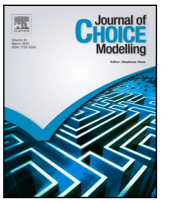
Journal of Choice Modelling 48 (2023) 100418

Contents lists available at ScienceDirect

Journal of Choice Modelling

journal homepage: www.elsevier.com/locate/jocm



How to ask twenty questions and win: Machine learning tools for assessing preferences from small samples of willingness-to-pay prices

Konstantina Sokratous, Anderson K. Fitch, Peter D. Kvam *

University of Florida, United States of America

ARTICLE INFO

Keywords:
Neural networks
Simulation
Cognitive modeling
Pricing
Risk preference

ABSTRACT

Subjective value has long been measured using binary choice experiments, yet responses like willingness-to-pay prices can be an effective and efficient way to assess individual differences risk preferences and value. Tony Marley's work illustrated that dynamic, stochastic models permit meaningful inferences about cognition from process-level data on paradigms beyond binary choice, yet many of these models remain difficult to use because their likelihoods must be approximated from simulation. In this paper, we develop and test an approach that uses deep neural networks to estimate the parameters of otherwise-intractable behavioral models. Once trained, these networks allow for accurate and instantaneous parameter estimation. We compare different network architectures and show that they accurately recover true risk preferences related to utility, response caution, anchoring, and non-decision processes. To illustrate the usefulness of the approach, it was then applied to estimate model parameters for a large, demographically representative sample of U.S. participants who completed a 20-question pricing task — an estimation task that is not feasible with previous methods. The results illustrate the utility of machine-learning approaches for fitting cognitive and economic models, providing efficient methods for quantifying meaningful differences in risk preferences from sparse data.

1. Introduction

For decades, the formal study of preference was framed around the lens of deterministic utilities and focused on algebraic, axiomatic models of choice (Savage, 1954). Despite the prevalence of choice as a way of measuring subjective value, this is not the only way to understand preferences and their representation. Deterministic theories often fail to adequately account for intra- and inter-individual variability in decision making (Marley and Regenwetter, 2016), and different methods of eliciting preference can even contradict the findings of choice experiments (Lichtenstein and Slovic, 1971; Slovic and Lichtenstein, 1983). Consequently, deterministic utility functions gave way to representations of value defined in probabilistic terms. These random utility models were pivotal because they allowed for modeling responses as a stochastic process, which in turn led to the development of new approaches to modeling subjective value that reconciled choice with other measurement processes like certainty equivalents and pricing schemes (Johnson and Busemeyer, 2005).

The pivot toward understanding value from multiple perspectives has led to a proliferation of models combining utility, risk, and value in choices, prices, and other methods of preference elicitation like demand tasks (Georgescu-Roegen, 1958; Koffarnus et al., 2015). However, the usability of various models is limited by our practical ability to fit and compare them. Probabilistic models

* Corresponding author.
E-mail address: pkvam@ufl.edu (P.D. Kvam).

https://doi.org/10.1016/j.jocm.2023.100418
Received 22 July 2022; Received in revised form 28 April 2023; Accepted 11 May 2023
Available online 19 May 2023

are accompanied by a host of practical problems stemming from the fact that many patterns of behavior can arise from stochastic processes. There are many models and even broad classes of theories that have never been explored simply because they do not have convenient probability density functions that specify the likelihood of different patterns of data for various combinations of their parameters. Fortunately, the advent of machine learning and "likelihood free" methods has made it possible to explore new theories and models that were previously impossible (Lueckmann et al., 2019; Gutmann and Corander, 2016; Fengler et al., 2021).

In this paper, we examine how these tools can be brought to bear on models that quantify risk preference from willingness-to-pay prices. These models are conceptually critical because they account for preference reversals as well as distributions and dynamics of pricing behavior (Kvam and Busemeyer, 2020). We show that applying deep learning to the fitting process of complex simulation-based models can reduce the required fitting time by several orders of magnitude, in our applications reducing the time from several months to several minutes. Furthermore, we show that the efficiency of deep neural networks for model fitting allows us to estimate parameters with a minimal amount of behavioral data — making it possible to understand latent processes related to risk aversion, utility, anchoring, and preference dynamics using pricing paradigms with a small number of trials. We conclude by showing that this approach allows for new model-based insights about risk preferences, applying it to fit pricing models to a large volume of participants to grant new insights into pricing behavior across different groups of people.

Throughout his research career, Tony Marley made strides to improve the effectiveness of our models of preference, in large part by extending and enriching random utility models. An impressive volume of Tony's work centered around the idea that internal representations of stimuli or value can be assigned to responses by comparing them to *anchors*, or exemplar values of stimuli that lie at along or at either end of the range of values a participant might encounter on a particular task (Marley and Cook, 1984; Lacouture and Marley, 1995, 2004; Brown et al., 2008). In some of his last work (Kvam et al., 2023), we showed that these anchor-based representations could be dynamically mapped onto both discrete and continuous scales. Although these types of models have traditionally been applied to perceptual decision making experiments, anchors and anchoring effects are directly relevant to value-based choice and pricing judgments (Kvam and Busemeyer, 2020; Tversky et al., 1990), connecting Tony's work on perceptual choice with his work on dynamic models of subjective value (Marley, 1989; Marley and Colonius, 1992; Hawkins et al., 2014). In making these connections, we can create a dynamic, stochastic theory of pricing that accounts for many of the phenomena of random utility models (Corbin and Marley, 1974) while also accounting for process data like response times (Luce, 1986). Specifically, the models we examine here focus on how the competing goals of high-probability and high-payoff prospects combine to determine subjective utility (Swait and Marley, 2013), but use pricing rather than choice as a measure of preference. These approaches resolve apparent preference reversals – where participants select Option A over Option B, but price Option B higher than Option A in willingness-to-pay judgments – in terms of a common currency of utility by attributing differences in preference to response processes rather than "true" differences in subjective value.

There are a number of reasons to examine pricing as a measure of subjective utility if a researcher is interested in assessing risk preferences, whether from a psychological or economic perspective. One is that pricing avoids relative comparisons between options that appear prevalent in binary choice (Busemeyer and Townsend, 1993; Scheibehenne et al., 2009; Gonzalez-Vallejo, 2002; Dai and Busemeyer, 2014), creating context effects based on the differences in payoffs and likelihoods between options. Naturally, there are still context effects from sequences of trials that create effects like contrast, range, and assimilation that are likely to affect pricing (Sherif et al., 1958; Brown et al., 2008) and cause anchoring effects that lead to preference reversals (Tversky et al., 1990). Fortunately, both choice and pricing appear to measure the same underlying utilities, or at least it is not apparent that they measure diverging preferences despite preference reversal phenomena (Johnson and Busemeyer, 2005; Kvam and Busemeyer, 2020). Therefore, pricing can be an effective route to understanding risk preferences as long as we account for the psychological and cognitive processes involved in responding alongside core economic concepts like utility.

Beyond the perspective it adds to complement binary choice, there are information theoretic reasons to favor pricing as a method of eliciting preferences as well. A binary decision provides 1 bit of information per choice problem (0 = choose safe option, 1 = choose risky option). In multi-alternative choice, we theoretically learn which option will be favored over all others in the choice set in a series of binary choices, yielding $n - 1$ bits of information for n alternatives. Ironically, this may be more information than a participant needs to collect to make their decision, which grows as a logistic rather than linear function (Hick, 1952). Different preference elicitation paradigms can further capitalize on this difference, yielding more and more information about a person's preferences from fast, simple responses. As Tony Marley's work suggested (Hawkins et al., 2014; Louviere et al., 2008), ranking and best-worst responses can provide substantially more information than binary choice alone – with $n!$ different rank orders, they give up to $\log_2(n!)$ bits of information per problem – but naturally these rankings take quite a long time for participants to complete. By contrast, pricing allows for a selection to be made from a wide range of options in matters of dollars and cents, which can theoretically provide an infinite amount of information by virtue of each response falling along a continuous scale. In practical terms, participants seem unlikely to be much more precise with their responses than 10–25 cents. For a range of $0–20, a precision of 25 cents provides us with ~ 6.32 bits of information, while a precision of 10 cents provides ~ 7.64 bits. Previous results suggest that people can obtain a precision of 10 cents within about 5 s (Kvam and Busemeyer, 2020), meaning that they can communicate at least 1 bit of information per second by responding with prices. Given the amount of time to complete a pricing problem is much less than 6 times the amount of time a participant spends on a binary choice – in fact, it is closer to $\sim 2\times$ the length of time (Kvam and Busemeyer, 2020) – pricing seems to be an efficient method of eliciting preferences among prospects.

In this paper, we explore how a pricing model based on a double-anchor theory of decision-making can be automated to make inferences about risk preferences from a small number of trials (20). This is made possible by leveraging the rich information provided by prices in combination with machine learning approaches to estimating pricing models, allowing us to maximize our use of response information to make parametric inferences about how people assign value. The procedure is accomplished by taking

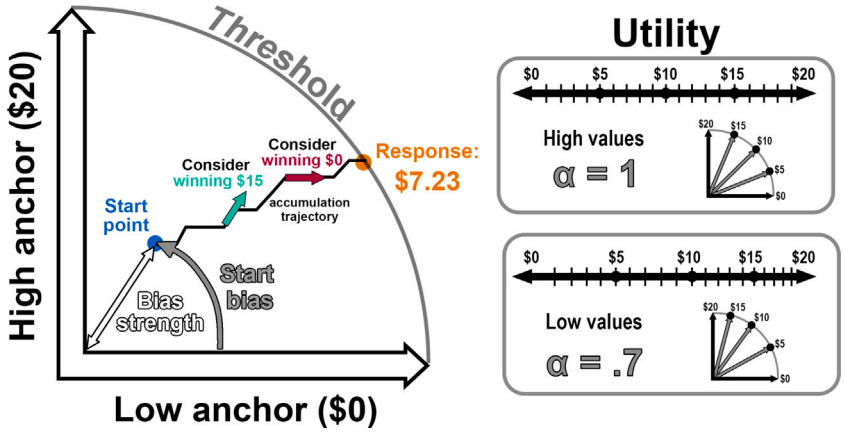


Fig. 1. Diagram of the price accumulation model and the effects of its parameters on its predictions. The utility parameter (right) determines the relative orientations of different dollar values around the quarter-circular scale (right), stretching or skewing the differences between low and high dollar amounts.

behavior on a set of pricing problems, using a neural network-based estimation approach (Radev et al., 2020b) to fit a pricing model, and then using the parameter estimates as individual differences to situate a participant in the broader context of their peers. In the next section, we begin by describing the cognitive/economic modeling approach we use to quantify behavior on pricing tasks (behavioral model), followed by a deep learning method (algorithmic model) to analyze the data on the task in terms of the parameters of the behavioral model. The ultimate aim of the paper is to present an alternative method that allows us to make rich inferences about behavior with a pricing model (Kvam and Busemeyer, 2020) and to show how this is enabled by the use of deep learning.

1.1. Behavioral model

To account for behavior on pricing tasks, we implement a dynamic model of pricing developed by Kvam and Busemeyer (2020), where we assume that prices people assign result from a process of accumulation driven by payoffs and their respective probabilities. Suppose a person is considering a simple lottery with probability p of winning $\$x$ and a probability $1-p$ of receiving \$0. The idea behind the model is that a person starts with an anchor at $\$x$ and then mentally simulates the possible outcomes they could get, i.e., $\$x$ and \$0. Over time, this provides support for different price responses — simulating a positive outcome pushes them toward higher prices near x, while simulating a negative outcome pushes them toward lower prices near 0. This stochastic simulation process continues to unfold until they reach sufficient support for a particular price response, which should fall between \$0 and $\$x$. The time it takes them to reach this response gives the corresponding response time.

A diagram of this model is shown in Fig. 1. In this approach, a price is generated as a participant thinks about the attributes of the risky prospect and assesses its value according to the mental simulation process. A participant making a price response first anchors on the outcome of the risky prospect as in anchoring & adjustment Goldstein and Einhorn, 1987, generating a "start point" for the price accumulation process. Formally, the location of a person's start point is specified by a proportion q specifying how much a person uses the anchor: the polar angle of their start point is $\phi = \frac{\pi \cdot \$_{max}}{2} \cdot q \cdot x$, where x is the payoff of the risky prospect and $\$_{max}$ is the upper anchor of the price scale. The strength of this anchor is given by another free parameter v specifying the distance (polar radius) of the starting point from zero. The x and y Cartesian coordinates of the starting point s are given as $s = [v\cos(\phi), v\sin(\phi)]$. This creates an initial state that is biased toward high prices for a risky/high-outcome gamble and biased toward low prices for a safe/low-outcome gamble, and which changes based on the perspective of the participant (e.g., higher for sellers and lower for buyers). Note that the start point carries all of the information relevant to anchoring, and is thus the primary mechanism for creating preference reversals when we compare pricing to choice (Tversky et al., 1990). This can be contrasted against more traditional accounts, where anchoring is built into the reference points and thus the utility function (Kahneman and Tversky, 1979).

From this starting point, the participant then considers the payoffs of the risky prospect and their relative likelihoods. Thinking of receiving the payoff drives them toward higher prices, resulting in their state s moving in the direction of the payout x, while thinking of failing to win the payoff drives them toward lower prices, resulting in their state s moving in the direction of \$0 (what they will receive before the delay elapses). The accumulation process unfolds as the participant steps in either direction at each moment in time, forming an accumulation trajectory like the one shown in the left panel of Fig. 1.

A participant continues this accumulation process until they obtain sufficient support for a price on the scale, where one price j on the continuum meets the condition $comp_j(s) \geq \theta$ — in plain terms, state moves far enough in the direction corresponding to price j on the quarter-circular scale. The parameter θ controls how much support a price requires before it is selected. When we offer participants a continuous scale for price, this creates a quarter-circular boundary to the price accumulation process as shown in Fig. 1 (Kvam, 2019). A higher value of θ will result in a longer accumulation process, while a lower value of θ will result in a

shorter accumulation process. Naturally, a longer accumulation process also dilutes the effect of the initial state, and thus results in weaker anchors. This means that low thresholds will result in price responses that are heavily influenced by the start biases, while high thresholds will result in responses that approach the expected utility of the gamble.

The final element of the pricing process involves taking a final state s – after hitting the threshold – and assigning it to a price response. In the simplest case, the radial scale corresponds linearly to price responses: 0 degrees is the lower anchor ($0), 90 degrees is the upper anchor or maximum dollar responses ($_{max}$; in this case, $20), and other responses are linearly spaced between them. Implementing such a model with a high threshold would yield price responses that are close to the expected value of a lottery. However, it is widely accepted that people assign a subjective utility to the outcomes of gambles (Savage, 1954) when considering their value. In the price accumulation model, much as in subjective utility-based models, this is described by a risk aversion/subjective utility parameter α. Its role in the price accumulation model is to distort the response space, as shown in the bottom-right of Fig. 1. The idea here is that the subjective difference between smaller price responses, such as $0 and $1, is larger than the subjective difference between large price responses, such as $18 and $19. If we are thinking in terms of anchors as Tony often did, we can think of there being a particularly strong anchor at $0 that makes subjective differences between values near $0 larger than those that are far from $0 (Kvam and Turner, 2021). Formally, the location of a particular price $\$j$ along the response scale, rather than being located at $\$j = \frac{\pi}{2} * \frac{j}{\$_{max}}$, is instead:

$$\$j = \frac{\pi}{2} * \left(\frac{j}{\$_{max}}\right)^{\alpha} \tag{1}$$

Smaller values of α therefore distort the scale so that smaller values are represented further apart and large values closer together. This view ties together the subjective utility differences with theories of perceptual differences, such that the perceived differences between small values is greater than the perceived differences between large values. The idea here is that both dollar values and stimulus magnitudes become more similar as they increase in value, owing to a common neural basis that produces Weber–Fechner relationships among perceived values (Dehaene, 2003).

In terms of estimation, the values of α interact with the stochastic accumulation process. Because the same amount of noise is present regardless of what the payoff is, there will be relatively greater noise among higher price responses with values of α less than one (the most common case). As a result, prices are more widely distributed among higher-value gambles than among low-value ones (Kvam and Busemeyer, 2020). These same low α that change the distributions of price responses will also result in choices favoring safe options over risky ones, indicating a connection between pricing and choice that concerns the *distribution* of prices rather than their rank order. Ultimately, the role of anchors in pricing – both at $0 and at the maximum value of each gamble – directly influences the response processes that differentiate pricing from other methods of eliciting preferences.

In particular, the role of anchors in the price accumulation model is the mechanism responsible for *preference reversals*, where participants price risky options higher but select safe options when a risky and safe option are presented side-by-side in binary choice (Lichtenstein and Slovic, 1971). By anchoring on the payoff as in this model, participants give it greater weight in determining their price responses. This is in line with the most common and widespread explanation for preference reversals — namely, that greater weight is given to payoffs in pricing problems than in choice (Tversky et al., 1990). The model helps disentangle this anchoring effect from the accumulation process, and connecting it to value-based decision models like decision field theory (Busemeyer and Townsend, 1993) shows that pricing and choice may in fact be driven by the same underlying subjective utilities (Johnson and Busemeyer, 2005; Kvam and Busemeyer, 2020) and values of the risk aversion parameter α.

The complete price accumulation process specified by this model will produce a joint distribution of prices and response times (Kvam and Busemeyer, 2020). Fig. 1 shows an example where a person is pricing a 40% chance of winning $15. A start bias of $\beta = .9$ yields a start point with a relatively high initially favored value, and a moderate value of start bias strength of $\nu = .4$ means that the start point is likely to have some influence over responses. As the decision-maker considers the possible outcomes, they step toward $15 on the scale (slightly above $\frac{3\pi}{8}$, due to a value of $\alpha = .8$) and toward $0. It accumulates until the decision-maker's state crosses the threshold at $\theta = 1$, and results in a response at $\frac{\pi}{5}$, a price corresponding approximately to $7.23. These same parameters might lead to price a safer gamble of ($7, 90%) at only $6.50. Yet in binary choice, we observe an expected utility of $15^{.9} \cdot .4 = 4.58$ utiles for ($15, 40%) and an expected utility of $7^{.9} \cdot .9 \sim 5.19$ utiles for ($7, 90%). As a result, we observe a preference reversal between pricing (where the risky option is favored) and binary choice (where the safe option is favored) produced by the anchor-based accumulation process of the price accumulation model.

Despite the importance of pricing models and the relevance of this particular anchor-based model to Tony Marley's work, there is a major problem standing in the way of their practical use. Specifically, the models do not have an analytic likelihood function, meaning that traditional likelihood-based approaches like maximum likelihood estimation or Bayesian/MCMC approaches cannot be applied. In the next section, we outline an approach for fitting this models using machine learning methods. A more complete and formal description of this approach – along with a comparison among different specific architectures, training sets, and other hyperpriors – is provided in the Algorithmic model section.

1.2. Estimation approach

One issue that becomes apparent when working out preference reversals is that models like expected utility make a single prediction for the value of each option and the probability of which option should be chosen. Even stochastic choice models like decision field theory (Busemeyer and Townsend, 1993) assign likelihoods to the choice probabilities of different options from a given set of parameters or predict analytic distributions of prices (Johnson and Busemeyer, 2005). However, this is not possible for

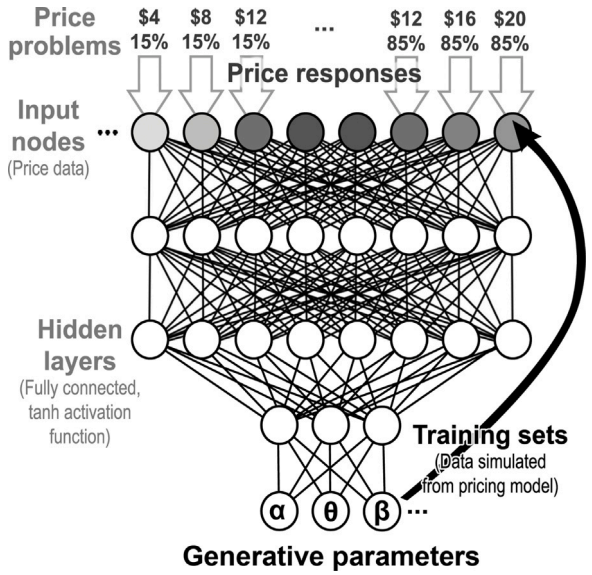


Fig. 2. Diagram of the structure of the neural network we trained to perform parameter estimation. Raw prices (top) were mapped onto generative model parameters (bottom) using multiple hidden layers with tanh activation functions.

the price accumulation model, which must be simulated in order to derive predictions for what prices will be observed. Instead, this likelihood has to be approximated using various methods such as approximate Bayesian computation (ABC) (Turner and Sederberg, 2012; Turner and Zandt, 2018) and specifically probability density approximation (Holmes, 2015; Turner and Sederberg, 2014). In these approaches, a large volume of simulated data is generated from the model for a given set of parameters, then compared to the observed data to evaluate the degree of match between the true and simulated data and compute a likelihood. This is repeated for each combination of parameters produced by a Markov chain Monte Carlo (MCMC) sampler or optimization procedure until the best match (or posterior distribution of matches) between simulated data and real data is found.

While effective, MCMC methods for model fitting can become exceedingly slow to fit, especially when the model includes a lengthy stochastic sampling process within each simulated data point like the pricing model we presented above. Furthermore, due to the technical sophistication involved, these methods are relatively inaccessible to psychological and economic researchers who are not already familiar with Bayesian methods and advanced computational modeling techniques. This deepens the training required to use contemporary computational models, a barrier which is already too high for many researchers to invest the time required to learn these tools.

In this paper, we present a potential method to overcome these challenges that uses deep neural networks to automate the process of parameter estimation (Radev et al., 2020b,a). In this approach, we simulate a large volume of data from randomly drawn parameters of the model (in our case, 100,000 simulated data sets) and teach a neural network to learn the relationship between observed data and the parameters used to generate them. Once the network learns the function that maps data to generative parameters, it can be used to estimate the best-fit parameters of real data simply by using the real data as an input to the network. This means that there is a large front-end investment in simulation, allowing us to "amortize" simulation-based computations for intractable models (Radev et al., 2020a,b; Lueckmann et al., 2019; Cranmer et al., 2020; Fengler et al., 2021).

We wish to emphasize that deep learning is being used here as a tool for parameter estimation — i.e., we want to obtain estimates of a generative model based on a set of behavioral data. The deep learning component allows us to (1) directly map observed behavior (price responses) onto the parameters of the behavioral/cognitive model described in the "Modeling approach" section above, and (2) to estimate the error in these estimates in terms of the mean squared error or posterior variance. Therefore, we do not inherit the "black box" properties of deep learning into our theory of pricing behavior, but permit it only to apply to our method of parameter estimation.

An outline of this approach is shown in Fig. 2. Essentially, a cognitive model is used as a simulator with an input vector of parameters θ sampled from a set of informative priors, generating "observed" data as an output vector randomly sampled through a sequence of unobserved states. This yields a set of known parameters and vast amounts of data that emanate from them, rendering parametric approximation possible for a particular model through the training of a deep neural network. The simulation process is simply reversed, with the simulated data set serving as an input for the network and the corresponding vector of parameters serving as an output. As such, the network will try to approximate the underlying function between data and parameters to map the former onto the latter. As opposed to traditional sampling algorithms, the neural network will learn the probabilistic relationship by transforming the random input vector into a series of samples from a target probability distribution in a process that is optimized by the network's weights; a technique often referred to as neural sampling (Hu et al., 2018).

In theory, any cognitive model could be used in this approach because of how the universal approximation theorem lends a universality trait to neural networks with at least one hidden layer and finite weight values (Cybenko, 1989; Zhou, 2020). The theorem posits that a neural network can successfully approximate *any* arbitrarily complex continuous function, thus identifying the probabilistic pattern between a set of synthetic data and its given parameters. This generates a round of predictions made by the network that can be tested against the originals. If the parameters produced by the neural network closely match the ones used to generate the data, the network successfully learned the underlying relationship imposed by the cognitive model, providing us a quick and reliable alternative for parameter estimation.

In the following sections, we examine how this approach to parameter estimation can be combined with information-rich research designs, such as pricing problems, to yield quick and effective estimates of a participant's risk preferences. A more complete and formal description of the deep learning methods we use to fit the behavioral (pricing) model, along with a preliminary comparison among different architectures and training approaches, is provided in the Algorithmic Model section of the Methods section. To illustrate the effectiveness of this approach and gain a better understanding of the range of risk preferences among a broad population, we apply this approach to model the behavior of a demographically representative sample of U.S. adults. This is also provided in the Methods section.

2. Methods

This paper features both a new approach to fitting the behavioral pricing model and an application to a large number of participants. The goal of the application to real data is to illustrate (a) how deep learning can allow previously intractable simulation-based models to be accurately estimated, and (b) to show that using it to fit a pricing model yields interesting insights that could not have been practically obtained using traditional methods. To address the first component, we focus on the structure of the deep learning algorithm for model fitting (Algorithmic Model). Then we focus on presenting the details of the empirical study (Application to Empirical Data) before continuing onto the results, where we apply the new fitting approach to generate insights about the empirical data.

2.1. Algorithmic model

The approach to using a neural network to approximate the relationship between data and model parameters was driven by the potential for these methods to closely approximate the functional relationship between the two. The initial universal approximation theorem proving that this is theoretically possible referred to learning relationships through a single layer perceptron with a scalar output (Cybenko, 1989), but it was later extended to deep neural networks (DNN) (Zhou, 2020). As such, we chose DNN for model development done using MathWorks' Deep Learning Toolbox in MATLAB. The model architecture was composed by one sequence input layer for sequence data and three fully connected hidden tanh layers of 100, 67 and 33 nodes each. We arrived at this architecture by initially testing 2, 3, 4, 5, and 6 layer networks and observing that the gain in model fit was minimal after moving to 3 hidden layers; and by noting that other activation functions resulted in non-convergence of the network. Below, we provide comparisons between this approach and alternative ones with flat layer structures (i.e., same number of nodes in each hidden layer) and varying numbers of nodes in the hidden layers.

A fully connected Rectified Linear Unit (ReLU) output layer was used to impose a threshold operation to each incoming element with a negative value by setting its value to zero, as all of the parameter values we estimate should theoretically be positive. For the loss function, we used a Root Mean Square Propagation algorithm with a weight decay (L2 Regularization). As with the layer structure, this appeared to be the most effective at achieving model convergence compared to stochastic gradient descent and Adam algorithms (Murphy, 2012; Kingma and Ba, 2014).

For model training, we ran 100,000 simulations to generate data from the pricing model. Each simulation involved a randomly drawn combination of the model parameters constrained on fixed priors as demonstrated below.

$$\alpha \sim Gamma(10, .08)$$

$$\beta \sim Unif(0, 1)$$

$$\nu \sim Unif(0, .99)$$

$$\theta \sim Gamma(5, .6)$$

$$\tau \sim Unif(.1, 2)$$

The learning rate was fixed at .001, while max epochs were fixed to 5000 with a mini batch size of 1000 and a validation frequency of 100. Once the first network was trained and tested, a second neural network was trained to estimate the error variance. Consequently, the participants' response times, their response prices along with their associated summary statistics will serve as an input for the second network, while the squared difference between the initial parameters and the first networks' predictions will serve as an output. Essentially, when paired together the two networks are not only capable of providing parameter estimates for a particular data set, but provide the percentage of error that is expected for the given data set. The error variance network was identical to the parameter estimation network.

Table 1
Parameter distributions for alternative training sets. Parameterization of the distributions that model parameters were drawn from for the "narrow" and "wide" data sets.

Parameter	Distribution	Narrow	Wide
α	Gamma	(4, .1)	(4, .6)
β	Beta	(.5, .5)	(1, 1)
ν	Beta	(.5, .5)	(1, 1)
θ	Gamma	(4, 1.25)	(4, 7.5)
τ	Gamma	(1.5, 0.75)	(1.5, 4.5)

2.1.1. Alternative training sets

The parameters on which the network was trained, specified above, were designed to cover ranges we might commonly expect in studies of dynamic choice and risky choice. For example, the values for α reflect those that are commonly found in studies using prospect theory or expected utility (Kahneman and Tversky, 1979; Tversky and Kahneman, 1992; Scheibehenne and Pachur, 2015), while the thresholds and non-decision times correspond to estimates obtained using the same type of pricing and judgment tasks (Yu et al., 2015; Kvam et al., 2015; Kvam and Busemeyer, 2020; Kvam et al., 2021). The priors for the remaining parameters correspond to maximal entropy over their valid ranges, which is [0,1] for both bias and bias strength.

These training sets might not match the "true" distributions of these parameters in the population. In general, a weaker match between training sets and the target population will lead to a worse fit; the same is true of prior distributions in a Bayesian framework. Before we fit the data, it is unclear what the prior distributions or training sets should be, meaning that we are likely to experience some degree of mismatch when working with any real data set. To examine the impact such discrepancy might have, we explore two other simulated data sets (priors) that do not match the aforementioned network architecture and test how well it can recover the parameters of these diverging data sets.

Specifically, we test two types of diverging data sets: a "narrow" data set where the true distribution of parameters in a population has lower variance than the one used to train the network, and a "wide" data set where the true distribution of parameters in a population has greater variance than the one used to train the network. The former data set challenges the network to do *interpolation* – predicting values within its trained range without inflating their variance. The latter challenges the network to do *extrapolation* – predicting values outside its trained range or with greater variability than the training set. We also test cases where a network trained on the narrow data set or wide data set is able to predict the parameters of the other two data sets.

Formally, the training data for the narrow and wide data sets were generated from the distributions in Table 1.

The results are shown in Table 2. In general, the original network demonstrated good parameter recovery when predicting data generated from both the narrow and wide range of parameter distributions ($r > .72$). Additionally, the networks trained on the narrow and wide distributions only struggled when the degree of mismatch between training and testing distributions was greatest. Specifically, The network trained on the narrow training set performed well when predicting its matched distributions (narrow: $r > .83$) and the original distributions ($r > .82$) but struggled to predict α ($r = .34$) and θ ($r = .56$) from the wide distributions. Like the network trained on the narrow set, the network trained on the wide set performed well when predicting its matched distributions ($r > .92$) and the original distributions ($r > .84$), but performed worse when predicting α ($r = .35$) and θ ($r = .48$) from the narrow distributions.

Regarding error estimation, each network performed best when predicting parameter estimation error from the set that matched the training distributions (original: $r > .36$; narrow: $r > .35$; wide: $r > .36$). However, performance for all networks was somewhat worse when predicting parameter error from mismatched distributions, particularly when the degree of mismatch was greatest (e.g., trained on narrow, tested on wide).

In summary, parameter and error recovery suggests that the network is able to perform both mild interpolation and extrapolation, but that a high degree of mismatch in either direction can worsen performance. Thus, prior distributions should be informed by previous findings in the literature and theoretical model constraints, rather than the highly uninformative priors commonly used in a Bayesian framework, to minimize the degree to mismatch between training distributions and the expected true parameter distributions of empirical data.

2.1.2. Alternative network architectures

In addition to different training sets, it is also possible to vary the structure of the neural network that we use for both parameter estimation and error estimation. While there are many potential hyperparameters to optimize (learning rate, regularization, training length/number of epochs, batch sizes, and so on), we chose to focus on two of the most important architectural choices by examining the number of hidden nodes and how they were organized in the hidden layers. Note that we were unable to achieve network convergence during training when using ReLU activation functions — we are unsure of exactly why this occurred. However, as we show below, hyperbolic tangent (tanh) activation was clearly sufficient to enable high-quality parameter recovery, indicating that there are few network design choices that could have improved performance. In additional tanh allows for a data compression phase during deep learning training that linear activation units do not (Saxe et al., 2019).

To examine the effects of the structure and size of the hidden layers, we examined four total layer sizes — networks with 25, 50, 100, or 200 total hidden nodes. We also examined whether their organization affected performance, by testing both a flat layer structure (same number of nodes in each layer) or an information-compression layer structure (iteratively reducing the number of

Table 2
Comparison of parameter and error recovery for alternative training sets. Networks trained on the original, "wide", and "narrow" training sets were tested on the validation set from the original, "wide", and "narrow" distributions. The correlation coefficient for parameter and error recovery for each model parameter is reported.

Prior Distribution		Parameter estimation					Error estimation				
Trained range	Prediction range	α	β	ν	θ	τ	α	β	ν	θ	τ
Normal	Normal	0.94	0.92	0.99	0.94	0.9	0.44	0.53	0.4	0.37	0.5
Normal	Narrow	0.76	0.86	0.98	0.85	0.83	0.24	0.38	0.11	0.33	0.42
Normal	Wide	0.73	0.9	0.95	0.81	0.85	0.22	0.31	0.18	0.26	0.25
Narrow	Normal	0.83	0.9	0.98	0.83	0.86	0.28	0.44	0.29	0.15	0.4
Narrow	Narrow	0.88	0.86	0.99	0.87	0.84	0.4	0.46	0.45	0.36	0.48
Narrow	Wide	0.34	0.81	0.84	0.56	0.79	0.03	0.23	0.15	0.07	0.24
Wide	Normal	0.88	0.88	0.97	0.86	0.85	0.39	0.45	0.18	0.13	0.2
Wide	Narrow	0.35	0.71	0.89	0.48	0.65	0.21	0.32	0.1	0.14	0.1
Wide	Wide	0.93	0.94	0.99	0.94	0.95	0.48	0.5	0.37	0.42	0.5

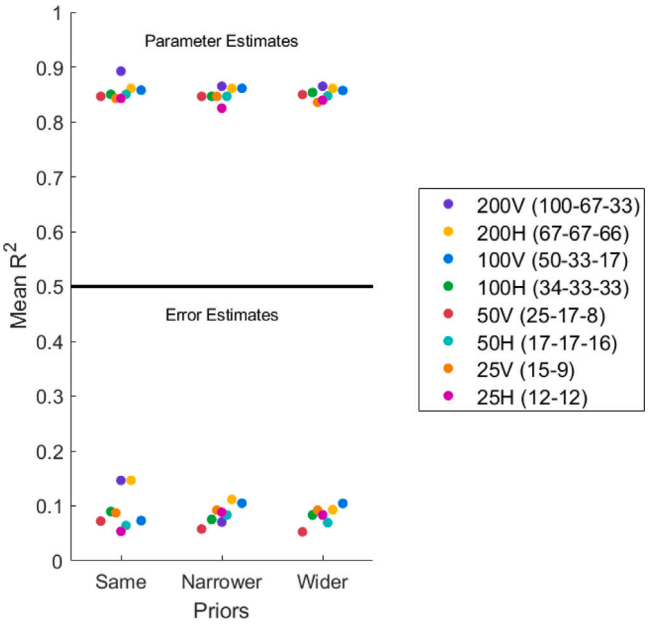


Fig. 3. Comparison of the performance of different neural network hidden layer structures in terms of their ability to account for variability in model parameters (top) and predict the degree of error between the predicted estimates and the true parameter values (bottom). Each network is described in terms of the total number of nodes in its hidden layers (25, 50, 100, or 200) and whether it had a flat layer structure with the same number of nodes in each hidden layer (H) or had an information-reducing structure with fewer nodes in each successive hidden layer (V). The number of nodes in each hidden layer, first to last layer, for each model is provided in parenthesis in the legend.

nodes per hidden layer). Theoretically, the information-compression structure should be better able to take the information provided in the inputs (behavior) and remove redundant information with each hidden layer by forcing the network to lose some information in each hidden layer (Saxe et al., 2019). A common heuristic is to reduce the size of each layer by 25%–50% (Walczak and Cerpa, 1999; Khoong, 2020), which is what we pursued in the information-compression layer structures.

Specifically, we tested networks with 2–3 hidden layers (depending on the total number of nodes; all models needed at least 5 nodes in the final hidden layer). The flat-layer networks divided their total nodes evenly across layers. For example, the 100-node flat-layer network consisted of 34 nodes in the first hidden layer, 33 in the second, and 33 in the third hidden layer. The information-compression network reduced the number of hidden nodes by 33% from the first to the second layer and 50% from the second to the third layer. For example, the 100-node information-compression network consisted of 50 nodes in the first hidden layer, 33 nodes in the second hidden layer, and 17 nodes in the third hidden layer.

The results are shown in Fig. 3. For each network architecture, we examined what proportion of variability in model parameters (generated from the pricing model; see Model-based results below for details on parameter recovery) could be captured by a trained neural network with the specified architecture. This is quantified by the R^2 value of the network, which corresponds to what proportion of variability in model parameters (where the variability comes from the known parameters of the generative model/training set) was predicted by the network. These are shown toward the top of the plot in Fig. 3. We also quantified what proportion of the error variance (squared difference between predicted model parameters and true model parameters) could be

captured by a second error-variance network with the same architecture. The performance of the error networks is shown toward the bottom of Fig. 3.

In general, there was not much difference among different model architectures in terms of how well they could recover the parameters of the generative model — each of the eight architectures performed essentially the same, with some drop in performance for the 25-node flat-layer network. The 200-node information-compression layer structure, which had 100 nodes in the first hidden layer, 67 in the second, and 33 in the third, appeared to perform the best by a slight margin. Therefore, this is the network architecture that we used to carry out the model fitting in the following sections.

We return to the performance of the parameter estimation network in the Results (Model-based results subsection). For now, it is sufficient to conclude that even very simple model architectures can be trained to map behavior onto generative model parameters, and that our approach is relatively robust to differences between the training set and the true data it is used to fit.

2.2. Empirical data

The approach to model fitting we outline above has several advantages relative to traditional model fitting approaches. First, it enables us to use the pricing model outlined in the Behavioral Model section. Second, even when we apply less traditional methods like probability density approximation in combination with approximate Bayesian computation or MCMC methods (Turner and Sederberg, 2014), the pricing model typically takes hours or even days to fit to even a single participant (Kvam and Busemeyer, 2020; Kvam and Turner, 2021) and requires a large volume of data from each individual in order to reliably capture risk preferences from pricing responses. By contrast, using deep learning to fit the model permits us to fit each participant instantaneously, and minimizes the amount of data required from each participant. Therefore, to exhibit the usefulness of the deep learning approach to fitting the pricing model, we show that it can fit a data set with a large number of participants but a small number of trials per person. Such a data set would previously have been effectively impossible to fit using the pricing model.

Specifically, a total of 300 participants took part in the study, and consisted of a demographically representative sample with respect to age, race, and sex in the United States. In total, there were 153 female/147 male participants. They had a mean age of 44.69 (SD = 16.27) with 35 in the age range of 18–27, 53 in the ages 28–37, 49 in the range of 38–47, 51 in the ages 48–57, and 94 aged 58+. This included 205 participants who identified as White, 45 as Black, 25 as Asian/Pacific Islander, 15 as Mixed, and 10 as Other. All participants were recruited through Prolific Academic and completed informed consent before beginning the study. The study received human subjects exempt approval through the University of Florida IRB#201902817. The data are publicly available at osf.io/4wt8d/.

Each participant completed a pricing experiment, which consisted of pricing 20 risky payoffs and a demographics questionnaire. Participants were provided with a debriefing explaining the purpose of the experiment at the end of the session. Experimental sessions were approximately 15 min long and delivered online through Qualtrics. Participants were compensated at a rate of $15 per hour (prorated based on how long they took), or $3.75 for 15 min.

When pricing risky payoffs, participants were shown a payoff (e.g., $12) and a chance of winning that payoff (e.g., 35%). They were instructed to use a slider ranging from $0 - $20 to indicate the hypothetical maximum amount they would be willing to pay for a chance to receive each risky payoff. Due to the imprecise nature of using a slider, participants were asked to respond within 10 cents of their true preferred amount. The amount of the payoffs ranged from $4 to $20 by increments of $4 (i.e., $4, $8, $12, $16, or $20). Each amount was crossed with a 15%, 35%, 65%, and 85% chance of receiving the payoff in a full factorial design. Participants were informed that if they did not win the payoff on a particular item, that they would receive $0, meaning that "$12, 15%" carried an implicit "$0, 85%" as the alternative outcome. These risky payoffs were presented one at a time to each participant in a random order.

The demographics questionnaire assessed age, gender, ethnicity, income, education, and political orientation regarding economic and social issues. This questionnaire confirmed the make-up of the sample reported above.

3. Results

Before moving on to the model-based results, it is helpful to first outline some basic components of the behavioral results of the pricing task. The analyses of the basic behavioral results were carried out in JASP (JASP Team, 2020) and MATLAB/JAGS (Plummer, 2003). We report both the mean values of each parameter estimate (for both statistical and model-based analyses) and the 95% highest-density interval [HDI] giving the range containing the 95% most probable posterior values for each estimated parameter. Unless otherwise specified, these estimates are based on uninformative priors and in the case of MCMC, groups of 4 chains of at least 1000 samples. In addition, for certain analyses we also report the Bayes factor in favor of the alternative hypothesis, BF_{10}, which directly quantifies the likelihood of a nonzero relative to a null effect for a particular parameter or coefficient. This is calculated using a Savage–Dickey approximation (Wagenmakers et al., 2010), quantifying the difference in credibility between the prior and posterior distributions at a null effect size of $b = 0$. Note that we use b to signify linear coefficients below.

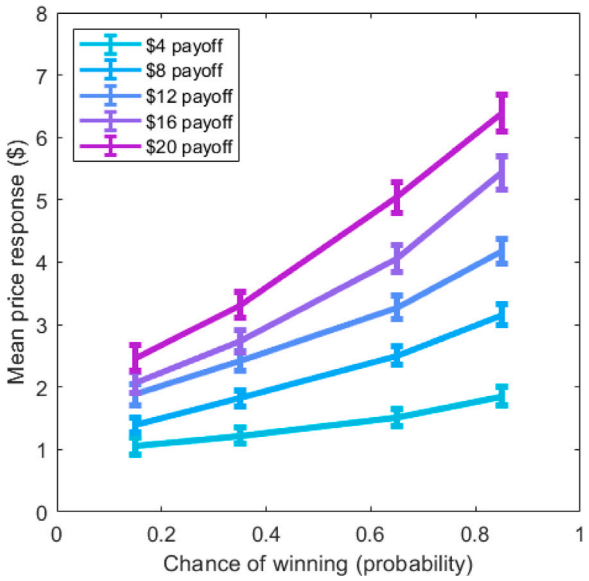


Fig. 4. Patterns of mean price responses (y) as a function of the probability of winning (x) and the payoff (different colors). Bars represent one unit of standard error. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

3.1. Empirical data

Beyond the modeling approach, the data give us the opportunity to quantify the relationships among risk, payoff, and valuation among a representative sample of U.S. participants. This is not the main focus of the paper, which centers mainly on the deep learning approach to fitting a complex pricing model, but it is data that is directly relevant to our understanding of value and preference. Those readers who are not interested in these empirical findings can skip to the model-based results. However, we would consider it a shame to not even characterize what are potentially rich empirical data on pricing behavior. To this end, we report the main effects and interactions of manipulations of probability and payoff on both the price participants were willing to pay for a prospect and the time it takes them to respond. For each of these analyses, we carried out a hierarchical Bayesian linear regression estimating the individual and group-level effects of each manipulation. The intercept, coefficient for the effect of payoff, coefficient for the effect of probability, and coefficient for the interaction were estimated for each participant, and each participant's parameters were pulled from a group-level distribution for each parameter. We used uninformative priors for the group-level distributions, such that the individual-level parameters b_i for each participant i were drawn from a group-level distribution specified by M_b and S_b:

$$b_i \sim Normal(M_b, S_b)$$

$$M_b \sim Normal(0, 100)$$

$$S_b \sim Exp(100)$$

These same priors and group-level distributions were used for each parameter estimated in the linear regression. Below, we quantify the group-level mean effects of each manipulation – i.e., the estimates of M_b – to understand how each one impacted prices and response times [RTs]. For each one, we present the 95% HDI as well as a Bayes factor computed using a Savage–Dickey approximation (Wagenmakers et al., 2010), quantifying the degree of support for each coefficient being nonzero. All of the predictors and outcomes were standardized prior to analysis.

The general trends are highlighted in Figs. 4 and 5. For price responses, all three effects were credible. The highest density intervals did not include zero and the Bayes factors approached ∞. Note that for these very large Bayes factors, we simply note once its value is greater than 10,000. The effects of payoff and probability were nearly identical, with probability having mean effect of M(Payoff) = 0.29 (95% HDI = $[0.26, 0.32]$) and probability having a group-level mean effect of M(Probability) = 0.29 (95% HDI = $[0.25, 0.32]$). The interaction effect was also substantial, with a mean of M(Interaction) = 0.13 (95% HDI = $[0.11, 0.15]$). The Bayes factors for all three coefficients were greater than 10,000, indicating extremely strong support for all three effects. Thus, the results suggest that subjects offered higher prices to acquire a bet as the potential payoff increased and the probability of acquiring the bet became larger(safer). The positive and credible effect of the interaction term predicts responses above and beyond the additive effects of payoff and probability separately.

For the analysis of response times, we log-transformed RTs before standardizing to make them roughly normally distributed, then used the same model structure as for the price responses to quantify the effects of payoff, probability, and their interaction. As before,

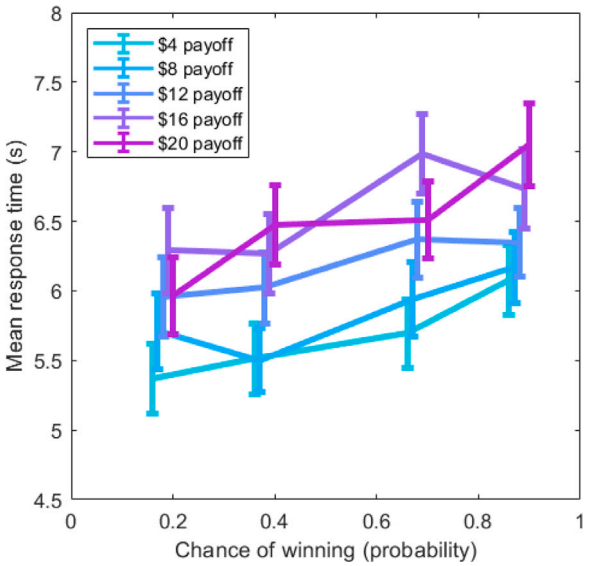


Fig. 5. Patterns of mean response times (y) as a function of the probability of winning (x) and the payoff (different colors). Bars represent one unit of standard error. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

we report the group-level effects from a hierarchical Bayesian analysis. Manipulations of payoff had a mean effect of $M(\text{Payoff}) = 0.08$ (95% HDI = $[0.06, 0.10]$) and probability had a slightly larger effect of $M(\text{Probability}) = 0.12$ (95% HDI = $[0.10, 0.14]$). Unlike the price response analysis, there was no effect of their interaction, $M(\text{Interaction}) = 0.01$ (95% HDI = $[-0.01, 0.03]$). Accordingly, the Bayes factors for both payoff and probability exceeded 10,000 while the Bayes factor for the interaction was $BF_{10}(Interaction) = 0.00047$, indicating strong support for the null hypothesis (i.e. no interaction).

The effects of payoff and probability on response time are somewhat unexpected based on past results, where response times mainly varied with probability such that high-variance gambles (close to 50%) resulted in longer response times. There are a couple potential reasons for this. One model-based interpretation is that participants anchored fairly low on the scale, and thus it took more time to accumulate toward the high end of the scale. This can happen when participants are used to repeatedly making low responses. This appeared to be the case here, as mean prices were mostly in the $1–7 range (see Fig. 4), meaning that the anchors may have simply been set too low to quickly make high-price responses.

Another possible interpretation as to why safer bets or larger payoffs led to longer response times could be attributed to the response scale. Participants had to use a slider to designate their desired prices. They were asked to click on the scale at the desired location, but some participants may have opted to drag the slider (from 0), meaning that they would have to drag the slider further down the number line to enter their choice, thus possibly leading to longer response times. Alternatively, participants may have simply placed their cursor nearer to low values (as we suggested above, this could be attributed to a prior bias toward low prices), resulting in slower RTs to high values.

In the appendix (Appendix), we train the network on a version of the model that does not use response times as inputs, meaning that it can estimate parameters without relying on response times that might be affected by idiosyncrasies of the response scale.

3.2. Model-based results

The first step in ensuring that our modeling and estimation approach is working is to evaluate how well it is able to correctly estimate known parameters — known as parameter recovery (Heathcote et al., 2015). While the neural network training process includes both fitting/training and validation sets, which we split in half so that there were 50,000 artificial data sets in each, it is also worth ensuring that the model will work on out-of-sample data. To test both in-sample and out-of-sample predictions of the model, we took all 50,000 simulated data sets from the training set plus an additional 50,000 data sets simulated from the same set of prior distribution of model parameters. We then used the trained network to fit each of these data sets and compare them to the generative parameters. The goal here is to simply ensure that the network is capturing known differences in each latent component of behavior.

The results of this parameter recovery are shown in Fig. 6 with in-sample data show in blue and out-of-sample data shown in orange. We only plot a sample of 300 data points so as not to overcrowd the plot, but the results are clear. Both in-sample and out-of-sample parameters are recovered extremely well for prices coming from only 20 responses and response times. The highest-fidelity recovery is for the utility parameter and the threshold, which are often the two most important parameters in the model, with utility being the most directly relevant to risk preferences. Conversely, the most difficult to recover is non-decision time, which is a nuisance parameter and arguably the least important to estimate precisely and accurately. Aside from non-decision time,

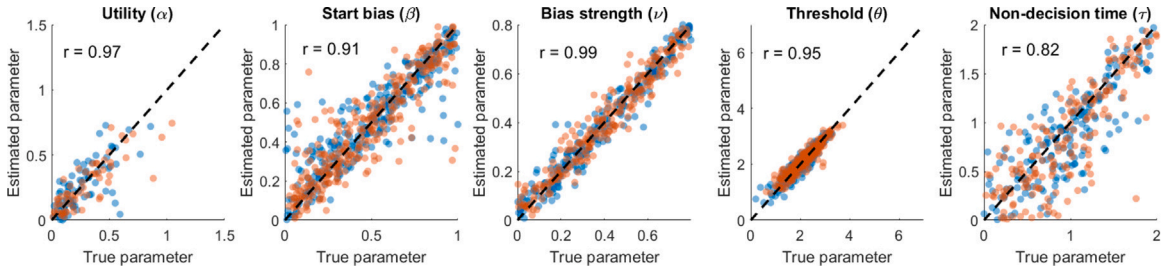


Fig. 6. True (x) versus network-estimated (y) parameter values for the price accumulation model. Blue points correspond to values that were in the training set, while orange dots correspond to an out-of-sample test set. For visual clarity, we only plot 300 random points out of the 50,000 from each set. Pearson linear correlation values (r) are provided for the relationship between true and estimated values for the out-of-sample test set. The diagonal dashed black line corresponds to perfect recovery of the true parameters. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

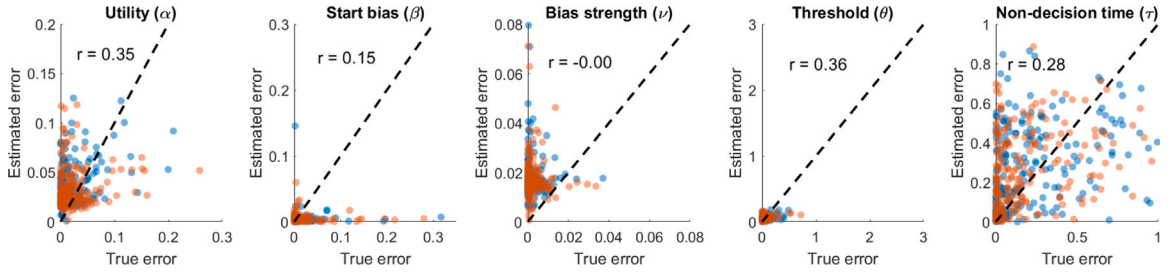


Fig. 7. True (x) and estimated (y) posterior error of the parameter estimates generated by the estimation network. Each panel shows the ability of a second error-estimation network to approximate the degree of error of the first estimation network (Fig. 6). As before, blue dots correspond to within-sample predictions from the training set, while orange dots correspond to out-of-sample predictions on which the network was not trained. The dashed black line corresponds to perfect estimation of the squared error between true and estimated values. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

there appear to be no directional biases in the error either, and errors are roughly normally distributed around the true parameters (i.e., the y-prediction in Fig. 6 is centered on the dotted black line).

In addition to providing the best estimates it can of generative model parameters, we can also train a second network to estimate the degree of error in the first network. This allows us some meta-level insight into the parameter estimates, corresponding to an estimate of the posterior error on each parameter (Radev et al., 2020a). Certain parts of the parameter space result in noisier data. Likewise, certain sets of data can be generated by a greater range of possible parameter values. Therefore, our estimates of the posterior variance should be informed both by the data and by the estimate of the first network.

To create this error-estimation network, we used the same architecture described above in the Algorithmic Model section, except that we also included the parameter estimate generated by the first estimation network. The desired output from the network was the squared difference between the true parameter and the parameter estimated by the first network.

The result is shown in Fig. 7. While not quite as impressive as the direct parameter estimation network, there was at least a positive relationship between the estimated error and the true error between the estimate and the true parameter value. Part of the "problem" so to say is that there is not much error in the first network. For example, the performance of the error estimation network is low for bias strength primarily because the error never exceeded .05 on a [0,1] scale. Ironically, the good performance of the first estimation network makes the job of the second error estimation network much harder. However, this network does still give us a rough estimate of the posterior variance of each parameter, yielding insight into how certain we can be of the estimates generated by the first network.

Overall, the results suggest that this network can reliably capture performance on the task in terms of the parameters of the pricing model introduced in the "Behavioral model" section. As we noted above, fitting such a model is extraordinarily difficult using traditional methods, yet the deep neural network was able to capture the relationship between behavior and model parameters with relative ease.

3.2.1. Estimates

Now that we have trained the network to estimate price accumulation model parameters, the next step is to apply it to the data we collected to make inferences about the risk preferences of our participants. To do so, we simply fed the true data from each participant as an input to the trained network and used the observed output as an estimate of the parameters of the price accumulation network. For completeness, we use the full RT-inclusive model, as it contains more information on thresholds and non-decision time than the model excluding response times. Estimating the parameters for each individual in the data set is effectively

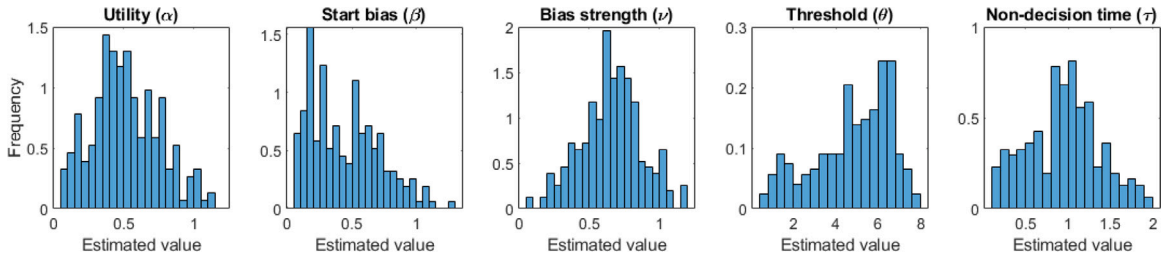


Fig. 8. Distributions of parameter estimates among participants in the study.

instantaneous, as all the difficult front-end work of network training and tuning removes the back-end work of estimating the model separately to each participant. As a result, the total time for model fitting is reduced from several hours or even days per participant (which would result in several months to fit our sample size of 300 participants) to the length of time it takes to simulate and train the neural network — typically around 10–20 min in total when using efficient simulation methods and training algorithms (Evans, 2019).

The results are shown in Fig. 8. Most participants had values of the utility parameter around $M(\alpha) = 0.48$ ($SD = 0.32$), corresponding to relatively strong risk aversion on average but including fairly substantial individual differences. In terms of pricing, a low value of α means that an average participant would have relatively large variability in their high price responses relative to low price responses. In terms of choices, we would expect these same participants to favor safe options over risky ones of similar expected value.

Participants seemed to anchor on the outcome of each gamble, with many start bias ranging across the possible values ($M(\beta) = 0.36$, $SD = 0.32$) and bias strength generally well above zero ($M(\nu) = 0.66$, $SD = 0.26$). This allows the model to produce violations of procedure invariance that appear to be responsible for preference reversals (Tversky et al., 1990), resulting from anchoring on the high payoffs.

Unusual in this particular data set was the high thresholds ($M(\theta) = 4.65$, $SD = 1.96$) and non-decision times ($M(\tau) = 0.96$, $SD = 0.61$) participants appeared to have. This is likely attributable to the way that participants entered their responses using a linear slider — as opposed to typical laboratory experiments where we can precisely control RT by using a radial scale (Kvam and Busemeyer, 2020; Kvam et al., 2015; Pleskac and Busemeyer, 2010). The most likely explanation for the slow responses in this paradigm, and for the longer response times for higher values (as noted above), is simply that participants took a longer time to click on the slider than they would to respond on a radial response scale. Fortunately, this has a minimal effect on estimates of start points and utility parameter estimates, which can be inferred from price responses alone similar to an utility plus anchor-integration model (Turner and Schley, 2016).

Having participant-level estimates from a large representative sample also allows us to take a look at how participant demographics interact with risk preferences and other parameters of the model. To test for differences in model parameters as a function of participant demographics, we ran a Bayesian linear regression predicting each model parameter as a function of age, gender, ethnicity, level of education, and income. Gender, ethnicity, and level of education were dummy coded with each different group having its own indicator variable. However, none of the dummy coded variables turned out to have a significant/credible effect, so we do not detail them further.

The only two variables that appeared to affect behavior on the pricing task, in terms of model parameter estimates, were age and income. Older participants tended to have slightly higher thresholds on the task ($M(b) = .13$, 95% HDI $= [.01, .25]$, $BF_{10} = .70$, reflecting a more cautious approach to pricing that led them to longer response times and prices that were closer to expected utility. This likely reflects a general tendency toward response caution that has been observed in older participants across a range of decision-making tasks (Starns and Ratcliff, 2010), which appear related to differences in neural activity in the striatum (Kühn et al., 2011).

There was also an effect of income on start point strength, such that participants with higher incomes tended to anchor less strongly on the payoff ($M(b) = -.14$, 95% HDI $= [-.01, -.25]$, $BF_{10} = .81$). This may correspond to greater financial literacy or a greater willingness to take financial risks, such that participants with higher incomes do not need to use the anchor as much as participants with low incomes. This lines up with work by Kato and Hidano (2007) suggesting that low-income participants are more susceptible to anchoring effects, although we did not replicate the effect of gender from their findings. In any case, the threshold and start point effects appear to reflect sensible relationships with model parameters that correspond to previous findings, indicating that the risk preferences assessed by the model hold at least some degree of predictive validity.

Put together, it is clear that the deep learning method for model fitting enables straightforward estimation of model parameters. It cuts the time required for model fitting down by several orders of magnitude — taking a PDA+MCMC procedure that would likely take months to fit 300 participants down to mere seconds. Beyond the pre-training time, which is typically on the order of 5–20 min depending on the structure of the network and training est, the neural network does not require any additional time to fit participants, enabling us to efficiently fit large-N data sets of participants.

4. Discussion

Indisputably, when studying interconnected and dynamic processes, there is a need for complex models that can adequately account for the phenomena in question. In our case with the valuation model, we were able to extract latent – and otherwise inaccessible – information about risky preferences while offering an efficient alternative method to measuring them. As highlighted in the previous section, the model parameters can be used to provide insight – more than what we could have gotten from a choice experiment (Haines et al., 2020) — about the concept of preference, while relating it to underlying cognitive mechanisms. The resulting individual and group-level estimates would have been extremely hard to recover through ABC methods and very computationally expensive. Even fitting relatively straightforward models of continuous responses can take hours or even days (Kvam and Turner, 2021), meaning that fitting 300 participants would be essentially intractable.

Thus, the primary aim of the paper was to provide a tractable alternative to traditional methods of model fitting for a model without an analytic likelihood function. Our approach amortizes parameter inference via deep learning, drastically reducing the time and computation required for parameter estimation. This circumvents usability limitations for intractable models, meaning that we can test a wider variety of interesting but computationally difficult models moving forward. Similar approaches have been developed focusing on approximating the likelihood function or approximating the posterior using summary statistics as inputs (Radev et al., 2020a,b, 2021), but our approach allows for the entire data set for each individual to be used to inform model parameter estimates. While this requires training a network to estimate parameters from a specific model applied to a specific set of decisions, our approach benefits from retaining trial-by-trial information that is lost during summary statistic aggregation, theoretically leading to better parameter estimates. In other words, our approach trades the flexibility to apply across different paradigms for greater estimation accuracy on a specific set of stimuli. While we favor the latter here, other applications will naturally be better served by using different inputs that summarize rather than reproduce the data.

However, the novelty of deep learning assisted simulations for likelihood-free inferences opens not only new possible avenues for research, but also the path to many new questions. This does not come without caveats, of course. The heavy front-end effort of simulating enough data to train the network falls on the initial researcher. This training set naturally biases the network toward whatever parameter values were used to train it, functioning as a prior distribution over likely parameter values that the network can produce. Naturally, this kind of approach is best suited to fixed experimental procedures, as it assumes a constant set of inputs. These are common enough in the psychological and social sciences that such an approach still has significant utility (Kirby et al., 1999; Greenwald et al., 1998). However, in many cases it will make more sense to summarize behavioral data in terms of a consistent set of summary statistics, such as response time quantiles (Heathcote et al., 2002), to allow them to serve as inputs to a neural network. Further work should explore the different ways in which behavioral data might be summarized in terms of a common set of inputs to neural networks, especially simulated data. This would broaden the range of applications of our approach and make it possible to use with many different experimental paradigms.

Another limitation in our current example is the black box nature of the neural network when it comes to mapping the data onto the parameters. One could imagine using two neural networks with different architecture characteristics that arrive at the same results, which poses numerous issues in matters of analysis, because of how the network provides no insights about how it approximates the underlying function f. This is almost antithetical to computational modeling, where generative knowledge and explainability are crucial. Part of this issue is the lack of systematic approaches to neural architectures (activation functions, layer width and depth, and so on) and clear theory on the suitability of different approaches. There are still various possible combinations of architectures that could be tested to identify (sub)optimal designs. Many optimization techniques exist and could be employed to finely tune the algorithmic model, but these do not solve the inherent limitation of interpretability. Our results at least indicate that there are many different network architectures that may all perform similarly well for the same input–output problem.

Additionally, this approach can and should be adapted as more real experiment data is collected. We used largely uninformative priors to generate the training set, to ensure that we had adequate coverage of the range of possible parameter values we could observe in the experiment. However, this choice means that priors were not optimized for the inferences that were made, as there was substantial mismatch between the training set and the observed estimates shown in Fig. 8. While we tested the robustness to different training sets and priors (“Alternative training sets” section), it is still best if the training set is as closely matched as possible to the behavioral data the modeler intends to fit. In future work using a similar paradigm, however, we can use these “posterior” estimates to create the new training set, retrain the network, and obtain more appropriate parameter estimates for similar data sets. Network training therefore becomes an iterative process, where more data allows us to improve the training set, the training set serves as a prior that better matches the true generative distribution, allowing us to make better inferences on the next data set, and so on. Further use of these methods therefore ensures that their performance will improve over time, paving the way for more effective parameter estimation on new and more powerful models. One way in which this might be improved is by including hierarchical constraints on parameter estimation, as in hierarchical Bayesian approaches e.g., Kruschke, 2014. This would potentially solve issues related to mismatched priors and training sets by allowing individual-level estimates to be informed by a hyper-prior or group-level distribution of a particular parameter, and vice versa. These constraints are particularly helpful for data sets with few trials but many participants (Molloy et al., 2020) like the ones we examined here. Such approaches are now in their infancy (Elsemüller et al., 2023), and we look forward to further development and application to problems like the one we face with simulation-based pricing models.

Despite the limitations we mention, the automated approach to model fitting overall alleviated barriers related to dynamic and stochastic theories of subjective value. Without the ability to fit models of price, we would have to resort to models based on binary choice, many of which require assuming procedure invariance which is clearly violated in many paradigms Lichtenstein and

Slovic, 1971; Tversky et al., 1990; Johnson and Busemeyer, 2005; Kvam and Busemeyer, 2020. While still useful and rather flexible, further use of choice models would severely limit our ability to account for the rich complexity that arises in paradigms like pricing. Naturally, individuals make choices every day – from products to careers and relationships – but the choices are rarely static, our preferences rarely constant and our options frequently numerous. Ignoring this richness and complexity makes it difficult to extract or disentangle crucial determinants of behavior, something Tony recognized and consistently illustrated in his work (Marley, 1989; Marley and Colonius, 1992; Marley and Louviere, 2005; Marley and Regenwetter, 2016; Kvam et al., 2023; Lacouture and Marley, 2004). The amount of information and the depth that we can glean from dynamic models of pricing far outstrips that of a binary choice, yet fully realizing widespread use of the sort of proposals Tony put forward requires new and innovative methods to address their computational challenges. We have shown here that the various trade-offs between models, such as practicality and theoretical depth, are alleviated with the use of artificial intelligence. Automated model fitting via deep neural networks can accelerate research as it expands not only the usability of more robust methods but also their accessibility to a broader audience.

CRediT authorship contribution statement

Konstantina Sokratous: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Writing – original draft, Writing – review & editing. **Anderson K. Fitch:** Investigation, Methodology, Resources, Writing – original draft, Writing – review & editing. **Peter D. Kvam:** Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Supervision, Validation, Visualization, Writing – original draft, Writing – review & edition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by a graduate fellowship to K. S. and a SEED grant to P. D. K. from the University of Florida, United States of America Informatics Institute, and a grant from the National Science Foundation, United States of America (SES-2237119) to P. D. K.

Appendix. Price-only network

A potential concern with the network incorporating both price response and response times is that the response times detract from, rather than add to, the potential fit of the model. In other cases, a researcher may be unable to record or save response times accurately. For example, we were only able to assess a proxy for response time by taking the time of the last click on the linear scale presented to participants. While participants perfectly following the directions would have clicked on their response and then hit the page submit, it seems likely that some participants did not follow this perfectly, either by making an additional click or by dragging the slider rather than clicking the scale. This seems supported by the response time analyses presented above, where higher responses resulted in longer response times.

Another reason to avoid incorporating response times in our estimates of risk preferences is that they are much more likely to yield outlier responses than willingness-to-pay values. Price responses are bounded on a \$0–20 scale physically and theoretically bounded on \$0-x when x is the maximum payoff for a particular trial (Levy, 1992). Conversely, response times are bounded on $[0, \infty]$, meaning that extreme outliers can be observed. While such outliers can be removed when cleaning the data, using the DNN approach to parameter estimation means that we want to have every trial as an input to the network. With so few trials, any estimation approach will be highly sensitive to outlying response times, meaning that a participant mis-clicking and missing the scale or failing to pay attention for 30 s or so can drastically alter estimates of thresholds and non-decision time, with knock-on effects for estimates of other parameters. Principled rules for data cleaning can help remove these outlying response times (Rahm and Do, 2000), but then we are left with the issue of determining what to do with the missing values. While there are methods to impute missing data or handle missing values, such as by using common values in the second layer of the network (Sharpe and Solly, 1995; Śmieja et al., 2018), it is ideal to not have to deal with outliers in the first place.

As a result, we may not even *want* to consider response times in our model estimates. While this does throw away some potential information, it avoids the contaminating effects of response times that systematically depart from the assumptions of the model. To handle this contingency, we created a version of the model where response times are omitted. The data generation procedure used to create the training, validation, and test sets was the same as the ones we used for the full 5-parameter model outlined above. However, this time we only gave the neural network access to pricing data and only trained it to recover the four parameters related to price, omitting non-decision time.

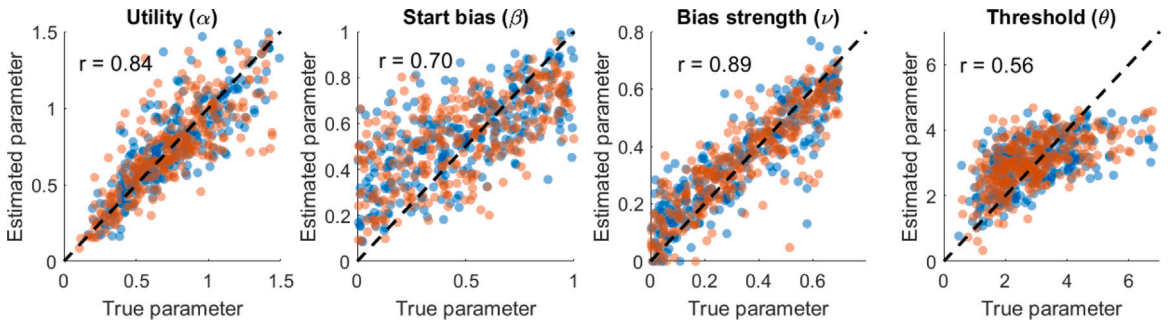


Fig. 9. True (x) versus network-estimated (y) parameter values for the price accumulation model. Blue points correspond to values that were in the training set, while orange dots correspond to an out-of-sample test set. For visual clarity, we only plot 300 random points out of the 50,000 from each set. Pearson linear correlation values (r) are provided for the relationship between true and estimated values for the out-of-sample test set. The diagonal dashed black line corresponds to perfect recovery of the true parameters. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

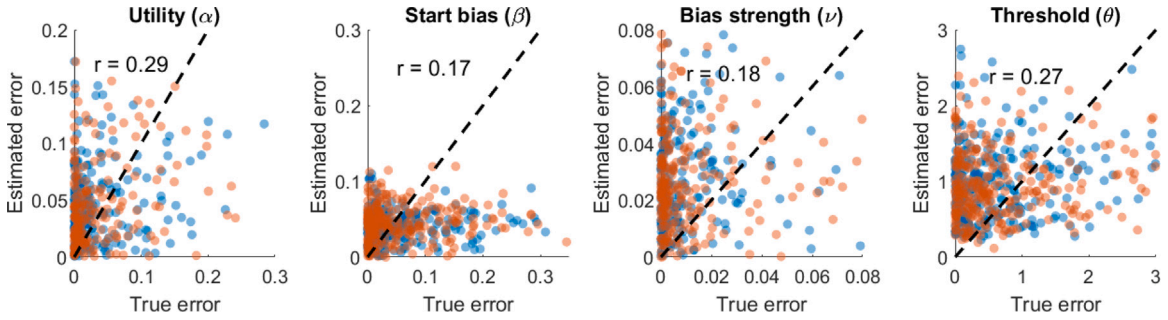


Fig. 10. True (x) and estimated (y) posterior error of the parameter estimates generated by the estimation network. Each panel shows the ability of a second error-estimation network to approximate the degree of error of the first estimation network (Fig. 9). As before, blue dots correspond to within-sample predictions from the training set, while orange dots correspond to out-of-sample predictions on which the network was not trained. The dashed black line corresponds to perfect estimation of the squared error between true and estimated values. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The network structure looked identical to the one we used for the full model, with the exception of there being 26 inputs (20 price responses, 1 mean, and 5 quantiles) and 4 outputs (utility/α, start bias/β, bias strength/ν, and threshold/θ). The hidden layers had 100, 50, 25, and then 4 nodes, where the last layer was a regression layer. Again, we trained it using root mean square propagation, using 5000 epochs, validation every 100 epochs, a default learning rate of .001, and L2 regularization of 10^{-8}. The low L2 regularization was to allow for relatively greater flexibility in parameter estimates, as we had more issues with under-fitting than over-fitting during training.

The results of the trained network are shown in Fig. 9. As with the network trained on response time data, it showed generally good recovery of the true model parameters. Its ability to estimate utility, start bias, and bias strength are almost identical to the original network. However, due to the lack of response times, its ability to estimate the threshold is significantly lower at only $r = .56$ (compared to $r = .92$ for the model including RTs). This is not too surprising, as the effect of threshold is mainly to determine stopping time with a more minor effect on the degree to which participants approach expected utility. It was possible that we would have been able to estimate thresholds at all, but it appears that even with just price data, we can get some insight into how strict or careful participants are being with their responses.

As before, we can also train a second network to estimate the error/posterior variance of the first network. We gave this second network the original data along with the estimates generated by the first network as inputs (for a total of 30 inputs), and trained it to estimate the squared error between the first network's estimates and the true generative parameters (4 outputs). The result is shown in Fig. 10. As before, there was not too much error to work with is many cases, meaning that parameters like bias strength had relatively low correlations with the true error in the model. However, this network still gives some insight into the degree of posterior variance in the estimate of the first network, making it potentially useful for assessing the goodness of the first network's performance for a particular data set without knowing the true generative parameters.

Overall, removing response times from the model and training procedure seems to have relatively minimal effects on the performance of our estimation approach. Removing non-decision time altogether, and slightly worse recovery of the threshold parameter, are relatively mild consequences for removing half of the data and all of the dynamic information accompanying price

responses. We therefore expect that the RT-free parameter estimation network can be a valuable tool for researchers assessing risk preferences in paradigms where response times cannot be reliably recorded, such as online surveys.

References

Brown, S.D., Marley, A.A.J., Donkin, C., Heathcote, A., 2008. An integrated model of choices and response times in absolute identification. Psychol. Rev. 115 (2), 396–425. http://dx.doi.org/10.1037/0033-295X.115.2.396.

Busemeyer, J.R., Townsend, J.T., 1993. Decision field theory: a dynamic-cognitive approach to decision making in an uncertain environment. Psychol. Rev. 100 (3), 432–459.

Corbin, R., Marley, A., 1974. Random utility models with equality: An apparent, but not actual, generalization of random utility models. J. Math. Psych. 11 (3), 274–293.

Cranmer, K., Brehmer, J., Louppe, G., 2020. The frontier of simulation-based inference. Proc. Natl. Acad. Sci. 117 (48), 30055–30062.

Cybenko, G.V., 1989. Approximation by superpositions of a sigmoidal function. Math. Control Signals Systems 2, 303–314.

Dai, J., Busemeyer, J.R., 2014. A probabilistic, dynamic, and attribute-wise model of intertemporal choice. J. Exp. Psychol. [Gen.] 143 (4), 1489.

Dehaene, S., 2003. The neural basis of the Weber–Fechner law: a logarithmic mental number line. Trends in Cognitive Sciences 7 (4), 145–147.

Elsemüller, L., Schnuerch, M., Bürkner, P.-C., Radev, S.T., 2023. A deep learning method for comparing bayesianhierarchical models. arXiv preprint arXiv: 2301.11873.

Evans, N.J., 2019. A method, framework, and tutorial for efficiently simulating models of decision-making. Behav. Res. Methods 51, 2390–2404.

Fengler, A., Govindarajan, L.N., Chen, T., Frank, M.J., 2021. Likelihood approximation networks (LANs) for fast inference of simulation models in cognitive neuroscience. Elife 10, e65074.

Georgescu-Roegen, N., 1958. Threshold in choice and the theory of demand. Econometrica 157–168.

Goldstein, W.M., Einhorn, H.J., 1987. Expression theory and the preference reversal phenomena. Psychol. Rev. 94 (2), 236.

Gonzalez-Vallejo, C., 2002. Making trade-offs: a probabilistic and context-sensitive model of choice behavior. Psychol. Rev. 109 (1), 137.

Greenwald, A.G., McGhee, D.E., Schwartz, J.L., 1998. Measuring individual differences in implicit cognition: the implicit association test. J. Personal. Soc. Psychol. 74 (6), 1464.

Gutmann, M.U., Corander, J., 2016. Bayesian optimization for likelihood-free inference of simulator-based statistical models. J. Mach. Learn. Res. 17 (125), 1–47, Retrieved from http://jmlr.org/papers/v17/15-017.html.

Haines, N., Kvam, P.D., Irving, L.H., Smith, C.T., Beauchaine, T.P., Pitt, M.A., Ahn, W.Y., Turner, B.M., 2020. Learning from the reliability paradox: how theoretically informed generative models can advance the social, behavioral, and brain sciences. PsyArXiv, Retrieved from psyarxiv.com/xr7y3/.

Hawkins, G.E., Marley, A., Heathcote, A., Flynn, T.N., Louviere, J.J., Brown, S.D., 2014. Integrating cognitive process and descriptive models of attitudes and preferences. Cogn. Sci. 38 (4), 701–735.

Heathcote, A., Brown, S., Mewhort, D.J.K., 2002. Quantile maximum likelihood estimation of response time distributions. Psychon. Bull. Rev. 9 (2), 394–401. http://dx.doi.org/10.3758/BF03196299.

Heathcote, A., Brown, S.D., Wagenmakers, E.-J., 2015. An introduction to good practices in cognitive modeling. In: An Introduction to Model-Based Cognitive Neuroscience. Springer, pp. 25–48.

Hick, W.E., 1952. On the rate of gain of information. Q. J. Exp. Psychol. 4 (1), 11–26. http://dx.doi.org/10.1080/17470215208416600.

Holmes, W.R., 2015. A practical guide to the probability density approximation (Pda) with improved implementation and error characterization. J. Math. Psych. 68, 13–24.

Hu, T., Chen, Z., Sun, H., Bai, J., Ye, M., Cheng, G., 2018. Stein neural sampler. http://dx.doi.org/10.48550/ARXIV.1810.03545, arXiv, Retrieved from https://arxiv.org/abs/1810.03545.

JASP Team, 2020. JASP (Version 0.14.1)[Computer software]. Retrieved from https://jasp-stats.org/.

Johnson, J.G., Busemeyer, J.R., 2005. A dynamic, stochastic, computational model of preference reversal phenomena. Psychol. Rev. 112 (4), 841–861.

Kahneman, D., Tversky, A., 1979. Prospect theory: an analysis of decision under risk. Econometrica 47 (2), 263–292.

Kato, T., Hidano, N., 2007. Anchoring effects, survey conditions, and respondents' characteristics: Contingent valuation of uncertain environmental changes. J. Risk Res. 10 (6), 773–792.

Khoong, W.H., 2020. A heuristic for efficient reduction in hidden layer combinations for feedforward neural networks. In: Intelligent Computing: Proceedings of the 2020 Computing Conference, Vol. 1. Springer, pp. 208–218.

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kirby, K.N., Petry, N.M., Bickel, W.K., 1999. Heroin addicts have higher discount rates for delayed rewards than non-drug-using controls. J. Exp. Psychol. [Gen.] 128 (1), 78.

Koffarnus, M.N., Franck, C.T., Stein, J.S., Bickel, W.K., 2015. A modified exponential behavioral economic demand model to better describe consumption data. Exp. Clin. Psychopharmacol. 23 (6), 504.

Kruschke, J., 2014. Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and STAN. Academic Press.

Kühn, S., Schmiedek, F., Schott, B., Ratcliff, R., Heinze, H.-J., Düzel, E., Lindenberger, U., Lövden, M., 2011. Brain areas consistently linked to individual differences in perceptual decision-making in younger as well as older adults before and after training. J. Cogn. Neurosci. 23 (9), 2147–2158.

Kvam, P.D., 2019. A geometric framework for modeling dynamic decisions among arbitrarily many alternatives. J. Math. Psych. 91, 14–37.

Kvam, P.D., Busemeyer, J.R., 2020. A distributional and dynamic theory of pricing and preference. Psychol. Rev. 127, 1053—1078.

Kvam, P.D., Busemeyer, J.R., Pleskac, T.J., 2021. Temporal oscillations in preference strength provide evidence for an open system model of constructed preference. Sci. Rep. 11 (1), 8169.

Kvam, P.D., Marley, A.A.J., Heathcote, A., 2023. A unified theory of discrete and continuous responding. Psychol. Rev.

Kvam, P.D., Pleskac, T.J., Yu, S., Busemeyer, J.R., 2015. Interference effects of choice on confidence: quantum characteristics of evidence accumulation. Proc. Natl. Acad. Sci. 112 (34), 10645–10650. http://dx.doi.org/10.1073/pnas.1500688112.

Kvam, P.D., Turner, B.M., 2021. Reconciling similarity across models of continuous selections. Psychol. Rev. 128 (4), 766–786.

Lacouture, Y., Marley, A.A., 1995. A mapping model of bow effects in absolute identification. J. Math. Psych. 39 (4), 383–395. http://dx.doi.org/10.1006/jmps.1995.1036.

Lacouture, Y., Marley, A.A.J., 2004. Choice and response time processes in the identification and categorization of unidimensional stimuli. Percept. Psychophys. 66 (7), 1206–1226.

Levy, H., 1992. Stochastic dominance and expected utility: Survey and analysis. Manage. Sci. 38 (4), 555–593.

Lichtenstein, S., Slovic, P., 1971. Reversals of preference between bids and choices in gambling decisions. J. Exp. Psychol. 89 (1), 46–55.

Louviere, J.J., Street, D., Burgess, L., Wasi, N., Islam, T., Marley, A.A., 2008. Modeling the choices of individual decision-makers by combining efficient choice experiment designs with extra preference information. J. Choice Modell. 1 (1), 128–164.

Luce, R.D., 1986. Response Times, No. 8. Oxford University Press.

Lueckmann, J.-M., Bassetto, G., Karaletsos, T., Macke, J.H., 2019. Likelihood-free inference with emulator networks. arXiv:1805.09294.

Marley, A.A.J., 1989. A random utility family that includes many of the 'classical'models and has closed form choice probabilities and choice reaction times. Br. J. Math. Stat. Psychol. 42 (1), 13–36.
Marley, A.A.J., Colonius, H., 1992. The "horse race" random utility model for choice probabilities and reaction times, and its compering risks interpretation. J. Math. Psych. 36 (1), 1–20.
Marley, A.A.J., Cook, V.T., 1984. A fixed rehearsal capacity interpretation of limits on absolute identification performance. Br. J. Math. Stat. Psychol. 37 (2), 136–151. http://dx.doi.org/10.1111/j.2044-8317.1984.tb00797.x.
Marley, A.A.J., Louviere, J.J., 2005. Some probabilistic models of best, worst, and best–worst choices. J. Math. Psych. 49 (6), 464–480.
Marley, A.A.J., Regenwetter, M., 2016. Choice, preference, and utility: Probabilistic and deterministic representations. pp. 374–453. http://dx.doi.org/10.1017/9781139245913.008,
Molloy, M.F., Romeu, R.J., Kvam, P.D., Finn, P.R., Busemeyer, J., Turner, B.M., 2020. Hierarchies improve individual assessment of temporal discounting behavior. Decision 7, 212–224.
Murphy, K.P., 2012. Machine Learning: A Probabilistic Perspective. MIT Press.
Pleskac, T.J., Busemeyer, J.R., 2010. Two-stage dynamic signal detection: a theory of choice, decision time, and confidence. Psychol. Rev. 117 (3), 864. http://dx.doi.org/10.1037/A0019737.
Plummer, M., 2003. JAGS: a program for analysis of bayesian graphical models using gibbs sampling. In: Proceedings of the 3rd International Workshop on Distributed Statistical Computing, Vol. 124. Vienna, Austria, p. 10.
Radev, S.T., D'Alessandro, M., Mertens, U.K., Voss, A., Köthe, U., Bürkner, P.-C., 2021. Amortized Bayesian model comparison with evidential deep learning. arXiv:2004.10629.
Radev, S.T., Mertens, U.K., Voss, A., Ardizzone, L., Köthe, U., 2020a. BayesFlow: Learning complex stochastic models with invertible neural networks. arXiv:2003.06281.
Radev, S.T., Mertens, U.K., Voss, A., Köthe, U., 2020b. Towards end-to-end likelihood-free inference with convolutional neural networks. Br. J. Math. Stat. Psychol. 73 (1), 23–43.
Rahm, E., Do, H.H., 2000. Data cleaning: Problems and current approaches. IEEE Data Eng. Bull. 23 (4), 3–13.
Savage, L.J., 1954. The Foundations of Statistics. John Wiley & Sons, Inc, New York, NY.
Saxe, A.M., Bansal, Y., Dapello, J., Advani, M., Kolchinsky, A., Tracey, B.D., Cox, D.D., 2019. On the information bottleneck theory of deep learning. J. Stat. Mech. Theory Exp. 2019 (12), 124020.
Scheibehenne, B., Pachur, T., 2015. Using Bayesian hierarchical parameter estimation to assess the generalizability of cognitive models of choice. Psychon. Bull. Rev. 22 (2), 391–407.
Scheibehenne, B., Rieskamp, J., González-Vallejo, C., 2009. Cognitive models of choice: Comparing decision field theory to the proportional difference model. Cogn. Sci. 33 (5), 911–939.
Sharpe, P.K., Solly, R., 1995. Dealing with missing values in neural network-based diagnostic systems. Neural Comput. Appl. 3 (2), 73–77.
Sherif, M., Taub, D., Hovland, C.I., 1958. Assimilation and contrast effects of anchoring stimuli on judgments. J. Exp. Psychol. 55 (2), 150.
Slovic, P., Lichtenstein, S., 1983. Preference reversals: a broader perspective. Amer. Econ. Rev. 73 (4), 596–605.
Śmieja, M., Struski, Ł., Tabor, J., Zieliński, B., Spurek, P., 2018. Processing of missing data by neural networks. Adv. Neural Inf. Process. Syst. 31.
Starns, J.J., Ratcliff, R., 2010. The effects of aging on the speed–accuracy compromise: Boundary optimality in the diffusion model. Psychol. Aging 25 (2), 377.
Swait, J., Marley, A.A., 2013. Probabilistic choice (models) as a result of balancing multiple goals. J. Math. Psych. 57 (1–2), 1–14.
Turner, B.M., Schley, D.R., 2016. The anchor integration model: A descriptive model of anchoring effects. Cogn. Psychol. 90, 1–47.
Turner, B.M., Sederberg, P.B., 2012. Approximate bayesian computation with differential evolution. J. Math. Psych. 56 (5), 375–385.
Turner, B.M., Sederberg, P.B., 2014. A generalized, likelihood-free method for posterior estimation. Psychon. Bull. Rev. 21 (2), 227–250.
Turner, B.M., Zandt, T.V., 2018. Approximating bayesian inference through model simulation. Trends in Cognitive Sciences 22 (9), 826–840. http://dx.doi.org/10.1016/j.tics.2018.06.003, Retrieved from https://www.sciencedirect.com/science/article/pii/S1364661318301438.
Tversky, A., Kahneman, D., 1992. Advances in prospect theory: cumulative representation of uncertainty. J. Risk Uncertain. 5 (4), 297–323.
Tversky, A., Slovic, P., Kahneman, D., 1990. The causes of preference reversal. Amer. Econ. Rev. 204–217.
Wagenmakers, E.J., Lodewyckx, T., Kuriyal, H., Grasman, R., 2010. Bayesian hypothesis testing for psychologists: a tutorial on the savage-dickey method. Cogn. Psychol. 60 (3), 158–189. http://dx.doi.org/10.1016/j.cogpsych.2009.12.001.
Walczak, S., Cerpa, N., 1999. Heuristic principles for the design of artificial neural networks. Inf. Softw. Technol. 41 (2), 107–117.
Yu, S., Pleskac, T.J., Zeigenfuse, M.D., 2015. Dynamics of postdecisional processing of confidence. J. Exp. Psychol. [Gen.] 144 (2), 489.
Zhou, D.-X., 2020. Universality of deep convolutional neural networks. Appl. Comput. Harmon. Anal. 48 (2), 787–794.