate our modeling approach, we first provide some illustrations of discovery challenges that involve search of complex combinatorial spaces, challenges for which machine learning-based prediction models appear to be fruitful. The common structure of such challenges is that an innovation involves a combination (e.g., a set of gene interactions or a chemical molecule) that has relevant properties or activities (e.g., a relationship to an intermediating cell variable or a ligand–protein binding affinity). The discovery processes we consider are typically multi-staged (e.g., predictive screening, synthesis, testing, etc.). We are especially interested in how researchers use machine learning to screen candidate combinations and thereby allow a ranking of combinations for further exploration along the discovery pipeline.

[10] In an elegant recent paper, Jones (2021) combines the insights of Kortum (1997); Weitzman (1998) to explore the links between combinatorial growth in the size of search and exponential economic growth. The driver of economic growth is that the innovator makes draws from a known distribution and growth results from an increase in the best alternative drawn. Jones shows how exponential growth results for certain distributions with thin tails when there is combinatorial growth in the number of draws. A key difference between his model and the one in this paper is that we assume that search (i.e., testing) is costly. This forces the innovator to prioritize instead of exhaustively searching the known landscape. Another important difference is the information available to our innovator is in the form of a fitness function over the search landscape, which contrasts with knowledge of the distribution in Jones' model. We introduce AI as a means to improve the model of the landscape and thus improve prioritization. We also highlight an important source of spillovers: data on past successes and failures. These data are used to develop improved prediction models and thus can be a source of growth-sustaining productivity improvement in the innovation search process.

[11] Another important strand of the economics of AI literature has focused on the effects of AI on the demands for different types of skill. Researchers in this area have used the idea of a task-based production function to allow for the possibility that the introduction of AI (and other new technologies such as robotics) could lower the employment and wages of certain types of workers depending on the tasks they perform (Acemoglu and Autor 2011; Autor 2015; Acemoglu and Restrepo 2018, 2019a, b). Such labor demand effects could take place for knowledge production tasks as well (Aghion et al. 2019).

Our first example is from genomic medicine. A genome is a hugely complex set of instructions for building an organism. This building process can be viewed as a combinatorial problem: genes interact in complex ways including the interaction between protein coding and regulatory regions within the genome. Genomic medicine exploits the relationship between DNA sequences and the risk of various diseases.

A central idea is that of gene expression, the process by which the information in the gene is first transcribed to make messenger RNA (mRNA) and then the mRNA is translated to make a protein (Leung et al. 2016). Researchers can utilize predictive models for various stages of this process. An encompassing approach is to model the relationship between DNA sequences and disease outcomes.[12]

Our second example is proteomics – the study of the structure and function of proteins. Understanding proteins is a complex task given their vast number (far exceeding the number of genes) and interactions with cell and environmental variables. Part of the biochemical process of gene expression is the translation via the ribosome of mRNA into the amino acid sequences that comprise the protein. As noted in the Introduction, one of the challenges of proteomics is the prediction of the tertiary (or 3D) structure of a protein given its amino acid sequence. Substantial progress is being made in this problem with the help of AI tools such as deep learning. An improved ability to predict protein structure is providing new targets for therapeutic medicine. The most recent dramatic example of the use of AI for therapeutic medicine is in the context of vaccine discovery for COVID-19. AI was used, for example, to predict which of the tens of thousands of different pieces of the virus the immune system was most likely to recognize, enabling immunologists to focus their vaccine designs on a more manageable number of potential targets (Waltz 2020). In the case of Moderna, a biotechnology company, AI was used to predict the optimal mRNA sequence to provide the information needed to make a protein to attack the virus. Moderna accomplished this by January 13, 2020, only 2 days after the Chinese authorities posted the genome sequence of the virus online (Iansiti et al. 2021).

Our third example is small molecule drugs – a mainstay of therapeutic medicine. Scholars have estimated the space of potential small organic molecules to contain more than 10^{60} possible structures (Virshup et al. 2013).[13] Taking the target (or "lock") as given, the challenge is then to identify a ligand (or "key") that binds effectively and leads to the desired therapeutic effect. This screening challenge

[12] However, given the complexity of the process, researchers are making significant progress by developing predictive models of the relationship between DNA sequences and various intermediating "cell variables" (Leung et al. 2016). These cell variables can provide potential targets for therapeutic interventions. New measurement methods can give high-throughput data on various cell variables, and advances in machine learning are allowing researchers to take advantage of these abundant data to develop predictive models of their functioning. The combination of breakthrough gene editing technologies (notably CRISPR-Cas9) and a better understanding of the complex links between genes and disease could underpin major medical advances.

[13] These chemical combinations (or ligands) interact with targets (e.g., a protein) that regulate biochemical processes within the body. Disease can occur when these targets malfunction. By binding to a malfunctioning target protein, the ligand may be able to alter its adverse bioactivity. As combinations of chemical elements, these small molecules can take a vast number of forms leading to a massive combinatorial search space even for a single protein target.

can be aided by machine learning models of binding efficacy (see, e.g., Chen et al. 2018). Some recent approaches take advantage of knowledge of the three-dimensional structures of both the target proteins and the small molecule ligands. Machine learning models such as convolutional neural networks (CNNs) – initially applied in tasks such as image recognition – are being successfully applied to utilize this information to improve predictions of bioactivity for drug discovery applications (see, e.g., Wallach et al. 2015; Gomes et al. 2017).

Our final example comes from materials science. As with drug discovery, the space of potential molecules is vast. Researchers have used computational chemistry to virtually screen for the properties of molecules, including methods such as quantum chemistry and molecular mechanics. However, the computational costs of such simulation methods can be prohibitive, leading to increasing interest in statistical approaches such as machine learning to prioritize molecules for simulation- or experiment-based characterization (see, e.g., Pyzer-Knapp et al. 2015). For example, input data on molecular descriptors and output data on molecular properties could be used to develop a machine-learning-based predictive model of a large chemical space that would otherwise be prohibitively costly through computational and experimental methods.

Materials discovery – for instance, new materials for clean energy technologies or medical devices – fits well with the multi-stage search over a combinatorial search space characterization of the innovation process. The tasks involved in the discovery of new materials include the predictive screening of potential molecules, the making (or synthesis) of those molecules, the testing of the molecules using high-throughput methods and the characterization of their properties. Among the innovation challenges to which machine learning is being applied are the development of new catalysts to convert earth-abundant molecules (such as CO_2, H_2O, and N_2) into fuels and chemicals, new photovoltaic and thermoelectric materials, and new forms of batteries for energy storage (Tabor et al. 2018). However, researchers are concerned that bottlenecks in the discovery process severely slow the flow of new discoveries.

4 Conceptual building blocks of a combinatorial model of innovation

Our examples of the growing use of machine learning in scientific discovery and innovation reveal a wide range of types of input data, output data, and algorithms to produce prediction models. In this section, we put these details aside and set out the conceptual building blocks of a highly stylized model of the innovation process as search over a vast combinatorial search space.

4.1 The search space

The starting point is to conceptualize innovation as the combination of more basic elements such as genes or molecules (Romer 1992; Weitzman 1998; Arthur 2009). We represent a combination as a string with A elements. In the simplest case, the string reflects whether an element is present or not in the combination, with a 1

indicating presence and a 0 absence. More generally, each of the elements in a string can have multiple states, with the set of M states denoted as its alphabet. We can then represent the string underlying a given combination by the particular states of the elements that comprise that combination.

For a given innovator, the available elements determine their *combinatorial search space* – that is, the set of all possible combinations that can be formed. In the binary case ($M = 2$), for an innovator with A available elements the number of all possible combinations that can be formed from these elements is 2^A.[14] We henceforth assume that the binary case holds.

Innovation output can now be thought of as a function of the potential combinations: $I = F\left(2^A\right)$. More precisely, we think of innovation as resulting from search over the set of potential combinations. In conceptualizing this search space, we make use of the idea of a *fitness landscape* (see, e.g., Kauffman 1993). For a given fitness landscape, we associate each potential combination with a scalar that reflects its (fitness) value according to some particular property of interest.[15]

Central to the fitness landscape is a measure of distance. The distance between any two strings (or combinations) is the number of states that differ between those strings (or Hamming distance). For a given string, the 1-neighbor strings are all those strings that differ by just one state. The d-neighbor strings are all those strings that differ in exactly d of the states. The correlation structure of the landscape determines how correlated the values of the combinations are across different distances. Low correlations between the values of combinations outside close neighborhoods are associated with more "rugged" fitness landscapes.

The mapping from a combination to its scalar value can be viewed as an index function. For a combination to be successful – i.e., lead to an innovation – we assume its index value must be equal to or greater than some threshold. We can thus recast our landscape in a simplified form so that combinations with a value at or above that threshold have a value of 1 and combinations with a value below the threshold have a value of 0.

4.2 The ranking function

Our search process over the combinatorial search space follows a special case of the general ("Pandora's box") search model developed in Weitzman (1979). The decision maker searches by deciding on the order to open boxes (which corresponds to a combination advancing along the discovery pipeline in our application) and when to stop the search. In the general model, the decision-maker knows ex ante the probability distribution of outcomes, faces boxes with different opening costs and different time lengths before the contents of any opened box is revealed. In the context of sequential search, Weitzman shows that each box can be assigned a "reservation

[14] More generally, if each idea can take any one of M states, the total number of possible combinations is M^A.

[15] Value may be multidimensional, so there can be multiple landscapes, each one associated with a particular property of interest. However, we will typically assume a single dimension of technological fitness for convenience.

price" that depends on the unique characteristics of that box. He derives "Pandora's rule" as the optimal search strategy: continue to open boxes in descending order of the reservation price until the value of the best outcome achieved is greater than the reservation prices of all remaining unopened boxes.

Weitzman considers a special case that matches our basic setup: the outcome for a given box is either a success or a failure with a given probability of success for each box; the cost of opening a box is the same for all boxes; and the length of time before the outcome is revealed is irrelevant as there is no time discounting. In this special case, it is optimal to open all boxes with an expected net value greater than or equal to zero in descending order of the probability of success and to stop the search when a success is achieved (see Appendix 1). In addition to examining sequential versions of the search process, we also examine cases of parallel search, where the innovator must choose all the combinations to be advanced (or boxes to be opened) before any testing can take place (i.e., the actual opening of the boxes).

We now introduce a second resource available to the innovator: data. In addition to the available elements – the stock of which determines the combinatorial search space – the innovator has knowledge of previous successes (i.e., combinations with a value of 1) and previous failures (i.e., combinations with a value of 0). The total number of labelled successes and failures is D. These data can be used by innovator to develop a model (or map) of fitness landscape in the form of prediction model that outputs the probability of success of any given potential combination.

Our innovator is faced with the task of searching over a potentially vast combinatorial search landscape of $2^A - D$ potential combinations, which for notational simplicity we denote as N. A higher value of D reduces the number of combinations available to be discovered but also provides "training data" for developing a prediction model for new successful combinations.

This innovation search is therefore characterized by great uncertainty over the location of the valuable undiscovered combinations on the landscape, which are assumed to be small in number relative to the size of the search space. However, our innovator also has access to the prediction model to help detect the location of the undiscovered successes. As discussed further in Section 5, this prediction model could be provided by theory, simulation, parametric statistical data modelling, machine learning, or even educated guesses – although all methods will likely rely to some degree on observations of prior successes and failures.

Combinations are ultimately either successes (1) or failures (0), but without conducting the determinative test the innovator is uncertain as to which and must form an expectation of the probability of success prior to testing. If a successful innovation is brought to market after completion of the necessary screening and testing, it results in a payoff to the innovator of π, which we normalize to 1 without loss of generality. We seek a decision rule at Stage 0 to determine which combinations to advance in the pipeline. In determining this decision rule, we assume that the innovator is risk-neutral and has the objective to maximize expected total value (net of the cost of conducting later screens and tests) of innovation.

There is a "ground truth" – here the truth as revealed by a test – only imperfectly known to the innovator as to the location of undiscovered successes on the landscape. There are G successes in total so that successes as a share of potential

combinations is G/N, where G is a natural measure of the fecundity of search space. A useful graphical representation of this ground truth is the unit step function shown in Fig. 1. Denoting the known probability of success of the r^{th} ranked potential combination as p_0^r, the unit step function is:

$$\begin{aligned} p_0^r &= 1 \quad \text{for } r \leq \\ &= 0 \quad \text{for } r > G \end{aligned} \tag{1}$$

Figure 1 shows the both case where there is perfect ability to discriminate between successes and failures (where the internal rankings within the subsets of both successes and failure is arbitrary) and also the case where there is no ability to discriminate between successes and failures. In the latter case, the probability of success is simply the probability of finding a success as a result of a random draw from the search space, i.e.,G/N.

We seek a functional form for the ranking function such that the probability of discovering a success is equal to G/N when the prediction model has zero discriminating power and approaches the ground truth as the model approaches perfect discrimination. The following logistic decay function has the property that the ground truth is approached as $\beta \to \infty$:

$$p_0^r = \frac{1}{1 + Ke^{\beta(r-G-1)}}, \tag{2}$$

where $\beta \geq 0$. The value of K can be chosen so that the probability of success is equal to G/N when the prediction model has zero discriminating power (i.e.,$\beta = 0$):

$$\begin{aligned} p_0^r &= \tfrac{1}{1+K} = \tfrac{G}{N} \\ &\Rightarrow K = \tfrac{N-G}{G} \end{aligned}, \tag{3}$$

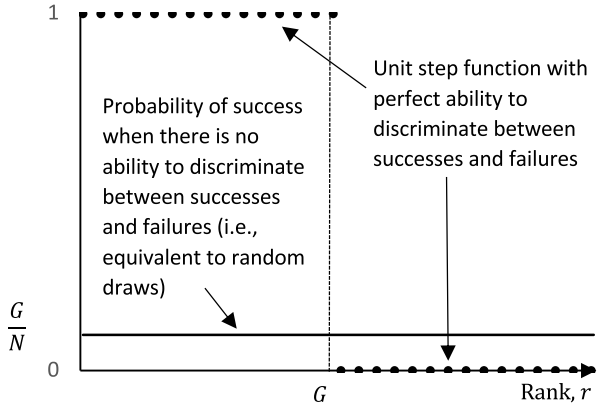

Fig. 1 Unit step function representing the ranking function for the ground truth. Notes: Where the innovator can perfectly discriminate between successes and failures, the ranking function takes the form of a unit step function. Within the subsets of successes and failures, the ranking is arbitrary. At the other extreme, where there is no ability to discriminate between successes and failures prior to testing, the ranking function is horizontal at the probability of finding a success based on a random draw from the search space

we thus chose this (appropriately restricted) logistic function as an analytically convenient representation of the ranking function. Furthermore, we assume that β monotonically increases with the performance of the prediction model. As shown in Fig. 2, the shape of ranking function curve varies from a horizontal line at G/N when $\beta = 0$, and converges to the ground truth unit step function as $\beta \to \infty$. Thus, the performance of the prediction model is controlled by a single parameter. Increases in β cause the ranking function curve to rotate in a clockwise direction around the point $(G + 1, G/N)$.[16]

Although an increase in β causes the probability of success to increase at any given rank from 1 to G, there is no presumption that the identity of the combinations at any given rank remains the same. Thus, for example, with the move to an improved prediction model we assume that the probability of success of the top-ranked combination will increase; however, the identity of the top-ranked combination could change, and indeed it is possible the probability of the previously top-ranked combination will fall.

A desirable feature of a ranking function is that the expected total number of successes equals G when aggregated over the entire space of N potential combinations. This is clearly the case when $\beta = 0$ (random search) and is approximately true as β goes to infinity (perfect discrimination). However, it is not generally true for intermediate values of β. We thus consider an augmented ranking function where a normalizing constant, $\theta(\beta)$, is added to Eq. (2), where the value of the necessary constant will depend on β (for given values of G, A *and* D) such that the sum of the probabilities of success equals G.

$$\begin{aligned} &\sum_{r=1}^{N}\left(\frac{1}{1+\frac{N-G}{G}e^{\beta(r-G-1)}}+\theta(\beta)\right)=G \\ &\Rightarrow \theta(\beta)=\frac{1}{N}\left[G-\sum_{r=1}^{N}\frac{1}{1+\frac{N-G}{G}e^{\beta(r-G-1)}}\right] \end{aligned} \tag{4}$$

[16] Numerous measures of performance exist for binary dependent variable models. Ideally, the measures of performance would be applied to a held back test sample given the risk of overfitting, especially for models that allow for highly flexible functional forms. One particularly intuitive measure for evaluating predictive performance both in and out of estimation sample is Tjur's coefficient of discrimination (Tjur 2009): $E[p(success)]|success) - E[p(success]|$failure. This coefficient varies from 0 to 1, with a coefficient of 1 indicating a perfectly discriminating model with the mean probability of success given the combination is an actual success equal to 1 and the mean probability of success given the combination is an actual failure equal to 0. When the mean probability of success estimated by the model is the same for actual successes and actual failures the coefficient has a value of zero. The area under the receiver operating characteristic curve (AUC) is also a widely used measure of the performance of a binary dependent variable model that can be usefully related to the accuracy of the probability of success rankings: the AUC can be interpreted as the probability that a randomly chosen actual success is ranked better (a lower number given our model) than a randomly chosen actual failure. In general, there is of course no reason that the probability rankings produced by a prediction model will follow a logistic curve. However, it is intuitive that a better performing model will tend to increase the estimated probabilities associated with better ranked combinations and decrease the estimated probabilities associated with poorly ranked combinations. In our model, an increase in the parameter β brings about the required clockwise rotation in the ranking function curve. This fact combined with the analytical convenience of the logistic form leads us to treat β as useful parameter to control the performance of the prediction model.

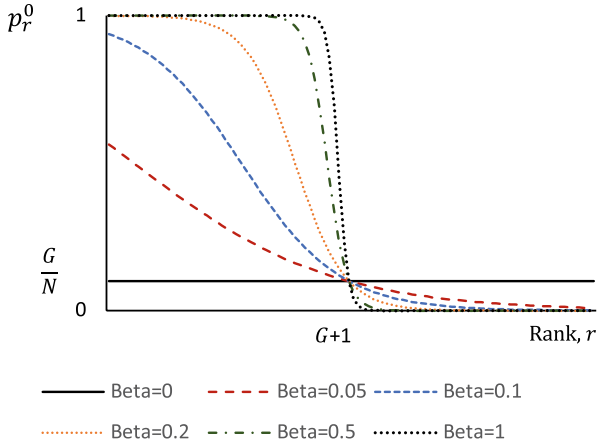

Fig. 2 Ranking function curves for different values of the discrimination parameter, *b*. Notes: The figure shows how the logistic ranking function changes as a result of an increase in β, which we equate with an improvement in Stage 0 prediction. The logistic ranking function is horizontal at the probability G/N when b is equal to zero. Increases in b cause the logistic ranking function to swivel in a clockwise direction around the point $(G+1, G/N)$. As β goes to infinity the ranking function will converge towards the unit step function representing perfect ability to discriminate between successes and failures

Given that the term in square brackets converges to a constant as N goes to infinity, the division by N ensures that $\theta(\beta)$ converges to zero. It follows that the non-normalized ranking function provides a good approximation to the augmented (normalized) ranking function for a sufficiently large search space. The non-normalized function also ensures that the probability of success lies in the [0,1] range and so we assume this function in what follows, recognizing that we are implicitly assuming a sufficiently large value of N so that any approximation error is negligible.

For considering the effects of improved prediction on expected search outcomes, it is useful to introduce an idea of dominance in comparing ranking functions. The swivel property of the ranking function ensures that as we increase β the probability of success at all ranks 1 to G will increase. Provided that the relevant range of ranking function is for probabilities of success greater than G/N, we can say that in comparing two ranking functions *R1* and *R2*, where *R1* has a higher value of β than *R2*, then *R1 dominates R2* over the relevant range. In the next section, we examine in the context of some simple search structures how an improvement in the prediction model – i.e., an increase in β – affects the search and the expected net value of innovation resulting from that search.

5 A two-stage (predict-test) example of searching with a map

In this section, we explore in the context of a two-stage example how access to an improved prediction model could affect the innovation process and its outcomes without for the moment considering the source of the improvement. We label the

stages 0 and 1. Stage 0 (prediction) is the development (or "training") of the prediction mode and the choice of probability threshold at or above which a combination is sent for testing (Stage 1). We conveniently represent the output of the prediction model as a ranking function mapping the rank (1 for the top ranked combination to N for the bottom ranked combination) to the model-based probability that the combination is a success. Stage 1 (testing) involves the conducting of the test, where we assume that the testing is a regulatory requirement that cannot be bypassed by taking a promising combination straight to market. The cost of a test is c_1.

Three cases of two-stage innovation search are now considered (see Fig. 3). In Case 1, the value of an innovation is independent of which other valuable innovations are discovered, combinations can be advanced in either sequentially or in parallel, and our risk-neutral innovator seeks to discover all valuable combinations with a positive expected value net of the cost of testing. In Case 2, the value to our innovator of finding additional successful combinations once one success is achieved is zero so that combinations that match the target are perfect substitutes. Combinations are again advanced in parallel but the innovator takes into account the probability that a success will be found with other combinations in the portfolio to be tested in deciding to add an additional combination to that portfolio. Finally, Case 3 most closely matches the Weitzman framework with sequential search and search stopping when a success is found. We explore both a version with identical payoffs conditional on success and identical testing costs and a version that allows for heterogeneity along one or both of these dimensions. We consider each case in turn focusing on how an improvement in the prediction model affects the decision to select combinations for testing and ultimately the expected total net value of innovation.

5.1 Case 1: Parallel search for innovations and innovations are independent

In our first case, our innovator seeks to discover as many as possible successes for a single target that is hidden in the search space. Using the analogy of "locks" and "keys" familiar from drug discovery, there is a single lock and G keys hidden in boxes spread across the search landscape. However, the boxes are costly to open, which forces the innovator to prioritize. We assume the keys are differentiated and it is possible to find multiple valuable keys to the lock with negligible competition between them, say different versions of a drug with the same therapeutic effect but appeal to different segments of the market.[17] It follows that the expected value of innovation is additive. We will dispense with the assumption of no-redundancy in Case 2 below.

To evaluate the expected value impact of an improved prediction technology it is useful to cast the testing decision in terms of a comparison of the expected marginal gross value and marginal cost of testing. We denote the optimal number of combinations to send for testing as r^*. Assuming an ordering of tests in terms of decreasing

[17] A more technical motivation comes from using a Dixit–Stiglitz–Ethier-type preferences/production function. With an additive objective function of this type, the demand curve for a new variety is unaffected by the discovery of an additional variety given the preference for variety built into these functions.

		Search Process	
		Parallel	*Sequential*
Substitutability	*Independent*	Case 1	
	Perfect Substitutes	Case 2	Case 3

Fig. 3 Cases of combinatorial search

expected marginal gross value of the innovation, the expected marginal gross value of the r^{th} test is simply:

$$MV_r^e = p_r^0. \tag{5}$$

The marginal cost of the r^{th} test is a constant, c_1.

The expected marginal gross value and marginal cost curves are illustrated in Fig. 4a, where we ignore the discrete nature of the problem for graphical simplicity. The subset of combinations that will be sent for testing comprises those combinations with an expected marginal gross value greater than or equal to marginal cost. If we make the additional simplifying assumption that at the last combination sent for testing marginal value and marginal cost are equated, then the probability threshold is given by:

$$p_{r^*}^0 = c_1. \tag{6}$$

Using Eq. (2) – our logistic ranking function – allows us to solve for the optional number of tests. All tests where the expected marginal gross value is greater than or equal marginal cost will be conducted:

$$\begin{aligned} &\frac{1}{1+\left(\frac{N-G}{G}\right)e^{\beta(r^*-G-1)}} = c_1 \\ &=> r^* = G + 1 - \frac{ln\left(\frac{N-G}{G}\right) - ln\left(\frac{1}{c_1}-1\right)}{\beta} \end{aligned} \tag{7}$$

More generally, the probability threshold (with associated r^*) will be the lowest probability in the ranking at which $MV_r^e \geq MC$.

Expected total net value is then:

$$V^e = -c_1 r^* + \sum_{r=1}^{r^*} p_r^0 = -c_1\left(G + 1 - \frac{1}{\beta}\left[ln\left(\frac{N-G}{G}\right) - ln\left(\frac{1}{c_1}-1\right)\right]\right) + \sum_{r=1}^{r^*}\left(\frac{1}{1+\left(\frac{N-G}{G}\right)e^{\beta(r-G-1)}}\right). \tag{8}$$

From Fig. 4b we can see the impact on the number of combinations that will be sent for testing of an improvement in the prediction model – i.e., an increase in β. Provided $c_1 > (G/N)$, the MC curve will intersect the MV^e curves above the crossing point of the MV^e curves at G/N. We assume that this condition holds, recalling that the number of undiscovered combinations is assumed to be small compared to the size of the combinatorial search space. The number of combinations sent for testing will be a non-decreasing function of β and strictly increasing for a large

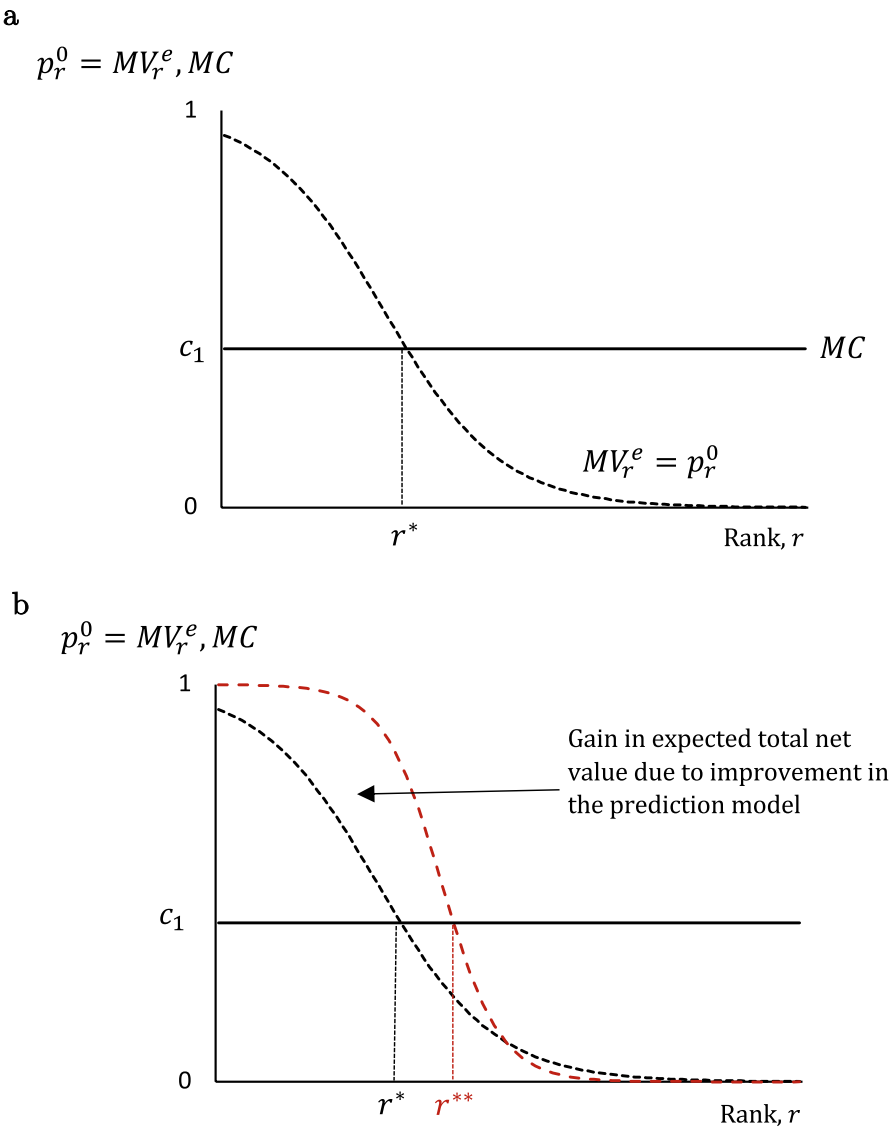

Fig. 4 **a** Determination of the optimal number of combinations to advance to testing. **b** Impact of an improvement in the prediction model on the optimal number of combinations to advance to testing

enough increase in β for it to be optimal to send at least one additional combination for testing.

Of most interest is what happens to the expected total net value of innovation, V^e, when there is an improvement in the prediction model. At the original optimal number of tests, r^*, the marginal expected gross value will be greater with the new (higher β) prediction model than with the original model (see Fig. 3b). It follows that V^e

must be higher even if the number of combinations going for testing did not change. However, any increase in the optimal number of tests from r^* to r^{**} as a result of the improved prediction model will lead to a further increase in V^e. The (approximate) overall increase in V^e is shown as the relevant area under the curve in Fig. 3b.

5.2 Case 2: Parallel search for innovations and Innovations are perfect substitutes

Case 1 made the strong assumption that the value of alternative innovations are independent and our risk-neutral innovator seeks all innovations that yield a positive expected net value. However, in many innovation search problems the innovator may be looking for a single combination that meets a particular target such as a small molecule drug that binds with a target protein to improve its functioning or a material for a battery with a desired property. Bringing multiple innovations to market that achieve the same target will be wasteful to the extent that these innovations are perfect substitutes.

Returning to our analogy of locks and keys, there is now a single target and G possible keys that open that lock. As with Case 1, the innovator must choose which boxes to open prior to opening any of the boxes (so it remains a case of parallel search), but in adding an additional box to the portfolio of boxes to be opened, the innovator takes into account that already included boxes in the portfolio may lead to the finding of the key that opens the box, making the additional box redundant ex post.[18]

The difference from Case 1 comes in the position and shape of the expected marginal gross value curve: as our innovator evaluates the expected marginal gross value of an additional test they must consider the probability that a combination meeting the target would have been discovered with the existing tests. The expected marginal gross value curve is now:

$$\begin{aligned} MV_r^e &= p_r^0 \text{ for } r = 1 \\ &= \left(\prod_{j=1}^{r-1}\left(1 - p_{r-j}^0\right)\right)p_r^0 \text{for } r = 2, 3, \ldots, 2^A - D \end{aligned} \tag{9}$$

For the marginal test, r, the term inside the first round brackets gives the probability that the desired target combination will not have been discovered in the previous $r-1$ tests. For example, for the third test, the probability that the target will not have been discovered in tests 1 and 2 is $\left(1-p_1^0\right)\left(1-p_2^0\right)$. The probability $\left(1-p_1^0\right)\left(1-p_2^0\right)p_3^0$ is thus the probability that target will be found in the third test and not before.

The marginal cost of a test is again equal to c_1. As illustrated in Fig. 5a, the optimal number of combinations to send for testing is identified by moving down the

[18] We assume that the model of the search landscape is given. However, if the model could be rerun before choosing the next combination to be added to the testing portfolio test (i.e., choosing the next additional box to open), the innovator could rerun the model based on the assumption that the previous combinations in the portfolio were failures and thereby improve the portfolio selected through incremental additions of combinations.

probability ranking and testing all combinations for which the expected marginal gross value is greater than or equal to the marginal cost. This gives the optimal number of tests, r^*, and the expected total net value is:

$$V^e = -c_1 r^* + \sum_{r=1}^{r^*} \left(\left(\prod_{j=1}^{r-1} \left(1 - p^0_{r-j}\right) \right) p^0_r \right). \tag{10}$$

Figure 5b, c shows the impact on the optimal number of tests when the innovator gets access to an improved prediction model, again captured by an increase in β. As can be seen from the figure, the marginal value curves (pre and post improvement) will cross at a certain point in the ranking. The impact of an improvement in the prediction model will depend on whether marginal cost, c_1, is above or below marginal value at this crossover point. Figure 4b shows a case where the crossover point occurs below c_1. In this case, the optimal number of tests will increase.

Figure 4c shows a case where the crossover occurs above c_1 and the optimal number of tests will decrease.

What is the effect of an increase in β on V^e? Provided p^0_r is greater than G/N, we can show that V^e increases regardless of whether the optimal number of tests rises or falls. To see why, note that for a higher value of β the probability of success is higher for each position in the ranking. Comparing the probability that a success will have been found at the r^{th} ranked combination with and without the improvement in the prediction (where a prime indicates the post-improvement probability at a given rank):

$$1 - \prod_{j=1}^{r} \left(1 - p^{0\prime}_{r-j+1}\right) > 1 - \prod_{j=1}^{r} \left(1 - p^0_{r-j+1}\right). \tag{11}$$

Given the improved prediction model, the probabilities on the left-hand side all greater than the probabilities on the right-hand side at a given rank. Thus, the cumulative probability that a success will have been discovered is also higher for every position in the ranking. In Fig. 5b, V^e is thus higher with the higher β at the original optimal number of tests, r^*. The move to the higher optimal number of tests r^{**} increases V^e still further. A similar situation arises in Fig. 4c even though the optimal number of tests now falls rather than rises. At the original number of tests V^e is again higher with the higher β. The move to the optimal number of tests (fewer tests in this situation) further increases this gain relative to the situation with the inferior prediction model (i.e., lower β).

5.3 Case 3. Sequential search where successful innovations are perfect substitutes

Our first two cases assume identical payoffs (gross value) across combinations conditional on a successful test and identical testing costs for all combinations. Both cases also involve parallel rather than sequential search. Our third case more closely follows the Weitzman structure with potentially combination-specific payoffs, π_i, conditional on a successful test and combination-specific second stage testing costs, $c_{1,i}$. The innovator

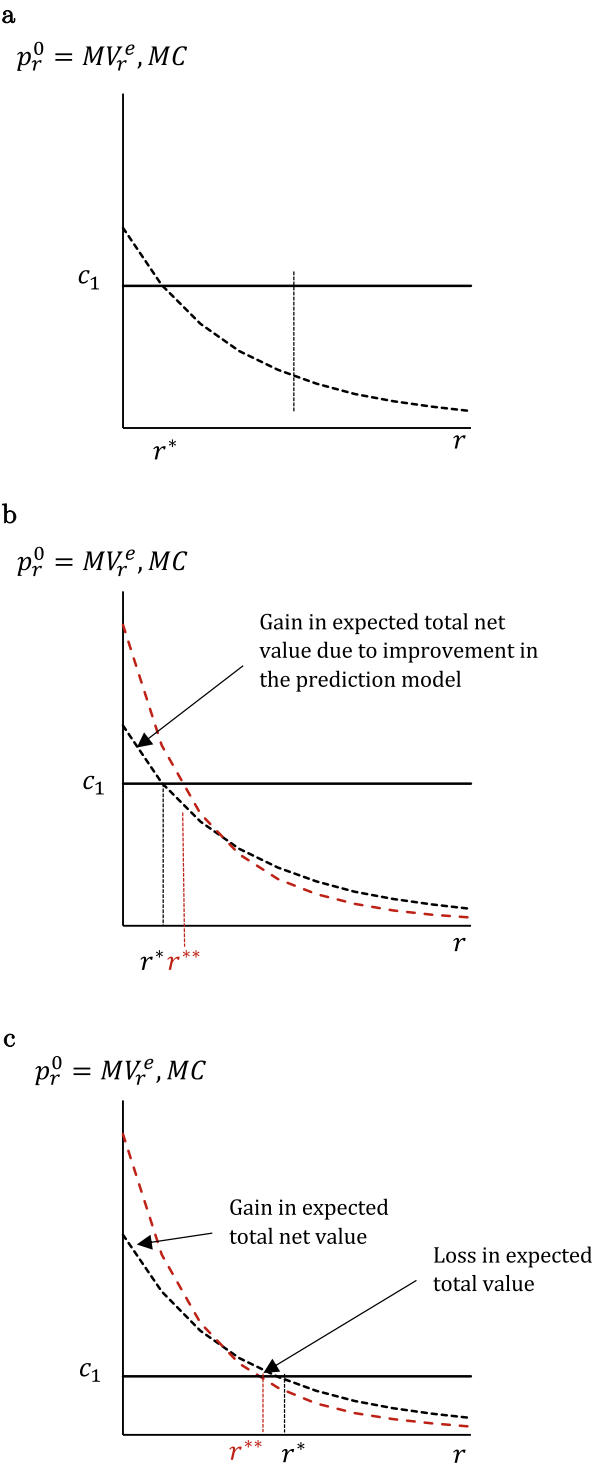

Fig. 5 **a** Optimal number of tests with a single innovation target. **b** Impact of an improvement in the prediction model on the optimal number of tests when the innovator has a single innovation target and the crossover probability is below c_1. **c** Impact of an improvement in the prediction model on the optimal number of tests when the innovator has a single innovation target crossover probability is above c_1

is seeking a match for a single target and search is sequential, so that a test must be completed before moving on to the next combination in the search order. As shown in Weitzman (1979), the optimal ranking of combinations (or boxes) in a sequential search problem is based in this setting on their "reservation price", z_i, given by:

$$\begin{aligned} & p_i^0 \pi_i + \left(1 - p_i^0\right) z_i - c_{1,i} = z_i \\ & => z_i = \pi_i - \frac{c_{1,i}}{p_i^0}. \end{aligned} \tag{12}$$

The first equation can be viewed as a comparison between a "sure thing" available to the innovator with payoff, z_i, and a lottery where the sure thing is available as a backup option. For a given combination, z_i is determined as the value that makes the innovator indifferent between the two alternatives. The innovator will sequentially search in decreasing order of reservation prices and stop the search when a success is achieved (see Appendix 1).

For ease of comparison with Cases 1 and 2, we first consider the special situation where combinations differ only in their probabilities of success (i.e., payoffs conditional on success and testing costs are the same across combinations). As with the other cases, we examine how an improvement in the prediction model (i.e., an increase in β) affects the combinations that advance for testing and the expected value of innovation. However, in contrast to parallel search, the number of combinations that the innovator will advance for testing (or, equivalently, the duration of search) is uncertain at the outset of testing.

We first compare the expected probability that a success will be found in a search with maximum duration, r^{max}, before and after the improvement in the prediction model, where the post improvement probabilities are again indicated with a prime:

$$1 - \prod_{j=1}^{r^{max}} \left(1 - p'^0_{r^{max}-j+1}\right) > 1 - \prod_{j=1}^{r^{max}} \left(1 - p^0_{r^{max}-j+1}\right). \tag{13}$$

Therefore, if there is no change in the maximum duration of search, the expected probability of success from the search will increase following the improvement in the prediction model.

Turning now to the expected duration of search, this expected duration (i.e., the expected number of combinations to be tested) is equal to the probability that the search will involve exactly r combinations times r, summed from $r = 1, \ldots, r^{max}$, plus the probability that the search will end with no success being found times r^{max}.

$$L^e = \sum_{r=1}^{r^{max}} \left(\left(\prod_{j=1}^{r-1} \left(1 - p^0_{r-j}\right) \right) p^0_r \right) r + \left(\prod_{j=1}^{r^{max}} \left(1 - p^0_j\right) \right) r^{max} = \sum_{r=1}^{r^{max}} \left(\prod_{j=1}^{r-1} \left(1 - p^0_{r-j}\right) \right). \tag{14}$$

The last equality uses a classic result in discrete survival analysis whereby the expected duration of a search with maximum length r^{max} is equal to the sum of the lagged survival functions up to r^{max} (see Agrawal et al. 2023, Appendix 2). Holding r^{max} constant, we can therefore see that the expected duration of the search will fall following an improvement in the prediction model:

$$\sum_{r=1}^{r^{max}}\left(\prod_{j=1}^{r-1}\left(1-p_{r-j}^{0\prime}\right)\right)<\sum_{r=1}^{r^{max}}\left(\prod_{j=1}^{r-1}\left(1-p_{r-j}^{0}\right)\right). \tag{15}$$

Putting the pieces together, we can see that if the innovator is restricted to the same maximum duration of search, an improvement in the prediction model will lead to an increase in the expected total net value of the search, as both the expected probability of success (and thus the expected value of the search) will rise and the expected duration of the search (and thus the expected cost of the search) will fall. Of course, the change in prediction model may also change the optimal maximum duration of the search. However, any re-optimization of this optimal maximum duration will lead to a further increase in expected total net value. Thus, as with Cases 1 and 2, an improvement in the prediction model (i.e., a higher value of β) will lead to a rise in expected total net value of the innovation search.

With sequential search, it is interesting to see the possible effects of relaxing the homogeneity assumption and instead allow for combinations to vary in terms of the payoff conditional on success (we continue to assume the testing cost is the same for all combinations). To see the potential implications of an improvement in the prediction model under such heterogeneity, note first that the partial derivative of the reservation price that determines the search ordering with respect to the probability of success is:

$$\frac{\partial z_i}{\partial p_i^0}=\frac{c_{1,i}}{\left(p_i^0\right)^2}. \tag{16}$$

The impact of a small change in the probability of success for a combination is thus decreasing in the initial probability of success.

To see how an improvement in the prediction model could lead to a shift in the sequential search ordering towards "riskier" combinations, we consider two combinations, i and j, with identical initial (i.e., pre-improvement in the prediction model) reservation prices (i.e.,$z_i = z_j$), but where $\pi_i < \pi_j$ and $p_i^0 > p_j^0$. Thus, box j is riskier in the sense that it has a lower probability of success but a higher payoff conditional on success. To bias things against the riskier combination we further assume that $\Delta p_i^0 > \Delta p_j^0 > 0$ as a result of the improved prediction model. The following approximations hold for relatively small increases in the predicted probabilities:

$$\Delta z_i \approx \frac{c_1}{\left(p_i^0\right)^2}\Delta p_i^0, \tag{17}$$

$$\Delta z_j \approx \frac{c_1}{\left(p_j^0\right)^2}\Delta p_j^0. \tag{18}$$

However, despite combination i having the larger increase in the probability of success, combination j – the riskier combination – will actually move ahead in the ranking when:

$$\begin{aligned} &\Delta z_j > \Delta z_i \\ &\Rightarrow \frac{p_i^0}{p_j^0} > \frac{\frac{\Delta p_i^0}{p_i^0}}{\frac{\Delta p_j^0}{p_j^0}}, \end{aligned} \tag{19}$$

That is, box j will move ahead in the ranking if the ratio of proportionate changes in the probability of success is less than the initial ratios of the probabilities of success. Therefore, it is possible for riskier combinations to move ahead in the ranking despite experiencing a smaller increase in their probability of success as a result of the improved prediction model. This suggests another possible effect of improved prediction: improved prediction leads to riskier combinations moving along the discovery pipeline, enhancing the chances of more radical innovations being discovered.

6 Making predictions: Advances in AI as a shock to the innovation process

In this section, we further discuss AI as the source of the shock to the innovation process. Our generic workflow in the multi-stage model is initial prediction followed by various screens and tests based on the sequential refinement of the predicted probability of success. There are a number of ways to generate such Stage 0 prediction models for the landscape. A useful distinction is between theory- and simulation-based approaches on one hand and data-based approaches on the other. For reasons to be explained below, we treat machine learning as a *subset* of the data-based approach. While data is obviously also central in the other stages (say through the use of controlled or natural experiments), the focus here is on how data is used to generate prediction models.

In order to better delineate the role of machine learning as a prediction tool, we first underline the role of theory in generating predictions in Popper's classic account of the scientific method (Popper 1959). However, we take it that the central requirement for science is that predictions can be tested and leave open the source of those predictions to include data-based approaches. Theory, of course, remains a major source of predictions even with the advance of AI. In theoretical chemistry, for example, the Schrödinger equation remains central to the predictions of molecular properties. Given the complexity of predictions beyond extremely small molecules, various approximations such as the Born-Oppenheimer (1927) approximation or Hartree–Fock theory are used (Szabo and Ostlund 1996). However, even with these approximations the complexity of the calculations requires the use of (typically costly) computer simulations.

The second broad method of prediction prior to testing is data-based prediction generation. As noted, we think of machine learning as a subset of data-based prediction – a subset for which there has been recent rapid progress. In distinguishing this subset, we find it useful to make use of Leo Breiman's (2001) contrast between "two cultures" of statistical modeling, which he labels the "data

modeling culture" and the "algorithmic modeling culture." We identify the latter with what is commonly referred to as machine learning, although we recognize that there is no widely accepted division between what is part of traditional statistics and what is part of machine learning. Breiman outlines what he sees are the broad differences between the two approaches, but a key feature for our purposes is that the former, typically informed explicitly by scientific theory, uses a parametric data model. This model will typically specify the dependent variable, the predictor variables, the functional form, and the stochastic form of the disturbance term in the model.

We now consider the effect of having access to machine learning as an alternative tool for statistical modeling of the landscape. Breiman (2001, p. 205) describes the alternative "algorithmic" approach as follows:

> The approach is that nature produces data in a black box whose insides are complex, mysterious, and, at least, partly unknowable. What is observed is a set of **x**'s that go in and a subsequent set of **y**'s that come out. The problem is to find an algorithm f(**x**) such that for a future **x** in a test set, f(**x**) will be a good predictor of **y**.

The data-generating processes in many combinatorial-type problems – drug discovery, materials science, genomics, etc. – do appear to fit the description of "complex, mysterious and, at least, partly unknowable." To the extent that the machine learning approach provides a better prediction model (in at least some circumstances) for valuable new combinations that are distributed over a vast and complex search space, it would appear to have the potential to boost the productivity of the innovation search process.[19]

How does machine learning fit our generic predict-screen-test workflow? We assume that part of the knowledge base consists of data on previous experiments that we treat for simplicity as indicating tested combinations were successes or failures. These binary outcome data together with input data on the combinations (amino acid sequences, molecular descriptors, etc.) are the training data for our supervised machine learning prediction model. The measure of a good prediction model will be how well it predicts (or generalizes) outside of the training sample. An improvement in the prediction model is captured parametrically in our innovation search model simply as an increase in the β parameter.

Of course, there is a vast – and rapidly growing – array of available machine learning algorithms. To give a flavor of their use in generating prediction models we note just a selection here and relate them to our generic workflow of classifying potential combinations into predicted successes and failures. Probably the most intuitive algorithm is k-nearest neighbors. The predicted probability of success of a

[19] It is important to note that textbooks on machine learning typically subsume parametric regression and classification models as forms of machine learning. However, we find Breiman's distinction to be useful in highlighting the shock to the innovation process that the rapid advance in machine learning has engendered, and take his narrower category to be what is meant by machine learning (see also Athey and Imbens 2019).

candidate combination is simply the average success rate of the k nearest neighbors in the search space. Decision tree methods work by segmenting the search space into success and failure regions. The more complex decision tree methods used in practice (e.g., random forests) involve multiple trees that are combined together to produce the predicted probability of success. A third example is the naïve Bayes classifier. In contrast to the approaches noted already, this approach concentrates on the individual "features" – the states of an element of the string describing a combination in our generic example – which are each assumed to have an independent effect on the success of a combination. The output is an estimate of the probability of success conditional on the states all the elements in the string describing that candidate combination.

In addition to the increased availability of data and computing power, much of the recent excitement concerning machine learning is due to rapid improvement in algorithms. Of particular note is the rapid improvement in prediction models based on artificial neural networks, most notably so-called deep learning algorithms.

> Deep learning is making major advances in solving problems that have resisted the best attempts of the artificial intelligence community for many years. It has turned out to be very good at discovering intricate structures in high-dimensional data and is therefore applicable to many domains of science, business, and government. (LeCun, et al. 2015, p. 436.)

Although an in-depth discussion of the technical advances underlying deep learning is beyond the scope of this paper, three aspects are worth highlighting. First, the development and optimization of multilayer neural networks allows for substantial improvement in the ability to predict outcomes in high-dimensional spaces with complex non-linear interactions (LeCun et al. 2015).[20] Second, given that previous generations of machine learning were constrained by the need to extract features (or explanatory variables) by hand before statistical analysis, a major advance in machine learning involves the use of "representation learning" to automatically extract the relevant features.[21] Third, recent optimism about developments in deep

[20] For example, a review of the use of deep learning in computational biology notes that the "rapid increase in biological data dimension and acquisition rate is challenging conventional analysis strategies," and that "[m]odern machine learning methods, such as deep learning, promise to leverage very large data sets for finding hidden structure within them, and for making accurate predictions" (Angermueller et al. 2016, p.1). Another review of the use of deep learning in computational chemistry highlights how deep learning has a "ubiquity and broad applicability to a wide range of challenges in the field, including quantitative structure activity relationship, virtual screening, protein structure prediction, quantum chemistry, materials design and property prediction" (Goh et al. 2017).

[21] As described by LeCun et al. (2015, p. 436), "[c]onventional machine-learning techniques were limited in their ability to process natural data in their raw form. For decades, constructing a pattern-recognition or machine-learning system required careful engineering and considerable domain expertise to design a feature extractor that transformed the raw data (such as the pixel values of an image) into a suitable internal representation or feature vector from which the learning subsystem, often a classifier, could detect or classify patterns in the input. . . . Representation learning is a set of methods that allows a machine to be fed with raw data and to automatically discover the representations needed for detection or classification."

learning relates to demonstrated out-of-sample performance of deep learning models across a range of tasks, including image recognition, speech recognition, language processing, and autonomous vehicles.[22]

Notwithstanding that the most publicized successes of deep learning have been in areas such as games of strategy including chess and go, the winning of image recognition competitions such as ImageNet, and, most recently, in the availability of high-performing large language models (LLMs) such as GPT4, parallels to the way in which the new methods work on unstructured data are increasingly being identified in many fields with similar data challenges to produce research breakthroughs.[23] While these new general-purpose research tools will certainly not replace traditional theory, simulation, and statistical data modeling – nor the researcher's intuition – as methods for developing predictive models of fitness landscapes, machine learning methods such as deep learning appear to offer powerful new tools for prediction, especially where the complexity of the underlying phenomena present obstacles to more traditional methods.[24]

7 Conclusion

AI has achieved well-documented recent successes in tasks such as image recognition, speech recognition, language translation, recommendation systems, and autonomous vehicles. It is now viewed as a GPT for prediction tasks that can be combined with other components to produce novel products and services (Agrawal et al. 2018). The machine-learning-based prediction models have the common feature that they can deal with vast combinatorial spaces (e.g., pixels in an image). This paper has focused on machine learning as a GPT for use in the discovery process itself. Conceptualizations of the innovation process as search over a combinatorial space suggests the value of machine learning in helping to map this space in the form of a prediction model.

The burgeoning scientific literature applying machine learning tools suggests the practical value of being able to make predictions when the number of potential combinations can be in the billions. However, prediction is just one part of

[22] While scholars consider deep learning models something of a black box due to their complexity, recent theoretical work has made progress in understanding why these models have had such success in generalizing beyond their training data (see, for example, Shwartz-Ziv and Tishby 2017).

[23] A recent review of deep learning applications in biomedicine usefully draws out these parallels: "With some imagination, parallels can be drawn between biological data and the types of data deep learning has shown the most success with – namely image and voice data. A gene expression profile, for instance, is essentially a 'snapshot,' or image, of what is going on in a given cell or tissue in the same way that patterns of pixilation are representative of the objects in a picture" (Mamoshina et al. 2016, p. 1445).

[24] A recent survey of the emerging use of machine learning in economics (including policy design) provides a pithy characterization of the power of the new methods: "The appeal of machine learning is that it manages to uncover generalizable patterns. In fact, the success of machine learning at intelligence tasks is largely due to its ability to discover complex structure that was not specified in advance. It manages to fit complex and very flexible functional forms to the data without simply overfitting; it finds functions that work well out of sample" (Mullainathan and Spiess 2017, p. 88).

the discovery process. We thus embedded the prediction task as just one part of a costly multi-stage process. In our model of AI-aided discovery, the prediction model produced a ranking function that essentially provided a priority list for later (costly) screening and testing. Improvements in the prediction model – say as a result of the availability of a better performing algorithm – allowed for a more discriminating prioritization and ultimately for a more productive discovery pipeline.

The main testable hypothesis from the model is therefore that access to AI will increase the productivity of the innovation process for combinatorial-type problems through improved prioritization. Although there has been rapid growth in the use of machine learning – and a great deal of optimism expressed about its effects – the productivity benefits have yet to be well established (Brown et al. 2020). Indeed, there is skepticism about its ultimate benefits, in part reflecting the previous waves of optimism and pessimism associated with AI. There have also been well-publicized cases where early optimistic scientific findings have been reversed and exits of high-profile research groups. Skeptics have a number of concerns, including: limited and error-prone data leading to poorly performing models; failure to replicate results; excessive concentration on already well-explored regions of the search space where data are plentiful; difficulties of interpretation of "black box" models and associated issues of trust; and models that appear to work well on test data sets but ultimately perform poorly due to redundancy between training and test data (Bender and Cortés-Cirano 2020; Wallach and Heifets 2018).

The next step in our research program is therefore to look for evidence of the productivity effects of increased access to AI-aided discovery. This is obviously complicated by the relative newness of some of the main developments (including the improved performance of deep neural network algorithms). However, we hope it will be possible to exploit plausibly exogenous variation in the access to AI tools. One idea we are exploring is to use variation in historic expertise at the department level in machine learning in computer science and engineering departments, and to see how it relates to later output in application domains within the university (medicine, chemistry, etc.) based on assumption of local knowledge spillovers. Another avenue is to look for access shocks to AI-prediction technologies and see how they differentially affect the productivity of researchers.

Appendix 1 Optimality of the Weitzman ordering over Bernoulli boxes

In this appendix, we provide a simple proof of the optimality of the Weitzman ordering for searching Bernoulli boxes. Each box, i, is described by a probability that it contains a success, p_i^0, a payoff conditional on success, π_i, and a cost of searching the box, $c_{1,i}$. We assume that p_i^0 and $c_{1,i}$ are strictly greater than zero and that there is no time discounting. As set out in the text, each box has a "reservation price," $z_i = \pi_i - \frac{c_{1,i}}{p_i^0}$. The reservation price has the interpretation of the "sure-thing" that has an expected payoff equal to the lottery of opening the box (at a cost) where there is a

back-up option equal to the sure-thing that is payable in the event that the opening of the box does not yield a success. In our setting, we interpret the opening of a box as the conduct of a costly test that determines whether it contains a success or not. Applied to Bernoulli boxes, the Weitzman result is that the expected value of the search for the best alternative is maximized by sequentially searching the boxes with an expected net value greater than zero in declining order of reservation prices and stopping the search once a success is achieved.

The optimality of this search strategy can be demonstrated by considering the optimal pairwise ordering of any two boxes i and j, where we initially assume that $z_i > z_j$. From the definition of the reservation price, we can immediately see that this implies that $\pi_i > z_j$. Therefore, if a success is found on opening box i the search will stop. The intuition is that the payoff from box i becomes the sure-thing for the lottery of opening box j and the sure-thing is greater than required for indifference between taking the sure thing and playing the (costly) lottery of opening box j.

The key question is whether it is optimal to open box i first. Contrariwise, consider the strategy of opening box j first. There are two possibilities assuming a success is found on opening the box. First, if $\pi_j < z_i$ it will not be optimal to stop the search and the search will continue to box i. Of course, since $\pi_i > \pi_j$ given $z_i > z_j$ and $\pi_j < z_i$, box i would always be chosen if a success is achieved with both boxes. Therefore, it would always be optimal to first open box i and only continuing to box j if a success is not found and otherwise saving on the search cost for box j if a success is found for box i.

Second, and more interesting, is the case where $\pi_j > z_i$ so that search would stop if a success is found on first opening box j. The choice facing the innovator is now whether to first search box i and stop if a success is found or to first search box j and stop if a success is found. The optimal order to open these boxes is found by comparing the expected value of the two strategies. The expected value of opening box i will be greater when:

$$p_i^0(\pi_i - c_{1,i}) + (1 - p_i^0)\left(p_j^0\pi_j - c_{1,i} - c_{1,j}\right) > p_j^0(\pi_j - c_{1,j}) + \left(1 - p_j^0\right)(p_i^0\pi_i - c_{1,j} - c_{1,i}) \Leftrightarrow z_i > z_j, \tag{20}$$

where the latter implication is derived by simple algebraic manipulation of the first inequality and the definition of the reservation price. Therefore, the strict ordering of the reservation prices in favor of box i is a necessary and sufficient for a strict preference for first searching box i. In the case where the reservation prices are equal, the expected value will be independent or the order in which the boxes are searched and the innovator will be indifferent as to the order of search. In this case, we assume that the order of the search is chosen randomly. (Note that $\pi_i > z_j$ and $\pi_j > z_i$ under indifference so that it will be optimal to stop the search upon finding a success no matter which of the boxes is searched first.) As such pairwise comparisons can be applied to every pair of boxes, we obtain, as required, a complete (and transitive) optimal search ordering of the full list of relevant boxes based on their reservation prices with search stopping once a success is achieved.

Appendix 2 Extensions to a two-stage (predict-test) example of searching with a map

Multi-objective-optimization (MOO)

Our basic model implicitly assumed a single objective (e.g., the binding efficacy of a small molecule drug with a given protein target). More realistically, a successful combination will have to meet multiple objectives – e.g., the drug is non-toxic in addition to efficacious binding. One straightforward extension is to assume that a combination must meet all requirements to be considered a success. That is, it must achieve a value of 1 on all necessary dimensions so that the overall indicator of success is the product of the individual success indicators. We then assume that the logistic ranking function applies to the overall probability of success.

A less strict approach where objectives are measured continuously is to identify a Pareto frontier whereby one objective (say non-toxicity) cannot be increased without sacrificing another objective (say efficacy). The trade-offs between objectives might then be considered at a later stage of the discovery process (e.g., lead optimization). We discuss a multi-stage extension of our two-stage model that allows for multiple intermediate screening stages prior to final determinative testing below.

Multi-task and transfer learning

One challenge in applying AI-based prediction to tasks such as discovering a small-molecule drug that is effective against a given target is the sparsity of relevant data. This has led to interest in techniques such as multi-task learning where the prediction model is estimated based on a vector of outcome variables.

Transfer learning refers to the ability to transfer knowledge from the use of data in related domains in the generation of the prediction model for new innovation tasks. We have already implicitly used the idea of transfer learning in the cases analyzed above in that related successes must be used in generating the prediction model for targets that by definition have not yielded a success.

Active learning

In our model, the innovator is only focused on the *exploitation* of the prediction model of the search space so as to discover new valuable combinations. Therefore, the choice of combinations to send for testing in both the cases discussed is solely based on the predicted probabilities of success of those combinations. However, another goal in selecting combinations to send for testing might be the improvement of the prediction model for future innovation search tasks. The optimal *exploration* might lead to different choices in terms of the ranking of combinations to send for testing. For example, the innovator may want to explore relatively unknown parts of the search space even if the benefit for current tasks is limited. In the context of an ongoing innovation effort, there can therefore be a trade-off between the goals of exploitation and exploration. An "active learning" strategy directly addresses this

trade-off between the immediate innovation value and future information value of testing.

A possible limitation of passive learning strategies based on exploitation of the existing prediction model is that there will be overconcentration on known parts of the combinatorial search space. In the context of, say, drug or materials design, active learning strategies have included techniques such as *uncertainty sampling* – testing combinations predicted with low confidence by the model – and *model-improvement* – testing combinations based on predictions of the how the new data point will improve the performance of the prediction model (Reker and Schneider 2014). Active learning strategies "assist the selection process by focusing on areas of chemical space that have the greatest chance of success while considering structural novelty. The core feature of these algorithms is their ability to adapt the structure-activity landscapes through feedback" (Reker and Schneider 2014, p. 458).

Discriminative versus generative models

We have modelled the prioritization process as an effort to *discriminate* between potential combinations based on their predicted probability of success. A complementary approach aims to directly "design" a combination – say a small molecule drug – that meets specific requirements using a *generative* approach (see e.g., Merk et al. 2018). The generative approach effectively inverts the process, whereby the researcher starts with the desired properties and uses the model to identify the combination that comes closest to achieving these properties. Such generative modelling using AI falls into the broader category of *de novo* design. These approaches may be especially important in the context of refinements of rankings (such as lead optimization in the case of drug discovery):

> While compound elimination by appropriate scoring models discards the bulk of the designs ("negative design") with acceptable accuracy, the selection of the best or most promising ("positive design") remains prone to error. More accurate activity prediction models that extend the capabilities of existing approaches could originate from advanced machine learning methods. (Schneider 2018, p. 109.)

Appendix 3 A multi-stage discovery process and with bottlenecks

Extension of the model to include multiple intermediate screening stages with Bayesian updating between stages

We assumed in Section 5 a highly simplified two-stage innovation process involving prediction (development of a prediction model choosing a testing threshold) and testing (testing all combinations with a probability of success at or above a threshold value that depended on the market value of an innovation success and the cost of conducting the test). The motivation for this simple setup is that the discovery

pipeline often involves the production of a priority list for testing where such testing is expensive. AI can be introduced as a possible way to improve the list.

Of course, in reality innovation processes are more complex and involve multiple stages rather than just two. The typical stages of the drug discovery process include target identification, hit generation, generation of lead compounds from the hits ("hits to lead"), optimization of the lead compounds, pre-clinical trials (animal studies), and phase I, II and III human clinical trials. In materials discovery, the stages can involve prediction of molecule properties, synthesis of the molecules, and characterization of the actual properties through testing, but this can be followed by further stages such as investigating the ability to synthesize at scale and testing the molecule under the different environmental conditions that could be observed in the field.

We therefore extend our two-stage prediction-test model in this section to a multi-stage setting. Our main focus is on the implications of bottlenecks in later stages of the discovery pipeline. The basic two-stage model already highlights one cause of bottlenecks – the costliness of later-stage testing. This has the implication that higher testing costs will reduce the number of combinations that advance to testing. However, another important source of bottlenecks is the poor efficacy of the various screens that make up the pipeline between the initial prediction stage and final testing stage. In the context of an option to abandon combinations following a negative screening result, we show that less effective screens also reduce the number of combinations that enter the pipeline. We discuss how such bottlenecks could limit the benefit from improved AI-based prediction and how AI might itself be applied to later stages in the pipeline to help alleviate those bottlenecks.

More specifically, we extend the basic two-stage model of prediction and testing to a discovery pipeline that involves potentially multiple intermediate screening stages in addition to the prediction and final (determinative) testing stages.[25] As in Case 1 above, we assume that the innovator is seeking to find all combinations with a positive net value. Although we continue to assume that the final testing stage is determinative, we allow for imperfections in the intermediate screening stages in the form of the possibility of false negatives and false positives in the screens. For simplicity, we assume the false negative rates and the false positive rates are constant across each of the intermediate stages. We denote the false-negative rate as x and the false-positive rate as y; the corresponding true positive and true negative are $1 - x$ and $1 - y$, respectively.

To minimize notation we again assume, without loss of generality, that the gross payoff from a success, π, is equal to 1. Thus, the expected gross value of a combination at the completion of any given stage is just the estimated probability of success at that stage. We assume initially that all stages must be completed to launch an innovation on the market so that it is not possible to skip a stage. At the completion of the final (test) stage, the gross value is equal to 1 (a success) or 0 (a failure), with

[25] As applied to a specific candidate combination, our multi-stage discovery process is an example of what Roberts and Weitzman (1981) call a *Sequential Development Project*: costs are additive across stages; value is received only at the end of the project; and there is a possibility of abandoning the project at the end of each stage.

the estimated probability of a success going into the testing stage equal to the estimated probability of success at the completion of the last intermediate stage.

There are S stages, $s = 0, \ldots, S$, including the Stage 0 prediction stage, the final Stage S testing stage and $S-1$ intermediate stages. For any given intermediate stage, $s \in \{1, \ldots, S-1\}$, the probability of success is optimally calculated using Bayes rule given the prior probability inherited from the previous stage. We thus model the pipeline as a series of screens between the initial prediction stage and the final determinative test that lead to Bayesian updating of the prior probability of success. Applying Bayes rule, this updating process is given by:

$$p_r^s|positive\ screen = \frac{(1-x)p_r^{s-1}}{(1-x)p_r^{s-1} + y\left(1-p_r^{s-1}\right)}. \tag{21}$$

$$p_r^s|negative\ screen = \frac{xp_r^{s-1}}{xp_r^{s-1} + (1-y)\left(1-p_r^{s-1}\right)}, \tag{22}$$

where the denominators in Eqs. 21 and 22 are the probability of a positive screen and a negative screen, respectively.

As a combination advances along the discovery pipeline, it is assumed to follow a discrete-stage Markov process as illustrated in Fig. 6. The assumption of

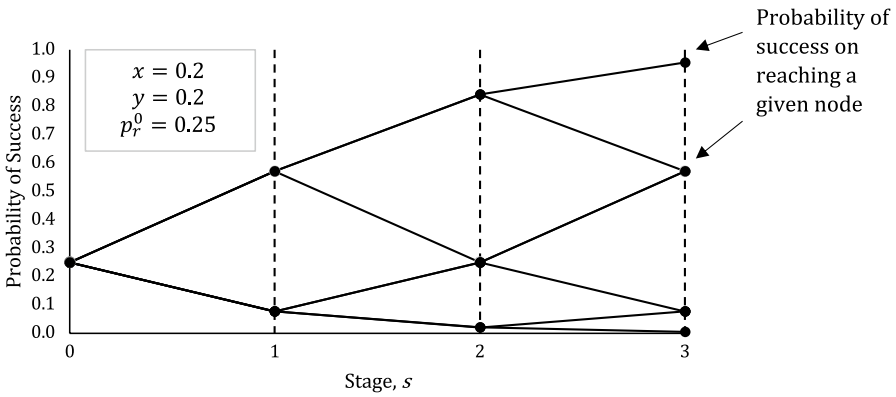

Fig. 6 Evolution of the probability of success through a multi-stage discovery pipeline. Notes: The figure shows an example of a five-stage discovery pipeline (where the final determinative testing stage is not shown). In addition to the initial Stage 0 prediction and final Stage 4 testing stages, there are three intermediate stages. The assumption of common values of both x (false negative rate) and y (false positive rate) for each of the screening stages produces the binomial lattice (or recombinant) structure of the possible evolutions of the probability of success of a given combination as it advances along the pipeline. The initial node gives the probability of success following the initial prediction stage ($p_r^0 = 0.25$ in the example above). At each intermediate stage, the node gives the probability of success given the outcome of the screen. The slopes of the lines on the graph are determined by ex ante probability of success or failures going into a given screening stage with depends on the relevant prior probability of success and the values of x and y. For a given stage,s, the number of possible values for the probability of success rises linearly with the number of the stage according to $s+1$. Viewed from Stage 0, the rational expectation of the probability of success at any subsequent stage is p_r^o

constant values of x and y at each stage means that we obtain the binomial lattice (or recombinant) structure shown in the figure, where the branches recombine to limit the number of possible values for the probability of success at each stage in the process. For example, the lattice structure implies that at Stage 3 the expected probability of success is the same for a given Stage 0 probability where there is a scenario of two positive screens followed by a negative screen, a scenario of a negative screen followed by two positive screens, or a scenario of a positive screen followed by a negative screen followed by another positive screen. Conveniently, the lattice structure implies that number of values that this probability can take at Stage s is $s+1$, so that the number of possible values rises only linearly with the number of the stage.[26]

For a given p_r^0, the probability of a reaching a given node at Stage s for a given number of positive and negative screens is:

$$q_r^s(h) = (1-x)^h x^{s-h} p_r^0 + y^h (1-y)^{s-h}(1-p_r^0). \tag{23}$$

In the case where any negative screen leads the combination to be abandoned, we obtain a simple expression for the probability that the combination survives s stages:

$$q_r^s(s) = (1-x)^s p_r^0 + y^s (1-p_r^0). \tag{24}$$

Figure 7 shows how the probability of surviving for a combination falls through multiple intermediate stages (assuming a combination is always abandoned following a negative screen). Thus, in addition to showing how the probability of success evolves as a combination travels through the pipeline, the model also allows us to make predictions about the survival rates along the pipeline.

Note that the probabilities for possible nodes that can be reached at a given stage s sum to 1:

$$\sum_{h=0}^{s} \binom{s}{h} [(1-x)^h x^{s-h} p_r^s + y^h (1-y)^{s-h}(1-p_r^0)] = 1, \tag{25}$$

where the number of paths to a given node is $\binom{s}{h}$. For example, at Stage 3, there are three paths to the nodes involving a total of two successes and one failure: $\binom{s}{n} = \binom{3}{2} = \frac{3!}{(3-2)!2!} = 3$. At Stage s, the probability of success given h positive screens and $s-h$ negative screens (i.e., the probability of success at the given node) is:

$$p_r^s | h positive\ screens = \frac{(1-x)^h x^{s-h} p_r^0}{(1-x)^h x^{s-h} p_r^0 + y^h (1-y)^{s-h}(1-p_r^0)}. \tag{26}$$

[26] With stage-specific values of x and y, the number of possible values for the probability of success would rise exponentially with the number of stages so that the number of possible values at stage s would be 2^s. Therefore, the lattice/recombinant structure dramatically simplifies the computational burden when there are multiple intermediate stages.

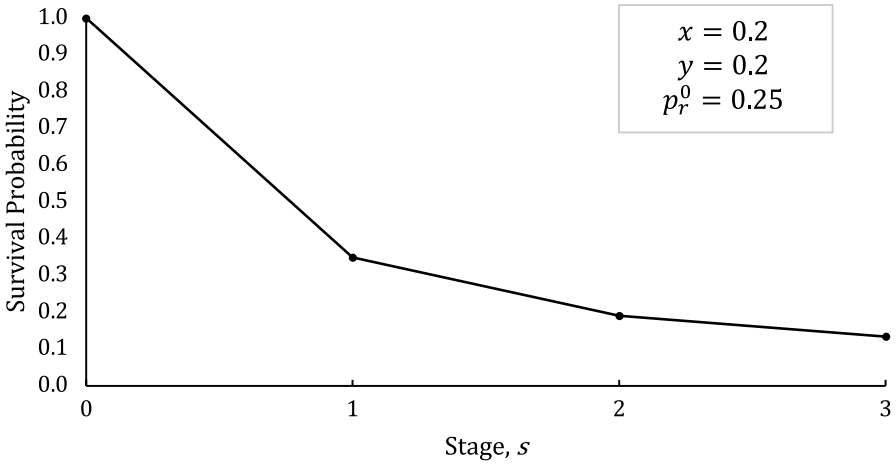

Fig. 7 Probability of survival following completion of stage, s Example of the case where the combination is always abandoned following a negative screen. The figure shows that survival probability through the pipeline under the assumption that the combination will be abandoned in the event of a negative screen. It thus gives the cumulative probability of s successful screens

we can verify that when viewed from the end of the Stage 0 (i.e., the prediction stage) the expected probability of success at Stage s remains equal to p_r^0:

$$\begin{aligned} p_r^s &= \sum_{h=0}^{s} \binom{s}{k} \left[(1-x)^h x^{s-h} p_r^0 + y^h (1-y)^{s-h}(1-p_r^0)\right] \frac{(1-x)^h x^{s-h} p_r^0}{(1-x)^h x^{s-h} p_r^0 + y^h (1-y)^{s-h}(1-p_r^0)} \\ &= \sum_{h=0}^{s} \binom{s}{k} (1-x)^h x^{s-h} p_r^0 = p_r^0. \end{aligned} \tag{27}$$

A three-stage "predict-screen-test" example

We now examine the implications of the existence of the option to abandon a combination upon the realization of a negative screen. We illustrate the implications for the number of combinations that initially enter the discovery pipeline with a three-stage example – that is a pipeline with an initial prediction stage (Stage 0) a single intermediate screening stage (Stage 1) and the final determinative testing stage (Stage 2). The cost of the intermediate screen is c_1 and the cost of final test is c_2. We assume that $c_1 + c_2 \leq 1$ and that both c_1 and c_2 are strictly positive. Upon receiving a negative result on the intermediate screen, the option to abandon will be exercised if:

$$\begin{aligned} &\frac{x p_r^0}{x p_r^0 + (1-y)(1-p_r^0)} < c_2. \\ &\Rightarrow p_r^0 < \frac{(1-y)c_2}{x + (1-x-y)c_2} \end{aligned} \tag{28}$$

Given the availability of the option to abandon, we now move attention back to the end of Stage 0. We assume that the innovator is operating in the range where the combination will be abandoned on the realization of a negative screen in Stage 1. We can therefore determine the cut-off Stage 0 probability below which combinations will not advance to intermediate screening, with a higher cut-off probability implying that fewer combinations advance given the ranking function. Equating expected marginal value with expected marginal cost allows us to identify the cut-off probability for advancing a combination to the intermediate stage. Moreover, given the logistic ranking function, the probability of success declines monotonically with the rank of the combination, so identifying the cut-off probability is identical to identifying the number of combinations to advance in the pipeline.

The equation of expected marginal value and expected marginal cost yields:

$$\begin{aligned}&\left[(1-x)p^0_{r^*}+y\left(1-p^0_{r^*}\right)\right]\left[\frac{(1-x)p^0_{r^*}}{(1-x)p^0_{r^*}+y\left(1-p^0_{r^*}\right)}\right]=c_1+\left[(1-x)p^0_{r^*}+y\left(1-p^0_{r^*}\right)\right]c_2,\\ &\Rightarrow p^0_{r^*}=\frac{c_1+yc_2}{(1-x)\left(1-c_2\right)+yc_2}.\end{aligned} \tag{29}$$

Note that for this cut-off probability to be strictly less than 1 we require that $c_1 < (1-x)\left(1-c_2\right)$. Moreover, for the denominator to be positive (and thus that the cut-off probability is greater than zero), we require that $(1-x)\left(1-c_2\right)+yc_2 > 0$. Finally, note that for the probability of a successful screen to increase with the prior probability of success we require that $1-x-y > 0$.

The innovator will choose to advance all combinations with a positive expected net value recognizing that Stage 2 cost will not be incurred (i.e., the option to abandon will be exercised) in the event of a negative screen. Figure 8 examines this decision by relating the expected net value of the r^{th} ranked combination, v^e_r, to p^0_r. The line with the intercept $-(c_1+c_2)$ shows how the expected net value changes with p^0_r in the absence of an option to abandon or where that option is not exercised. The line with the intercept $-(c_1+yc_2)$ shows how expected net value evolves with p^0_r assuming the option to abandon is always exercised when there is a negative result on the Stage 1 intermediate screen. The bold line shows the evolution of expected net value given the optimal advancement of combinations from Stage 0 as given by Eq. (21) and the optimal exercise of the option to abandon after the screening result given by Eq. (20). For low values of p^0_r the combination will not advance to the screening stage and the expected net value is zero. For intermediate values of p^0_r the combination will advance to screening but will be abandoned in the event of a negative screen, saving on Stage 2 testing costs. For high enough values of p^0_r the combination will not be abandoned even in the event of a negative screen. However, we assume here that the screen must take place due to it being an indispensable part of the final testing stage so that it needs to be conducted regardless of whether the combination advances to the final stage. In the next section, we examine the implications of being able to skip the screening stage altogether where Stage 0 predictions are sufficiently high and go straight to determinative testing.

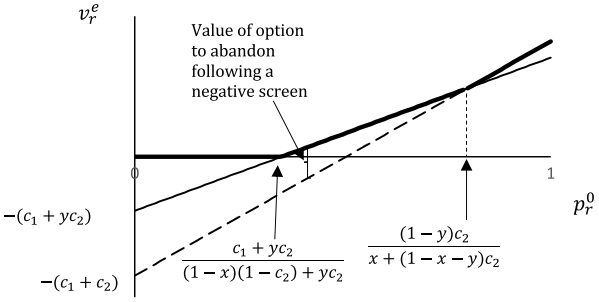

Fig. 8 Expected net value of a combination with option to abandon and screen must be completed to advance a combination to testing. Notes: The figure shows how the expected net value of a combination varies with the post-Stage 0 probability of success for a three-stage discovery pipeline with a single intermediate screening stage. The line with intercept $-(c_1 + c_2)$ shows how the expected net value would evolve if both the intermediate screening and final testing stage have to be completed. The line with intercept $-(c_1 + yc_2)$ shows how the expected net value would evolve if a combination is always abandoned if there is a negative screen. The bold line reflects optimal behaviour on the part of the innovator where: (i) negative expected net value combinations will not advance from Stage 0 even after taking into account the option to abandon after the screening stage; and (ii) for high enough values of the Stage 0 probability of success the combinations will abandoned even after receiving a negative result at the intermediate screening stage. The first upward pointing arrow indicates the cut-off value for the Stage 0 probability of success for advancement to the intermediate state. The second upward pointing arrow indicates the Stage 0 probability of success below which the option to abandon would be exercised on receiving a negative screen. The gap between the dashed line and non-bold solid line gives the value of the option to abandon for combinations where the option to abandon would be exercised upon receipt of a negative screen

We finally examine how a change in either the cost of the screening or testing stages, or in the accuracy of the screen (which is affected by both x and y), impacts the number of combinations that advance from Stage 0. Starting with a change in either of the costs, we find:

$$\frac{\partial p_{r^*}^0}{\partial c_1} = \frac{1}{(1-x)\left(1-c_2\right) + yc_2} > 0. \tag{30}$$

$$\frac{\partial p_{r^*}^0}{\partial c_2} = \frac{y[(1-x)\left(1-c_2\right) + yc_2] + (1-x-y)[c_1 + yc_2]}{[(1-x)\left(1-c_2\right) + yc_2]^2} > 0 \tag{31}$$

The accuracy of the test will decrease with either a rise in the false-negative rate (i.e., an increase in x) or a rise in the false-positive rate (i.e., an increase in y) on the screen. In both cases, the effect is to raise the cut-off probability and thus decrease the number of combinations advancing in the pipeline:

$$\frac{\partial p_{r^*}^0}{\partial x} = \frac{(1-c_2)[c_1 + yc_2]}{[(1-x)\left(1-c_2\right) + yc_2]^2} > 0. \tag{32}$$

$$\frac{\partial p^0_{r^*}}{\partial y} = \frac{c_2[(1-x)(1-c_2)-c_1]}{[(1-x)(1-c_2)+yc_2]^2} > 0. \tag{33}$$

Therefore, in addition to the bottlenecks that result from the costliness of later stages in the discovery pipeline, this simple model of the pipeline shows how an additional source of bottlenecks can result from poorly performing screens as captured by high false negative rates and/or high false positive rates. As an increase in either x or y will effectively decrease the uncertainty associated with a given screen, it decreases the value of the option to abandon. A bottleneck is then a reflection of *low* uncertainty relating to the screen given the poor discriminating power of that screen. As is familiar with options pricing in the financial and real options contexts, this reduction in uncertainty as a result of low efficacy screens will reduce the option value component of the total net expected value.

The effect of an increase in bottlenecks (i.e., some combination of higher values of c_1 and/or c_2 and higher values of x and/or y) on the number of combinations advancing from Stage 0 is shown in Fig. 9). A worsening of any of the determinants of the bottlenecks will shift the horizontal line upwards, leading to an increase in the cut-off Stage 0 probability of success and thus a reduction in the number of combinations that enter the pipeline (i.e., a fall in r^*).

The logic of the three-stage example can be extended to a discovery process with $S-1$ intermediate stages. Entering the penultimate stage (i.e., the last intermediate stage) the analysis is identical at any given node to the three-stage example given above. Any positive option value for this sub-problem will result in positive option value for the decision problem overall. A similar argument applies to any other node along the decision tree, where the decision maker assumes they will act optimally in terms of any decision to abandon at any subsequent node. Therefore, if the option to abandon has value at any node in the decision tree, then the existence of the option has value for the decision process overall.

The challenge faced by innovators is sometimes thought of in terms of the high failure rate along the discovery pipeline. However, this analysis points to the importance of "failing" early before significant costs are incurred in pursuing ultimately unsuccessful innovations. Being able to identify poor combinations early increases

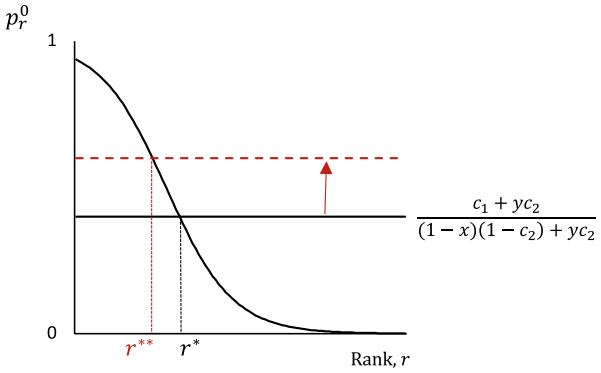

Fig 9 Impact of an increase in bottlenecks on the number of combinations advancing to the screening stage in the discovery pipeline

the incentive to allow combinations to advance from the AI-assisted prediction stage. The goal might be thought of as: "fail early, fail often" (Babineaux and Krumboltz 2013). We can think of an important adverse effect from the bottleneck as being combinations that stay too long in the pipeline but eventually fail, decreasing the incentive to enter combinations into the post-prediction pipeline to begin with.

Appendix 4. Interaction of bottlenecks with an AI-based improvement in prediction

Examples of negative interactions

Our model of the discovery process has shown: (i) that an AI-based improvement in first-stage prediction leads to an increase in the expected net value of innovation; and (ii) for any given combination that enters the discovery pipe, that bottlenecks in the pipeline due to costly screening and testing and/or the poor efficacy of screens reduces the expected net value of that combination. A remaining question is how the existence of bottlenecks interacts with the AI-based improvement in prediction. While the interaction is generally quite complex, depending, *inter alia*, on the nature of the innovation task, we illustrate the possibility of a negative interaction for the independent innovations case where the innovator is seeking all possible positive expected value innovations. We examine both a two-stage and a three-stage example.

A two-stage (predict-test) example

We begin with a simple two-stage example where the only source of bottleneck is the cost of Stage 1 testing. Figure 10 shows the expected net gain from the adoption of the AI-prediction technology. Ignoring the discrete nature of the problem,

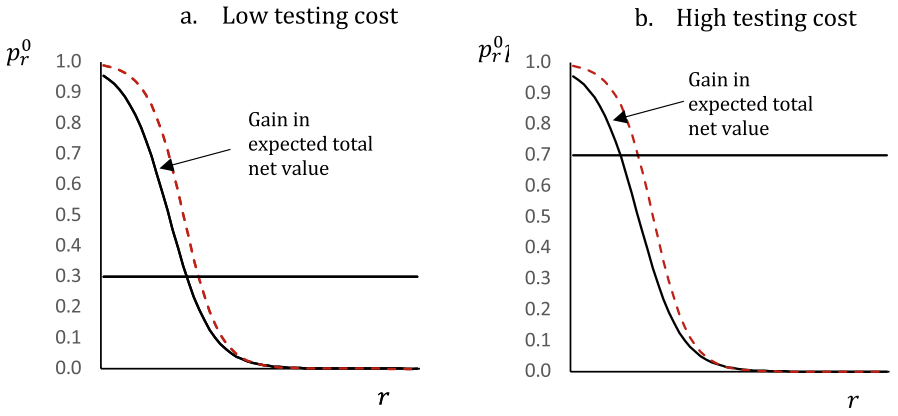

Fig. 10 Expected gain in total net value from access to improved AI-based prediction technology in the two-stage model (Case 1)

the area between the ranking functions up to the number of combinations that advance for testing is the total expected gain from the access to AI. The left panel shows the case of a low cost of second-stage testing (i.e., less severe bottleneck) and the right panel shows the case of high cost second-stage testing (i.e., more severe bottleneck). The increase in expected total net value from access to the improved prediction technology is negatively affected where the extent of the bottleneck is greater. In addition to reducing the positive impact of the improved prediction technology once adopted, the existence of the more severe bottleneck could affect the decision to invest in the AI-based prediction technology to begin with. Such negative interaction could therefore lead to a limited impact on the innovation process of the availability of the improved prediction technology, even though in expectation the impact remains positive.

A three-stage (predict-screen-test) example

We next consider the interaction between an improved prediction model and the wider set of potential bottlenecks in the three-stage model. We make the additional simplifying assumption that the false-positive rate, y, is zero, leaving three sources of potential bottlenecks – high values of c_1, c_2 or x. Using Eq. (26), we can write the expected net value of a combination at a given rank, r, as:

$$\begin{aligned} v_r^e &= (1-x)p_r^0 - c_1 - \left[(1-x)p_r^0\right]c_2 \\ &= \left[(1-x)\left(1-c_2\right)\right]p_r^0 - c_1. \end{aligned} \tag{34}$$

we can therefore visualize the expected total net value, V^e, as the area between the two curves shown in Fig. 11a and b up to the optimal number of combinations to be sent for testing. Figure 10a shows the impact of an improved prediction model on V^e for both high and low values of c_1. The increase in V^e is higher where c_1 is low rather than where it is high, indicating again a negative interaction between the improvement in the prediction model and the extent of the bottleneck. Figure 11b compares the impact of an improved prediction model in the case where either c_2 or x is low compared to where either c_2 or x is high. The size of both c_2 or x will affect how much the downward-sloping curve will shift following an improvement in the prediction model. As with the first example, we see a negative interaction between the improved model and the size of the bottleneck. More substantial bottlenecks will again make it less likely that an investment in the improved prediction technology will be undertaken.

We hasten to add that both these examples assume independent innovations (Case 1). As we have seen in our analysis of Case 1 in Section 4, an improvement in the prediction technology leads to *more* combinations advancing in the pipeline. This explains why higher costs and less effective screens tend to blunt the positive impact of the improved technology. However, as was evident in Cases 2 and 3, it is possible that the improved technology leads to more targeted search with fewer combinations advancing. With less combinations advancing the existence of bottlenecks becomes less of a concern (e.g., fewer costly tests need to be

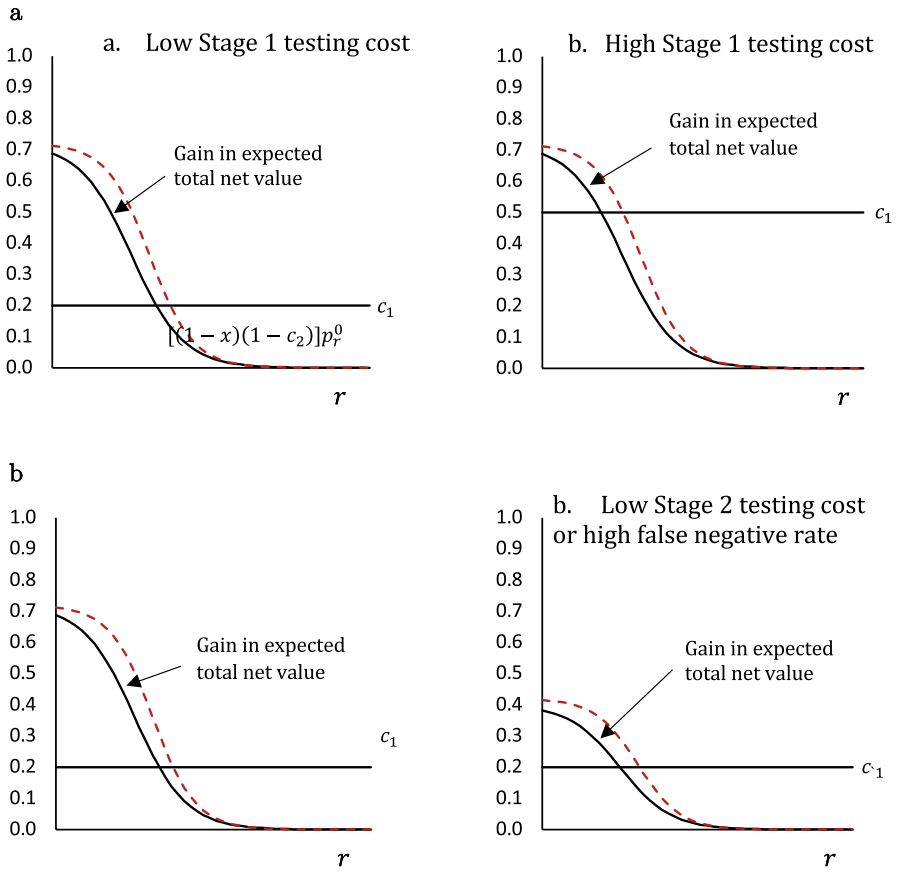

Fig. 11 **a** Expected gain in total net value from access to improved AI-based prediction technology in the three-stage model (low versus high values of c_1)**. b** expected gain in total net value from access to improved AI-based prediction technology in the three-stage model (low versus high values of c_2 or x)

conducted), so that the improvement in the prediction technology and the extent of the bottlenecks interact positively rather than negatively. Nevertheless, putting aside the interactions, it remains true that if AI can be applied to later stages of the pipeline to reduce bottlenecks, there will be a *direct* gain in expected value (e.g., lower costs for any advancing combination). We finally briefly discuss how AI could be applied beyond the prediction stage and consider both the possible direct and indirect benefits for innovation.

Could AI reduce bottlenecks over time?

As they relate to the innovation process, one characterization of recent advances in AI-based prediction is as a new general-purpose technology (GPT) for invention (Agrawal et al. 2018; Cockburn et al. 2019). As has been identified in other contexts, new GPTs may only have their full effect after a significant elapse of time due to the need for

complementary upstream and downstream investments. In a classic paper, David (1990) has explored the lagged productivity effects of both the invention of electricity and the computer, and similar lagged effects might be operating for AI as a technology for discovery.[27] Brynjolfsson et al. (2017) explore four potential causes of the general (not discovery-specific) productivity paradox in AI – false hopes, mismeasurement, redistribution, and implementation lags – and conclude that precisely this issue, implementation lags due to under developed complements, is the most likely culprit.

In the example of drug discovery, the benefit of AI in generating and ranking a large number of leads may be compromised, say, by a bottleneck following lead identification stage. It is worth considering a David-type solution that might evolve over time. As AI-assisted prediction develops, it might also be applied directly in other stages, including substituting for human judgment in the lead optimization (see Agrawal et al. 2019b on the roles of prediction and judgement in a multi-stage decision process).[28] One possibility is that AI is used to predict the outcome of later screens – say the toxicity of a drug. This could lead to the early abandonment of unpromising combinations before significant costs are incurred.

Although it could be some distance in the future, with well-enough performing AI it might be possible to eliminate certain screening stages altogether and replace them with AI-based predictions, eliminating, say, the need for animal-based safety screens to allow a candidate drug to advance to clinical trials.

In the context of our three-stage example of predict-screen-test, we have already seen that if the Stage 0 predicted probability of success is high enough then the innovator knows that combination will not be abandoned even with a negative screen. We previously assumed that the costly screen had to be conducted (possibly for regulatory reasons). However, in the case where the screen is redundant, we can imagine a situation where the innovator is allowed to skip the screening stage. It follows if p_r^0 is high enough we return essentially to the two-stage case of predict and test.

Figure 12) amends Fig. 8) to allow for this possibility. If the Stage 0 predicted probability of success is high enough, we see a discontinuous jump in the bold line that links that probability to the expected net value of the combination, v_r^e. Therefore, if the use of AI-based prediction is successful enough in terms of generating high probability of success candidate combinations, it may directly obviate the requirement for costly (and time consuming) downstream parts of the discovery pipeline.

[27] Drawing on historical analogies of delayed productivity effects, David cautioned against undue pessimism due to the apparent limited impact of computers on productivity, notwithstanding their growing prevalence: "Closer study of some economic history of technology, and familiarity with the story of the dynamo revolution in particular, should help us avoid the pitfall of undue sanguinity and the pitfall of unrealistic impatience into which current discussions of the productivity paradox seem to plunge all too frequently," (p. 359–360).

[28] Much of the recent interest in machine learning in the econometrics and policy evaluation literature has focused on the value of machine learning in causal inference and policy evaluation (see Athey 2019, for a recent survey). Machine learning is being applied to help control for confounding variables to estimate average treatment effects and also to estimate heterogeneous treatment effects (Athey 2019; Athey and Imbens 2019). Such tools may be especially relevant in later stages of a multi-stage discovery process as complements, say, to random and natural experiments. Thus, machine learning may itself help relieve the bottleneck problem over time.

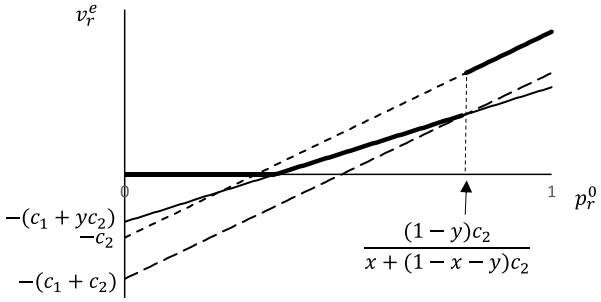

Fig. 12 Expected net value of a combination with option to abandon and a redundant screen can be skipped**.** Notes: This figure modifies Fig.8 to allow for the screening stage to be skipped when even a negative result on the screen does not lead to the exercise of the option to abandon the combination. Such an outcome will only result where the Stage 0 probability of success is sufficiently high: $p_r^0 \geq (1-y)c_2/[x+(1-x-y)c_2]$. There is then a discontinuous jump in the bold line that relates a given Stage 0 probability of success for a combination to the expected net value of that combination. If an AI-based improvement in prediction results in such high probabilities, it would have an additional effect of altering the subsequent downstream discovery pipeline

One additional active area of interest in terms of complementary investments to improved AI-based prediction technologies is the development of "autonomous discovery" systems (Aspuru-Guzik and Persson 2018). Such systems are particularly relevant where it is possible to have rapid feedback in terms of success data on combinations that advance along the pipeline, where those data can be used to improve the prediction model as part of an active learning strategy. For example, machine-learning-based predictions determine which candidates are tested using robotic high throughput screening (HTP) methods.[29]

Lamenting the slow speed and high cost of the development and deployment of advanced materials using the traditional approach – new materials typically "reach the market after 10–20 years of basic and applied research" – Tabor et al. (2018, p.5) outline what they see as required for an autonomous (closed-loop) innovation process:

> To fully exploit the advances in autonomous robotics, machine learning, high-throughput virtual screening, combinatorial methods and in situ or in operando characterization, we must close the loop in the research process. This means that humans must partner with autonomous research robots to design experimental campaigns and that the research robots perform experiments, analyze the results, update our understanding and then use AI and machine learning to design new experiments optimized to the research goals, thus completing one experimental loop.

[29] An early example of such an autonomous system is the Robot Scientist (Sparkes et al. 2010). The first prototype Robot Scientist, Adam, generated hypotheses and carried out experiments related to the functional genomics of a yeast. While it is unlikely that truly closed-loop systems will lead to a mass replacement of scientists, the potential exists for the greater use of AI and automation to ease bottlenecks across the discovery pipeline.

Current processes are affected by the bottleneck problem, which Tabor et al. (2018, p.16) suggest is in the "experimental synthesis, characterization and testing of theoretically proposed materials," but that if an autonomous approach could be implemented "the bottleneck will move to AI."

Acknowledgments We thank participants in the NBER conference on the Economics of Artificial Intelligence held in Toronto in September 2019, and the Brooking Center on Regulation and Markets Artificial Intelligence Authors Conference held online in May 2021, for their feedback. In particular, we thank our discussants Timothy Bresnahan (NBER) and David Audretsch (Brookings) for their valuable suggestions for improving the paper. We also thank the editors and two anonymous referees for their helpful advice. Of course, responsibility for all remaining deficiencies is our own. Finally, we thank the Center on Regulation and Markets at Brookings and Science Foundation Ireland (Grant Number 17/SPR/5329) for financial support.

Funding The authors received funding from the Center on Regulation and Markets at the Brookings Institute and McHale received financial support from the Science Foundation Ireland (Grant Number 17/SPR/5329).

Data availability We do not analyze or generate any datasets because our work proceeds within a theoretical and mathematical approach.

Declarations

Conflict of interest Agrawal: https://agrawal.ca/disclosure
McHale/Oettl: None
This research did not involve human subjects.

References

Acemoglu D, Autor D (2011): Skills, tasks and technologies: Implications for employment and earnings, In Handbook of labor economics, Elsevier, vol. 4, 1043–1171

Acemoglu D, Restrepo P (2018) The race between man and machine: Implications of technology for growth, factor shares, and employment. Am Econ Rev 108:1488–1542

Acemoglu D, Restrepo P (2019a) Artificial intelligence, automation, and work. In: Agrawal AK, Gans JS, Goldfarb A (eds) The economics of artificial intelligence: an agenda. University of Chicago Press, Chicago

Acemoglu D, Restrepo P (2019b) Automation and new tasks: How technology displaces and reinstates labor. J Econ Perspect 33:3–30

Aghion P, Howitt P (1992) A model of growth through creative destruction. Econometrica 60(2):323–351

Aghion P, Jones B, Jones C (2019) Artificial intelligence and economic growth. In: Agrawal AK, Gans JS, Goldfarb A (eds) The economics of artificial intelligence: An agenda. University of Chicago Press, Chicago

Agrawal AK, Gans JS, Goldfarb A (2018) Prediction Machines: The simple economics of artificial intelligence. Harvard Business Review Press, Boston

Agrawal A, McHale J, Oettl A (2019a) Finding needles in haystacks: artificial intelligence and recombinant growth. In: Agrawal AK, Gans JS, Goldfarb A (eds) The economics of artificial intelligence: An agenda. University of Chicago Press, Chicago

Agrawal AK, Gans JS, Goldfarb A (2019b) Exploring the impact of artificial intelligence: Prediction versus judgment. In: Agrawal AK, Gans JS, Goldfarb A (eds) The economics of artificial intelligence: An agenda. University of Chicago Press, Chicago

Agrawal A, McHale J, Oettl A (2022): "Superhuman science: How artificial intelligence may impact innovation," Centre on Regulation and Markets at Brookings, Working Paper, Washington D.C

Agrawal A, McHale J, Oettl A (2023) Artificial intelligence and scientific discovery: A model of prioritized search, NBER Working Paper 31558 (August), Cambridge

Angermueller C, Pärnamaa T, Parts L, Stegle O (2016) Deep learning for computational biology. Mol Syst Biol 12:878
Arrow KJ (1962) The economic implications of learning by doing. Rev Econ Stud 29:155–173
Arthur BW (2009) The Nature of Technology: What it is and How it Evolves. Penguin Books, London
Aspuru-Guzik A. Persson K (2018) Materials acceleration platform: Accelerating advanced energy materials discovery by integrating high-throughput methods and artificial intelligence. Tech. rep., Canadian Institute for Advanced Research
Athey S (2017) Beyond prediction: Using big data for policy problems. Science 355:483–485
Athey S (2019) The impact of machine learning on economics. In: Agrawal AK, Gans JK, Goldfarb A (eds) The Economics of Artificial Intelligence: An Agenda. University of Chicago Press, Chicago
Athey S, Imbens G (2019) Machine learning methods economists should know about, arXiv preprint arXiv:1903.10075
Autor DH (2015) Why are there still so many jobs? The history and future of workplace automation. J Econ Perspect 29:3–30
Babineaux R, Krumboltz J (2013). *Fail fast, fail often: How losing can help you win*. TarcherPerigee.
Bender A, Cortés-Ciriano I (2020) "Artificial intelligence in drug discovery: what is realistic, what are the illusions? Part 1: Ways to make impact and why we are not there yet. Drug Discov Today 26(2):511–524
Born M, Oppenheimer R (1927) Zur quantentheorie der molekeln. Annalen der physik 389(20):457–484
Breiman L (2001) Statistical modeling: The two cultures. Statist Sci 16:199–231
Brown N, Ertl P, Lewis R, Luksch T, Reker D, Schneider N (2020) Artificial intelligence in chemistry and drug design. J Comput Aided Mol Des 34:709–715
Brynjolfsson E, Rock D, Syverson C (2017) Artificial intelligence and the modern productivity paradox: A clash of expectations and statistics, NBER Working Paper No. 24001
Butler KT, Davies DW, Cartwright H, Isayev O, Walsh A (2018) Machine learning for molecular and materials science. Nature 559:547
Chen H, Engkvist O, Wang Y, Olivecrona M, Blaschke T (2018) The rise of deep learning in drug discovery. Drug Discov Today 23(6):1241–1250
Cockburn I, Henderson R, Stern S (2019) The impact of artificial intelligence on innovation. In: Agrawal AK, Gans JS, Goldfarb A (eds) The economics of artificial intelligence: An agenda. University of Chicago Press, Chicago
David PA (1990) The dynamo and the computer: an historical perspective on the modern productivity paradox. Am Econ Rev 80:355–361
Dixit AK, Pindyck RS (1994) Investment under Uncertainty. Princeton University Press, Princeton
Dosi G (1982) Technological paradigms and technological trajectories: a suggested interpretation of the determinants and directions of technical change. Res Policy 11(3):147–162
Fleming L (2001) Recombinant uncertainty in technological search. Manag Sci 47:117–132
Fleming L, Sorenson O (2004) Science as a map in technological search. Strat Manag J 25:909–928
Gavetti G, Levinthal D (2000) Looking forward and looking backward: Cognitive and experiential search. Adm Sci Q 45:113–137
Goh GB, Hodas NO, Vishnu A (2017) Deep learning for computational chemistry. J Comput Chem 38:1291–1307
Gomes J, Ramsundar B, Feinberg EN, Pande VS (2017) Atomic convolutional networks for predicting protein-ligand binding affinity," arXiv preprint arXiv:1703.10603
Grossman G, Helpman E (1991) Quality ladders and product cycles. Q J Econ 106:557–586
Iansiti M, Lakhani KM, Mayer H, Herman K (2021) Moderna Harvard Business School Case # 9-621-032. Harvard Business School Publishing
Jones C (1995) R&D-based models of economic growth. J Polit Econ 103:759–784
Jones C (2005) Growth and ideas, In Handbook of economic growth, vol. 1, Elsevier, p 1063–1111
Jones C (2021) Recipes and economic growth: A combinatorial march down an exponential tail, NBER Working Paper 28340
Kauffman S (1993) The origins of order: Self-organization and selection in evolution. Oxford University Press, Oxford and New York
Kauffman S, Lobo J, Macready WG (2000) Optimal search on a technology landscape. J Econ Behav Organ 43:141–166

Keith JA, Vassilev-Galindo V, Cheng B, Chmiela S, Gastegger M, Müller KR, Tkatchenko A (2021) Combining machine learning and computational chemistry for predictive insights into chemical systems, arXiv: 2102.06321v1[physics.chem-ph]
Kortum (1997) Research, patenting, and technological change. Econometrica 65(6):1389–1419
LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521:436
Leung MK, Delong A, Alipanahi B, Frey BJ (2016) Machine learning in genomic medicine: a review of computational problems and data sets. Proc IEEE 104:176–197
Levinthal DA (1997) Adaptation on rugged landscapes. Manag Sci 43:934–950
Mamoshina P, Vieira A, Putin E, Zhavoronkov A (2016) Applications of deep learning in biomedicine. Mol Pharm 5:1445
Merk D, Friedich L, Grisoni F, Schneider G (2018) Do novo design of bioactive small molecules by artificial intelligence. Mol Inf 37(1):1700153
Mullainathan S, Spiess J (2017) Machine learning: an applied econometric approach. J Econ Perspect 31:87–106
Nature Communications (2020) Computation sparks chemical discovery, Nature Communications (Editorial), p 1-3
Nelson RR, Winter SG (1982) An evolutionary theory of economic change. Harvard University Press, Cambridge
Popper KR (1959) The logic of scientific discovery. Hutchinson, London
Pyzer-Knapp EO, Li K, Aspuru-Guzik A (2015) Learning from the Harvard clean energy project: The use of neural networks to accelerate materials discovery. Adv Funct Mater 25:6495–6502
Reker D, Schneider G (2014) "Active-Learning Strategies in Computer-Assisted Drug Discovery. Drug Discovery Today 20(4):458–465
Rivkin JW (2000) Imitation of complex strategies. Manag Sci 46:824–844
Roberts K, Weitzman ML (1981) Funding criteria for research, development and exploration projects. Econometrica 49(5):1261–1288
Romer P (1990) Endogenous technical change. J Polit Econ 94:S71–S102
Romer P (1992) Two strategies for economic development: using and producing ideas. World Bank Econ Rev 6(suppl_1):63–91
Schneider G (2018) "Automating drug discovery. Nat Rev: Drug Discov 17:97–113
Schumpeter J A (1939) Business cycles, vol. 1, McGraw-Hill New York
Shwartz-Ziv R, Tishby N (2017) Opening the black box of deep neural networks via information, arXiv preprint arXiv:1703.00810
Sparkes A, Aubrey W, Byrne E, Clare A, Khan MN, Liakata M, Markham M, Rowland J, Soldatova LN, Whelan KE, Young M, King RD (2010) Towards robot scientists for autonomous scientific discovery. Autom Exp 2:1
Stigler GJ (1961) The economics of information. J Polit Econ 69(3):213–225
Szabo A, Ostlund NS (1996) Modern quantum chemistry : Introduction to advanced electronic structure theory. Dover Publishing, Mineola
Tabor DP, Roch LM, Saikin SK, Kreisbeck C, Sheberla D, Montoya JH, Dwaraknath S, Aykol M, Ortiz C, Tribukait H et al (2018) Accelerating the discovery of materials for clean energy in the era of smart automation. Nat. Rev. Mater. 3:5–20
Taddy M (2019) The technological elements of artificial intelligence. In: Agrawal AK, Gans JS, Goldfarb A (eds) The economics of artificial intelligence: An agenda. University of Chicago Press, Chicago
Tang BZ, Pan K Yin, Khateeb A (2019) Recent advances of deep learning in bioinformatics and computational biology. Front Genet 10(2019):214
Tjur T (2009) Coefficients of determination in logistic regression models—A new proposal: The coefficient of discrimination. Am Stat 63:366–372
Trammell P, Korinek A (2023) Economic growth under transformative AI. NBER Working Paper 31815
Usher AP (1929) A history of mechanical inventions, revised. McGraw-Hill, New York
Vamathevan J, Clark D, Czodrowski P, Dunham I, Ferran E, Lee G, Li B, Madabhushi A, Shah P, Spitzer M, Zhao S (2019) Applications of machine learning in drug discovery and development. Nat Rev Drug Discov 18:463–477
Virshup AM, Contreras-García J, Wipf P, Yang W, Beratan DN (2013) Stochastic voyages into uncharted chemical space produce a representative library of all possible drug-like compounds. J Am Chem Soc 135:7296–7303
Wainberg M, Merico D, Delong A, Frey BJ (2018) Deep learning in biomedicine. Nat Biotechnol 36:829

Wallach I, Dzamba M, Heifets A (2015) AtomNet: A deep convolutional neural network for bioactivity prediction in structure-based drug discovery, arXiv preprint arXiv:1510.02855.
Wallach I, Heifets A (2018) Most ligand-based classification benchmarks reward memorization rather than generalization. J Chem Inf Model 58:916–932
Waltz E (2020) What AI Can – and can't – do in the race for a coronavirus vaccine. In: IEEE Spectrum (online: https://spectrum.ieee.org/what-ai-can-and-cant-do-in-the-race-for-a-coronavirus-vaccine). Accessed 9 Oct 2021
Weitzman ML (1979) Optimal search for the best alternative. Econometrica 47(3):641–654
Weitzman ML (1998) Recombinant growth. Q J Econ 113:331–360
Wright S (1932) The roles of mutation, inbreeding, crossbreeding and selection in evolution, In: Proceedings of the 6th International Congress of Genetics, vol. 1, p 356–366
Zou J, Huss M, Abid A, Mohammadi P, Torkamani A, Telenti A (2019) A primer on deep learning in genomics. Nat Gen 51:12–18