Journal of Economic Behavior and Organization 236 (2025) 107054

Contents lists available at ScienceDirect

Journal of Economic Behavior and Organization

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

Research paper

Logit neural-network utility☆

Sung-Lin Hsieh [a], Shaowei Ke [b], Zhaoran Wang [c], Chen Zhao [d,*]

[a] Independent researcher, Taipei, Taiwan
[b] Department of Economics, China Europe International Business School, China
[c] Department of Industrial Engineering and Management Sciences, Northwestern University, United States of America
[d] Faculty of Business and Economics, University of Hong Kong, China

ARTICLE INFO

Keywords:
Neural network
Stochastic choice
Logit Choice Model

ABSTRACT

We introduce stochastic choice models that feature neural networks, one of which is called the logit neural-network utility (NU) model. We show how to use simple neurons, referred to as behavioral neurons, to capture behavioral effects, such as the certainty effect and reference dependence. We find that simple logit NU models with natural interpretation provide better out-of-sample predictions than expected utility theory and cumulative prospect theory, especially for choice problems that involve lotteries with both positive and negative prizes. We also find that the use of behavioral neurons mitigates overfitting and significantly improves our models' performance, consistent with numerous successes in introducing useful inductive biases in the machine-learning literature.

1. Introduction

Over the last decade, machine-learning models have demonstrated strong predictive power in many decision problems. For example, when we shop at Amazon, Amazon recommends products to us with the help of machine learning. For a machine-learning model to perform well in this regard, the model must make good predictions about the likelihood that a consumer will buy a product after it is recommended.

The fact that a machine-learning model predicts people's behavior well, however, does not necessarily make it a good model of decision-making. For example, suppose there is a true model that describes how a decision maker behaves. From the model, one may gain insights about the decision process. A machine-learning model may approximate the true model in some decision problems well—and therefore predict well for those problems—but the insights from the true model may be lost in the approximation.

Nonetheless, it is possible that of the numerous machine-learning models with the potential to predict well, some are indeed good models of how people make decisions. In a recent paper by Ke and Zhao (2024), the authors show that a decision model that features neural networks called the *neural-network utility* (NU) model can be a good model of how people make choices: The model is characterized by simple axioms motivated by empirical evidence imposed on people's choice behavior, and yields an intuitive and plausible interpretation of how people make choices.

The primitive of Ke and Zhao (2024), however, is a decision maker's preference. In other words, their model is deterministic. It is well known that stochastic choice models are more suitable for empirical analyses than deterministic ones. Hence, in this paper

☆ We are grateful to Tilman Börgers, Yan Chen, In-Koo Cho, David Dillenberger, Andrew Ellis, Wayne Gao, Faruk Gul, Paulo Natenzon, Pietro Ortoleva, Wolfgang Pesendorfer, and seminar participants at numerous seminars and conferences for helpful comments. Ke thanks the University of Pennsylvania and the Cowles Foundation at Yale University for hospitality and support when some of the work on this paper was undertaken, and gratefully acknowledges the financial support of the Michigan Institute for Teaching and Research in Economics (MITRE).

* Corresponding author.
E-mail addresses: slhsieh@umich.edu (S.-L. Hsieh), shaoweike@ceibs.edu (S. Ke), zhaoran.wang@northwestern.edu (Z. Wang), czhao@hku.hk (C. Zhao).

https://doi.org/10.1016/j.jebo.2025.107054
Received 6 February 2025; Received in revised form 2 May 2025; Accepted 7 May 2025

Available online 6 June 2025

we introduce and axiomatically characterize stochastic-choice versions of the NU model. One of them is a logit version of the NU model, which we use for empirical analyses. We show empirically that some simple logit NU models that are easy to interpret perform better than expected utility theory and cumulative prospect theory out of sample.

Specifically, consider a convex set of choice alternatives X in $\mathbb{R}^N$. For example, an alternative may be a product described by its N attributes or may be a lottery that specifies the probability that the decision maker will receive each of the N prizes. To explain our models and findings, we first describe the NU model. In an NU model, the decision maker takes an alternative as the input of a feedforward neural network and derives the utility of the alternative as the output of the network. A feedforward neural network may have multiple hidden layers, and each hidden layer may have multiple neurons. A neuron consists of two operations. First, it applies an affine aggregation to its child neurons' values.[1]. Second, it compares the outcome of aggregation to a normalized threshold 0, which determines whether a neuron is activated.

In our empirical analysis, X is the probability simplex and choice alternatives are lotteries. In that case, the interpretation of the NU model is as follows. Note that an affine function on the probability simplex is equivalent to an expected utility function. Therefore, in an NU model, it is as if that the decision maker first considers multiple expected utility functions (the affine functions of the first hidden layer's neurons) plausible, and uses the activated ones to evaluate a lottery. Next, the decision maker aggregates the multiple risk attitudes. Again, she may not have a unique way to aggregate them. This is captured by the second hidden layer. The aggregation continues recursively until the decision maker reaches the last hidden layer and forms the final evaluation of the lottery.

The primitive of our theory is a *stochastic choice function* that specifies the choice frequency of each alternative in an arbitrary set of alternatives. Let $\rho(x, A)$ denote the choice frequency of alternative x when the available set of alternatives is A. We introduce and characterize two stochastic-choice versions of the NU model. The Luce NU model has an NU function U and a strictly increasing and positive-valued function ϕ such that

$$\rho(x, A) = \frac{\phi(U(x))}{\sum_{y\in A} \phi(U(y))}.$$

The logit NU model uses the familiar logit formula:

$$\rho(x, A) = \text{Prob}\left[U(x) + \varepsilon_x \geqslant \max_{y\in A}\ U(y) + \varepsilon_y\right] = \frac{e^{U(x)}}{\sum_{y\in A} e^{U(y)}},$$

in which ε_y's are independent and identically distributed random variables that follow the Gumbel distribution (type-I generalized extreme value distribution) with scale parameter 1. Clearly, the logit NU model is a special case of the Luce NU model.

The Luce NU model and the logit NU model both contain an NU function. We show how to construct simple neural-network structures in an NU function to capture well-known behavioral phenomena such as the certainty effect, reference dependence, etc. We call these structures *behavioral neurons*. They will play an important role in our empirical analyses.

We study the empirical performance of the logit NU model, examine the complexity required for a neural network to explain and predict people's choice behavior well, and identify the choice problems in which the logit NU model outperforms benchmark models. The answer to the second question is useful because, arguably, if a rather complex neural network is required to explain and predict the data well, the interpretation the logit NU model offers might be too complex to be interesting or insightful.

We analyze the logit NU model using the training and testing datasets provided by the Choice Prediction Competition 2018 (see Plonsky et al. (2019)). After aggregating the individual choice data, each data point consists of a description of two lotteries (see Fig. 6) and the proportion of experiment participants who choose the first lottery over the second. We use the training dataset to estimate a model and then measure the model's out-of-sample performance by computing its mean square error in the testing dataset (testing error).

To do so, we begin by taking expected utility theory and cumulative prospect theory (see Tversky and Kahneman (1992)) as the benchmark. We must parameterize these models to avoid *overfitting*. For example, rather than estimating a general Bernoulli index for expected utility, which requires a large amount of data, we may estimate the constant-absolute-risk-aversion (CARA) special case. The same applies to cumulative prospect theory. Our first observation is that, consistent with recent empirical findings, expected utility theory performs well: Cumulative prospect theory under standard parameterization cannot outperform CARA expected utility in out-of-sample prediction.[2]

Similarly, we need to parameterize the NU function in the logit NU model to mitigate overfitting. We combine two ideas to achieve this. First, note that the affine aggregation functions of the first-hidden-layer neurons of the NU function are defined on the probability simplex. An affine function on the probability simplex is equivalent to an expected utility function. Therefore, a natural idea is to require that those functions be CARA expected utility functions.[3] Second, recall that we can use behavioral neurons to capture well-documented behavioral effects. These behavioral neurons are parameterized and may provide useful flexibility for the NU function. Therefore, to parameterize the NU function, we require that its first hidden layer consist of at most the following three

[1] For a neuron in the first hidden layer, its child neurons are all components of the alternative. For a neuron in the nth hidden layer ($n > 1$), its child neurons are all neurons in the $(n-1)$th hidden layer

[2] See, among others, Harless and Camerer (1994); Blavatskyy et al. (2022); Bouchouicha et al. (2023); and Dembo et al. (2024).

[3] If we only utilize this idea for parameterization, we will remove too much flexibility from the NU function: The model's performance is essentially identical to the CARA expected utility benchmark.

types of neurons: (i) neurons that evaluate CARA expected utility, (ii) neurons that capture the certainty effect, and (iii) neurons that capture reference dependence. Then, additional standard hidden layers may be concatenated with this first hidden layer.

We consider different configurations of the network structure and find that a two-hidden-layer logit NU model with all three types of neurons used in the first layer has the lowest testing error. In addition, the testing error of this logit NU model is lower than that of the CARA expected utility model. Hence, a reasonably complex NU function that has a natural interpretation has the best performance.

Moreover, the two-hidden-layer logit NU model's estimated parameters reveal some interesting characteristics. First, most neurons that capture reference dependence have reference points close to zero in absolute value. As a result, for pure-gain or pure-loss lotteries, the model is essentially expected utility with a boost for higher probabilities of winning a prize (due to neurons that capture the certainty effect). Second, indifference curves exhibit more complex patterns for mixed lotteries—i.e., lotteries with both gains and losses, which potentially reflects the difficulty of evaluating such lotteries for the decision maker. Third, when evaluating a mixed lottery, the ratio of the probability of gaining over the probability of losing seems to be a key factor for making choices.

Remarkably, we find that the logit NU model performs well in choice problems with mixed lotteries, but the CARA expected utility model does not. In fact, the dominance of the logit NU model over the CARA expected utility model is mainly driven by their performances in those choice problems. Most of the recent empirical studies that support the expected utility model have primarily focused on pure-gain lotteries.[4] Therefore, our findings imply that non-expected utility models may be more effective in dealing with mixed lotteries.

Overall, the use of behavioral neurons improves the logit NU model's performance significantly, which shows that economists' domain knowledge in decision-making is useful even for predictions using powerful machine-learning methods. This observation echoes numerous successful examples in deep learning. In deep learning, there is a recurring theme called the inductive bias. This refers to a set of assumptions on the structure of neural networks or training algorithms that guide the learning process and incorporate prior knowledge of the true model so that deep learning can be more efficiently implemented in a more restricted subspace (see Neyshabur et al. (2014); Goyal and Bengio (2022); and Goldblum et al. (2023)). Inductive biases may allow neural-network models to learn faster and generalize better out of sample. Our approach introduces behavioral neurons as the inductive bias to neural networks. Compared with general neural networks, our inductive bias restricts the search space to the model class that better captures human behavior. Consequently, we achieve significantly better generalization performance.

As mentioned earlier, our paper provides an axiomatic characterization of the Luce and logit NU models. These axioms are useful in two ways. First, while predictive accuracy matters to us, maximizing predictive accuracy is not our primary objective. Rather, we seek predictive accuracy within a coherent economic framework. Without this restriction, one might freely choose among various machine learning methods and simply select the model with the highest predictive accuracy.[5] In this sense, the axioms define the economic framework that constrains our empirical analysis: We rule out models inconsistent with our axioms. Second, the axiomatic characterization clarifies how our model generalizes or differs from existing economic models, and provides a foundation for future research in this interdisciplinary area. Hence, the axiomatic characterization is not independent of our empirical analysis, but an essential component of its foundation.

1.1. Related literature

The Luce NU model and logit NU model are stochastic choice models. Many other axiomatic stochastic choice models that are related to the Luce rule or the logit model have been studied. Kovach and Tserenjigmid (2022a) characterize an axiomatic model called nested stochastic choice, as well as one of its special cases, the nested logit model, which is a well-known solution to the red-bus-blue-bus problem (see Debreu (1960)). Kovach and Tserenjigmid (2022b) introduce a generalization of the Luce rule in which the decision maker divides the available alternatives into two groups; one group is focal and more likely to be chosen. Saito (2018) characterizes another generalization of the logit model that is widely used in empirical studies—the mixed logit model—and Lu and Saito (2022) analyze the differences between the mixed logit model and pure characteristic models. Echenique and Saito (2019) extend the Luce rule to deal with zero-probability choices. Fudenberg and Strzalecki (2015) study a dynamic extension of the logit model with aversion to large menus.

A growing literature combines economic theory with machine learning. Similar to our work, Plonsky et al. (2017), Plonsky et al. (2019), and Peterson et al. (2021) also combine behavioral economics with machine learning methods to study choice behavior. Their approach is closer to machine learning. Their models do not have axiomatic foundations and may violate, for example, basic behavioral properties such as transitivity, which is different from our approach. Peterson et al. (2021) measure the performance of their models using cross-validation errors, while the other two papers and ours measure out-of-sample prediction errors. Fudenberg and Liang (2019) combine game theory with machine learning and use the decision tree algorithm to study the initial play of games. Cho and Libgober (2021) study a problem in which an agent uses historical data and algorithms to provide action recommendations to a sequence of players in order to maximize their average long-run payoffs. Caplin et al. (2022) and Ke et al. (2024) analyze how to model machine learning and how to model people learning from complex machine-learning algorithms, respectively.

[4] See Blavatskyy et al. (2022); Bouchouicha et al. (2023); Dembo et al. (2024); and McGranaghan et al. (2024).

[5] See the related discussion on Plonsky et al. (2017) and Plonsky et al. (2019) in Section 6.

Our empirical analysis is related to the empirical analysis of non-expected utility theory, such as Chew and Waller (1986); Battalio et al. (1990); Harless and Camerer (1994); Starmer (2000); Wu et al. (2005); Choi et al. (2007); Bernheim and Sprenger (2020); Blavatskyy et al. (2022); Bouchouicha et al. (2023); Dembo et al. (2024); and McGranaghan et al. (2024). Among these, Harless and Camerer (1994) show that in some cases expected utility theory has the best performance, which is consistent with one of our findings. Different from our analysis, they do not examine models' predictive power using a testing dataset that is not accessed when estimating models. Bernheim and Sprenger (2020) find that decision weights are not sensitive to the ranks of outcomes. Our finding is consistent with theirs—when we estimate the probability-weighting function of cumulative prospect theory, we find little distortion of probabilities. Also consistent with our finding, Blavatskyy et al. (2022), Bouchouicha et al. (2023), and Dembo et al. (2024) find that expected utility theory performs well empirically.

Our paper is also related to the machine-learning literature that focuses on interpreting machine-learning models, especially neural-network models; see Murdoch et al. (2019) for a survey. Our paper provides an axiomatic foundation and interpretation for our neural-network model. In addition, our use of behavioral neurons is related to introducing inductive biases to neural networks. For example, see Neyshabur et al. (2014); Goyal and Bengio (2022); and Goldblum et al. (2023).

The rest of the paper is organized as follows. Sections 2 and 3 introduce the Luce NU model, the logit NU model, and behavioral neurons. Section 4 takes the logit NU model to data. Section 5 provides an axiomatic characterization for our models, and Section 6 concludes.

2. Setup and models

Let $X \subseteq \mathbb{R}^N$ be nonempty convex and compact and have nonempty interior. An element $x = (x_1, \dots, x_N)$ of X represents a choice alternative. One may think of X as a space of products that are described by their attributes, in which case x_i is the value of product x's ith attribute. One can also let X be the probability simplex in $\mathbb{R}^N$, and think of it as the set of all probability measures over N prizes. In this case, we call $x \in X$ a lottery. We use w, x, y, z to denote generic elements of X, and for any $\lambda \in [0,1]$, we use λxy to denote the convex combination $\lambda x + (1-\lambda)y$. Let $\mathcal{M}$ denote the collection of nonempty finite subsets of X. Elements of $\mathcal{M}$, denoted by A, B, C, are called menus.

The primitive of our model is a *stochastic choice function* (SCF) that describes for any menu the choice probability of each alternative in the menu.

Definition 1. A function $\rho : X \times \mathcal{M} \to [0,1]$ is an SCF if for any $A \in \mathcal{M}$, $\sum_{x \in A} \rho(x, A) = 1$.

When the menu is A, the choice probability of $x \in A$ is $\rho(x, A)$. As usual, there are two ways to interpret an SCF. The group interpretation, which is more relevant for our empirical study, says that for a fixed group of homogeneous decision makers, the SCF specifies the choice frequency distribution for any menu when all decision makers in that group choose from the menu independently.[6]

We introduce and later will characterize two representations of the decision maker's SCF. One of them, the logit NU model, will be used in the empirical analysis. We first define the NU function. Given any vector-valued function τ, we use $\tau^{(j)}$ to denote the jth component of τ.

Definition 2. A function $U : X \to \mathbb{R}$ is an NU function if there exist
(i) $h, w_0, \dots, w_{h+1} \in \mathbb{N}$ with $w_0 = n$ and $w_{h+1} = 1$, and
(ii) affine functions $\tau_i : \mathbb{R}^{w_{i-1}} \to \mathbb{R}^{w_i}$, $i = 1, \dots, h+1$, such that for any $x \in X$,

$$U(x) = \tau_{h+1} \circ \theta \circ \tau_h \circ \dots \circ \theta \circ \tau_2 \circ \theta \circ \tau_1(x), \tag{1}$$

in which θ is an entry-wise operation such that for any $w \in \mathbb{N}$ and $b \in \mathbb{R}^w$, we have $\theta(b) = (\max\{b_i, 0\}, \dots, \max\{b_w, 0\})$.

In Definition 2, $\theta \circ \tau_i$ is called the ith *hidden layer*, and $(\theta \circ \tau_i)^{(j)} = \max\{\tau_i^{(j)}(\cdot), 0\}$ is called a *neuron*.[7] Hence, Eq. (1) characterizes a network of neurons with h hidden layers, and the ith hidden layer has w_i neurons. Fig. 1 provides an example of an NU function.

Suppose X is the probability simplex. The economic interpretation of the NU function is as follows. Recall that an affine function defined on the probability simplex must be an expected utility function and vice versa. Therefore, if the decision maker's preference is represented by an NU function, it is as if she first considers multiple ways to evaluate the uncertainty of a lottery, which corresponds to the affine functions of the first hidden layer. For instance, she may have one neuron that activates when the expected value of prizes is high and another that activates whenever the downside risk is high.

With multiple risk attitudes, the decision maker wants to aggregate them. She may also consider multiple ways to do so. This is captured by the second hidden layer, with each affine function of the second-hidden-layer neuron representing one way of aggregation if activated. The aggregation continues recursively until the decision maker reaches the final evaluation of the lottery.[8]

Finally, we introduce two representations of the SCF.

[6] In the other interpretation, fixing one decision maker, the SCF specifies for any menu the decision maker's ex ante choice probability of each alternative in that menu.

[7] The function θ is called the *activation function*. It may take other functional forms in general, but the form we assume in Definition 2, known as the rectified linear unit, is considered to be the most popular and to have strong biological motivations (see Hahnloser et al. (2000) and LeCun et al. (2015), among others).

[8] The interpretation of the NU function when X is, for example, a space of products characterized by their attributes can be found in Ke and Zhao (2024). The probability simplex case is more relevant in our analysis.

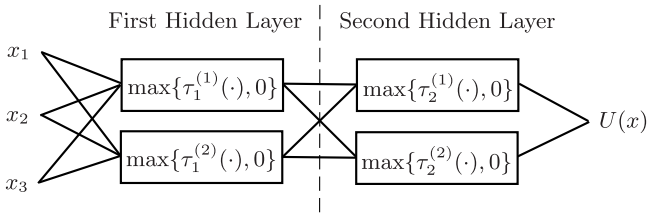


Fig. 1. This NU function evaluates an alternative $x = (x_1, x_2, x_3)$ through a neural network that has two hidden layers, with each layer having two neurons. Each affine $\tau_1^{(j)}$ is from the choice domain (a subset of $\mathbb{R}^3$) to $\mathbb{R}$, and each affine $\tau_2^{(j)}$ is from $\mathbb{R}^2$ to $\mathbb{R}$. Neurons in the first layer are called *child neurons* of neurons in the second layer. Neurons in the second layer are called *parent neurons* of neurons in the first layer.

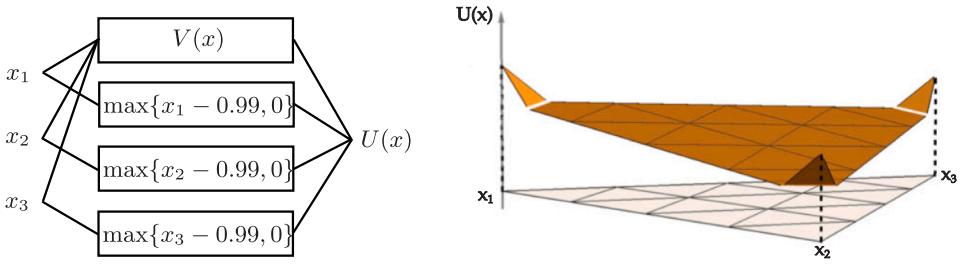


Fig. 2. In the first neuron, V is an expected utility function, which is affine. If the probability of receiving a prize is larger than 0.99 ($x_i > 0.99$ for some $i \in \{1, 2, 3\}$), one of the three other neurons that capture the bias toward certainty will be activated. Finally, $U(x)$ is equal to a weighted sum of all neurons' values.

Definition 3. The SCF has a Luce NU representation if there exists an NU function $U : X \to \mathbb{R}$ and a strictly increasing continuous function $\phi : U(X) \to \mathbb{R}_{++}$ such that for any $A \in \mathcal{M}$ and any $x \in A$, $\rho(x, A) = \frac{\phi(U(x))}{\sum_{y \in A} \phi(U(y))}$.

If a random variable ε follows the Gumbel distribution (the type-I generalized extreme value distribution) with scale parameter 1, we write $\varepsilon \sim \text{GEV}_\text{I}(1)$.

Definition 4. The SCF has a logit NU representation if there exists an NU function $U : X \to \mathbb{R}$ such that for any $A \in \mathcal{M}$ and any $x \in A$,

$$\rho(x, A) = \text{Prob}\left[U(x) + \varepsilon_x \geqslant \max_{y \in A} U(y) + \varepsilon_y\right] = \frac{e^{U(x)}}{\sum_{y \in A} e^{U(y)}},$$

in which $\varepsilon_y \overset{\text{iid}}{\sim} \text{GEV}_\text{I}(1)$.

3. Behavioral neurons in NU functions

From here on, we focus on the case in which X is the probability simplex. We first introduce a few examples to show how we can construct neurons in an NU function to capture well-known empirical findings. Such construction will be useful in our empirical analysis of the logit NU model.

3.1. Behavioral neurons of certainty effects

The first example appeared in Ke and Zhao (2024). Suppose there are three prizes and X is the set of all lotteries for these prizes. From the Allais paradox, we know that decision makers are often biased toward certainty. Fig. 2 presents an NU function in which the first neuron captures standard expected utility evaluation, while the other three neurons capture the bias toward certainty for the three prizes respectively.[9] We call the neurons that capture the bias toward certainty *certainty-effect neurons*.

The first neuron in Fig. 2 does not compare the outcome of its aggregation to zero, as required for an NU function. This is for simplicity and without loss of generality. Let $\underline{V} = \min_{x \in X} V(x)$. If $\underline{V} \geqslant 0$, $V(x) = \max\{V(x), 0\}$. Otherwise, we replace the first neuron with $\max\{V(x) - \underline{V}, 0\} = V(x) - \underline{V}$. Then, at its parent neurons, we add $\underline{V}$ back to the affine aggregation.

[9] This function appears in Chapter 2.4.4.2 of Schmidt (1998), although its connection to neural-network models is not explored. We thank David Dillenberger for pointing this out.

3.2. Behavioral neurons of reference dependence

The second example is related to reference dependence. Suppose there are n monetary prizes denoted by $\{\pi_1, \dots, \pi_n\} \subseteq \mathbb{R}$ and X is the set of all lotteries on those prizes. As pointed out by Kahneman and Tversky (1979) and many other papers, prizes are often evaluated relative to a reference point, and people treat gains and losses (i.e., prizes better than and worse than the reference point, respectively) differently. It is also documented that the difference disappears when prizes do not deviate much from the reference point (see, for example, Ert and Erev (2013)).

We can use an NU function to capture these ideas in a natural way. Let there be two neurons in the first hidden layer. The first neuron, $V(x) = \sum_i x_i v(x_i)$, again computes the expected utility of x, with v being the Bernoulli index. The second neuron captures loss aversion relative to the reference point $\gamma \in \mathbb{R}$ with a threshold $\varepsilon > 0$:

$$V_l(x) = \max\left\{\sum_{i=1}^{n} x_i \max\{\gamma - \pi_i, 0\} - \varepsilon, 0\right\}.$$

We call such neurons *reference-dependence neurons*. Note that $\sum_{i=1}^{n} x_i \max\{\gamma - \pi_i, 0\} - \varepsilon$ is an affine function of x, and the loss of prize π_i is given by $\max\{\gamma - \pi_i, 0\}$. The neuron V_l is activated if and only if the expected loss is larger than ε. Finally, $U(x) = V(x) - \lambda V_l(x)$, with $\lambda > 0$ being the loss-aversion coefficient. Clearly, in this NU function, loss aversion only occurs when prizes deviate from the reference point significantly.

4. Empirical analysis of the logit NU model

In this section, we show how one can estimate the logit NU model and how it performs empirically; that is, how well it explains and predicts people's choice behavior out of sample. We will examine the complexity required for a neural network to explain and predict well, and identify the choice problems in which the logit NU model outperforms the benchmark. About the former, we might believe that—even as an as-if model—the decision maker's logit NU function should not be too complex. A simple logit NU function is easier to interpret and may provide more useful insights. We will examine whether this conjecture is correct.

4.1. Data description and training models

We use the training and testing datasets provided by the aggregate-behavior track of the Choice Prediction Competition 2018 (see Plonsky et al. (2019)).[10] The datasets come from several experiments conducted at the Hebrew University of Jerusalem and Technion–Israel Institute of Technology. Each participant in the experiments faces 750 binary choice problems over lotteries, in which a lottery is instantiated by the description of a probability vector defined over its support, a nonempty finite set of monetary prizes (see the horizontal axis of Fig. 3 for the set of all possible monetary prizes in these experiments).[11] In each binary choice problem, a participant must choose one lottery of the two.[12]

The 750 binary choice problems each participant faces consist of 30 different problems (with each problem characterized by the pair of lotteries that the participant faces) presented in a random order, and given the random order of the 30 different problems, each of the 30 different problems is repeated 25 times consecutively. For example, a participant may face the choice between lotteries w and x for 25 times consecutively, then face the choice between lotteries y and z for 25 times, and so on. Different participants may face different binary choice problems.

In total, there are 270 different binary choice problems that any of the participants may face in the experiment. Henceforth, when we say a binary choice problem, we mean one of the 270 different ones. Of these, 210 are in the training dataset and 60 in the testing dataset. Moreover, 30 binary choice problems are designed to replicate 14 well-known behavioral phenomena, including the certainty effect, the reflection effect, overweighting of small probabilities, etc.[13] They are in the training dataset. The other binary choice problems are generated somewhat randomly (see Erev et al. (2017) for more details). Crucially, we take as given the division of the training data and the testing data in Plonsky et al. (2019)—we have no control over how the dataset is divided.

Some information from the original datasets is discarded. First, our analysis does not make use of the demographic information. We could have first divided the dataset by participants' demographic information and then estimated a model for each type of participants. However, as will be explained in Footnote , this turns out to worsen the overfitting problem at least for the benchmark cases. Second, recall that each binary choice problem is repeated 25 times for each participant. After the first 5 repetitions, a participant can observe the realization of the lotteries from previous repetitions. Our theory has little to offer about how choice behavior will be affected by feedback. Therefore, in our empirical analysis, we do not use this information. We treat all repetitions of the same binary choice problem equivalently. Finally, our theory cannot deal with binary choice problems that involve ambiguity (lotteries' probabilities are not specified) and have little to say about binary choice problems in which realizations of lotteries are correlated. We exclude those binary choice problems (41 in the training dataset and 15 in the testing dataset) from our analysis. Eventually, the training dataset contains 169 data points and the testing dataset contains 45 data points.

[10] The datasets are publicly available at https://cpc-18.com.

[11] Since the total number of prizes that show up in the datasets is finite, we continue using our notations from Section 2 for lotteries and prizes.

[12] See Fig. 6 in Appendix A.

[13] The behavioral biases are successfully replicated, but the magnitude is smaller than in the original studies that document the biases. See Erev et al. (2017) for more details.

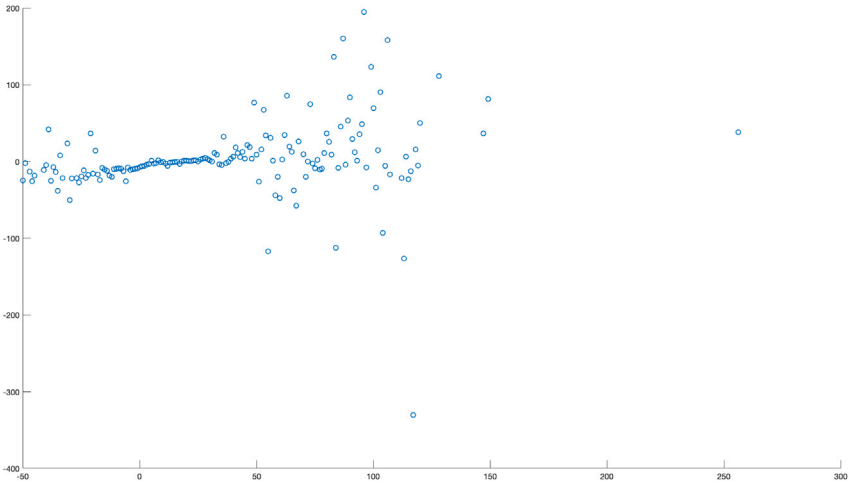


Fig. 3. The horizontal axis represents the prizes. The vertical axis corresponds to the value of the estimated Bernoulli index for the expected utility model. Note that only finitely many prizes are in the support of the lotteries involved in the experiments. Therefore, we see only dots in the plot of the Bernoulli index.

We aggregate individual choice data for each of the 214 (169+45) binary choice problems; that is, for each binary choice problem, we calculate the fraction of participants who choose each lottery. We call these fractions *choice probabilities*. A data point contains information about the two lotteries (as the covariate) and the choice probabilities (as the response).

For each model we consider, we estimate/train the model using the training dataset, and then evaluate its performance on the testing dataset. Following Plonsky et al. (2019), we use the mean squared error (MSE) of predicted choice probabilities (compared with actual choice probabilities) as the metric to evaluate the performance of the model.

We will compare the performance of the logit NU model with two benchmarks: expected utility, and cumulative prospect theory (CPT). For these two, we take the standard approach to combine them with the logit model for estimation (see Train (2003)). For example, for the expected utility benchmark, take any expected utility function $U : X \to \mathbb{R}$. Given any data point with lotteries x and y, we use the probability that $U(x) + \varepsilon_x > U(y) + \varepsilon_y$ to predict the choice probability of x over y, in which $\varepsilon_x, \varepsilon_y \overset{\text{iid}}{\sim} \text{GEV}_{\text{I}}(1)$.[14] To evaluate the performance, we first find the expected utility function that minimizes the MSE using the training dataset (the *training MSE*). Then, we take the estimated expected utility function to compute the MSE using the testing dataset (the *testing MSE*) to measure the expected utility model's performance. We will use the same approach to analyze the CPT model.

We roughly describe how we train the logit NU model and evaluate its performance. More details can only be provided later (see Section 4.3), after we describe how we will parameterize the model. We estimate the logit NU model based on the training dataset using *cross-validation* and evaluate its performance with the test dataset. Our estimation procedure has two steps. The first step is to select the *hyperparameters*, such as the number of hidden layers and the width of each hidden layer. Obviously, if we select hyperparameters by minimizing the training MSE, we will want bigger networks and may overfit. Therefore, for each set of hyperparameters, we use the training dataset to estimate the logit NU model and compute the *leave-one-out* cross-validation (LOOCV) MSE (see Chapters 5 and 7 of Hastie et al. (2009)). We select the hyperparameters that yield the lowest LOOCV MSE. The second step is to estimate/train the logit NU model under the selected hyperparameters using the training dataset and compute its training MSE. To evaluate the performance of the model, we take the trained logit NU model to compute the testing MSE using the testing dataset. As is standard in machine learning literature, the training algorithm is random in nature. Thus, we estimate the logit NU model under the selected hyperparameters for multiple times, evaluate the testing MSE each time, and report the average testing MSE across the repetitions.

4.2. Parameterization of the benchmark

Two classic theories will be used as our benchmark, expected utility theory and CPT. Our first observation is that under the current data we must parameterize the models, including the expected utility model, to avoid overfitting.

Let us illustrate this through the expected utility model. Let $\{\pi_1, \ldots, \pi_n\} \subseteq \mathbb{R}$ denote the set of all monetary prizes. The expected utility model is $U(x) = \sum_{i=1}^{n} x_i u(\pi_i)$ for each lottery x, in which u is the Bernoulli index. Also consider the CARA[15] expected utility

[14] The literature using this approach to estimate expected and non-expected utility models is immense. See Harrison et al. (2007) and Noussair et al. (2014), among others.

[15] We do not consider the alternative and equally popular expected utility model, the constant-relative-risk-aversion (CRRA) expected utility model, mainly because some prizes are negative and hence are not well defined for the CRRA Bernoulli index.

model: For each lottery x,

$$U_{\text{CARA}}(x) = \sum_{i=1}^{n} x_i u(\pi_i) \text{ with } u(\pi_i) = \frac{\beta}{\alpha}(e^{\alpha\pi_i} - 1). \tag{2}$$

As usual, $-\alpha$ measures the decision maker's risk aversion. As α approaches 0, $u(\pi_i)$ converges to $\beta\pi_i$, which is a risk-neutral Bernoulli utility index. The parameter $\beta \in \mathbb{R}_+$ is a normalization parameter that is necessary in discrete choice estimations.

Combining these with the logit model, as explained in the previous subsection, we can find the best expected utility function and the best CARA expected utility function. The training and testing MSE×100 for the CARA expected utility model are 2.28 and 1.98, respectively, with essentially zero standard deviations. Compared with the CARA expected utility model, the expected utility model's training MSE is certainly lower, since the expected utility model is more general, but its testing MSE turns out to be about 10 times higher.[16]

This is largely due to overfitting (see Fig. 3). A more general model can explain more phenomena (have a lower training MSE), but that does not imply that it will predict well.[17] The classic example is to use a polynomial to fit a dataset generated by a linear function plus noises. The same issue applies to the CPT model and the logit NU model. Hence, parameterization will be necessary for the CPT model and the logit NU model as well.

Next, we examine the CPT benchmark, which is arguably the most popular non-expected utility model. We consider a standard parameterization in the literature: The probability-weighting function is equal to $\frac{\delta p^\gamma}{\delta p^\gamma + (1-p)^\gamma}$ for any cumulative/survival probability $p \in [0, 1]$ irrespective of the outcome. The value function takes the CARA form in both the gain (risk-averse) and loss (risk-seeking) regions, with a loss-aversion coefficient weakly larger than 1. Note that due to the convexity of the value function in the loss region, even when probabilities are not distorted the CARA expected utility model is not a special case of the CPT model. These two models only intersect at the risk-neutral case without probability distortion.

We find that CPT's training MSE×100 is 2.26 and testing MSE×100 is 1.98. Hence, under the current dataset, CPT does not seem to outperform the (CARA) expected utility model in terms of predictive power, although its performance is significantly better than the risk-neutral expected utility model.[18] One potential reason is that the testing dataset does not include the kind of lotteries involved in the fourfold pattern of risk attitudes, since the binary choice problems designed to replicate well-known behavioral phenomena are all in the training dataset. This also explains why for most of our results the testing MSE is lower than the training MSE—presumably, the replication problems are likely the harder ones for the participants.

It should also be noted that our analysis omits a factor that might influence the relative performance of the expected utility model and the CPT model. Specifically, each problem in the dataset is repeated 25 times, and after the first 5 repetitions, participants can observe the realized outcomes of lotteries from previous repetitions. Hertwig and Erev (2009) point out that decision makers tend to overestimate/overweight rare events when learning about lotteries through descriptions, but tend to underestimate/underweight these events when learning from experience, such as observing lottery outcomes. In our empirical analysis, we do not differentiate between choices made before and after observing lottery realizations, as our theoretical framework is silent about this distinction. Therefore, pooling these two types of choices might affect the relative predictive accuracy of the expected utility and CPT models. However, our subsequent analysis using the logit NU model should not be much affected, since the benchmark we adopt from this subsection is already optimized for the pooled choice data.

Note that our CPT estimation suggests little probability distortion, which is consistent with recent findings by Bernheim and Sprenger (2020). In addition, in our estimation, the constraint that the subjects are risk seeking in the loss domain turns out to be binding, suggesting that this behavioral phenomenon may not be consistent with the dataset. Our finding that the expected utility model performs well is also consistent with a series of recent empirical results (see Footnote 1).

Therefore, for the rest of the paper, we use the CARA expected utility model as the benchmark, whose training MSE×100 is 2.28 and testing MSE×100 is 1.98.

4.3. Behavioral neurons and parameterization of the NU function

As we have explained, given the current dataset, parameterization is necessary for the logit NU model. In particular, the parameterization we are looking for ought to help overcome the overfitting problem and, at the same time, retain the model's flexibility in the right manner.

The first idea comes from the observation that the affine functions in the first hidden layer of the NU function are expected utility functions. Therefore, it is possible that replacing first-hidden-layer neurons with CARA expected utility functions could help.

[16] If we have a large amount of data, the expected utility model should outperform the CARA expected utility model—but given the current dataset, the expected utility model's training MSE×100 is 1.07 and its testing MSE×100 is 19.47. Note that there are prizes that appear in the testing dataset but not in the training dataset. For those prizes, their Bernoulli indices cannot be properly estimated. However, even if we exclude the binary choice problems that contain prizes that only appear in the testing dataset, the testing MSE×100 of the expected utility model only reduces to 17.82, which is still much higher than that of the CARA expected utility model.

[17] We also examine a generalization of the above CARA expected utility model by allowing the CARA parameter to depend on participants' genders. The resulting testing MSE×100 is higher than 1.98 for both genders (2.03 for females and 2.14 for males), which suggests that analyzing the pooled data may be more effective.

[18] The testing error is still higher than the CARA expected utility model's if the value function is parameterized via the CRRA form, or if we allow the value function to be convex in the gain region or concave in the loss region so that the CARA expected utility model is nested as a special case. We also compute the testing MSE of the estimated CPT model from Tversky and Kahneman (1992) with our data, which turns out to be significantly higher.

By doing so, we also drop the activation function in the first hidden layer. However, the resulting model is still a logit NU model (see Section 3.1).

It turns out that this restriction destroys too much flexibility of the logit NU model. We allow the first hidden layer's width to be 15, 20, or 25; the number of hidden layers above the first to be 0, 1, or 2; and the width of the hidden layers above the first to be 15, 20, or 25. The best testing MSE×100 we obtain from these NEU functions is 1.97, which is barely better than the CARA benchmark.

To see what kind of useful flexibility has been removed, consider the certainty-effect neurons and reference-dependence neurons from Sections 3.1 and 3.2. These are neurons in the first hidden layer and help us capture well-documented behavioral effects, but are assumed away if we focus on CARA expected utility functions for the first hidden layer.

Presumably, there are other useful behavioral neurons and we may use statistical methods to endogenously select which behavioral neurons are best to use. This will be a more general approach, but it is not entirely clear which list of behavioral neurons should be used. In this paper, we take a simple approach. We show that by requiring that the first hidden layer consist of at most the following three kinds of neurons, a CARA expected utility function, certainty-effect neurons, and reference-dependence neurons, we can already mitigate the overfitting problem and obtain significant empirical performance improvements. This special first hidden layer will be called a *behavioral layer*. The behavioral layer is subsequently concatenated with additional standard hidden layers (see Definition 2), except that we will apply one restriction to the second-hidden-layer neurons due to the use of certainty-effect neurons, which we will explain shortly.

Specifically, the behavioral layer may consist of the following three types of neurons:

1. *The CARA neuron*: A CARA neuron is the function $U_{\text{CARA}} : X \to \mathbb{R}$ defined in (2). When estimating its parameters α and β, α is initialized uniformly at random in $[-1, 1]$ and β is initialized uniformly at random in $[0, 1]$.
2. *The certainty-effect (CE) neuron*: For any lottery x, a CE neuron with respect to the ith prize is a function from X to $\mathbb{R}$ that takes the following form:

   $$U^j_{\text{CE}_i}(x) = \max\{x_i - \eta_j, 0\},$$

   $j = 1, 2$ (see Section 3.1); that is, we allow the decision maker to have two types of CE neurons, each with a possibly different η_j (the threshold parameter).
   Fixing j, we require that η_j be identical across different prizes. Otherwise, if some prize never shows up in the training dataset, we will not be able to estimate the two threshold parameters for that prize. For the same reason, fixing j, we require that every second-hidden-layer neuron attach the same weight to the CE neurons $U^j_{CE_i}$, $i = 1, \dots, n$, in the affine aggregation. This is the restriction on the second hidden layer we stated previously. The threshold parameter η_j is initialized uniformly at random in $[0.9, 0.99]$.
3. *The reference-dependence (RD) neuron*: For any lottery x, an RD neuron is a function from X to $\mathbb{R}$ that takes the following form:

   $$U^j_{\text{RD}}(x) = \max\left\{\sum_{i=1}^{n} x_i \max\{\lambda_j \pi_i - \gamma_j, 0\}, \kappa_j\right\},$$

   $j \in \{1, \dots, n_{\text{RD}}\}$, in which π_i is the monetary prize to which x assigns probability x_i.[19] We allow the decision maker to have n_{RD} types of RD neurons, each of which is characterized by three parameters: the loss-aversion coefficient $-\lambda_j$, the reference point $\frac{\gamma_j}{\lambda_j}$, and the threshold parameter κ_j. All three parameters are initialized at random according to the standard Gaussian distribution, and n_{RD} is left as a hyperparameter.

To summarize, in the behavioral layer we at most have one CARA neuron, two types of CE neurons, and n_{RD} RD neurons. We do not turn the number of types of CE neurons into a hyperparameter only to shorten computation time. The CE neurons only affect a small area of X, and we believe that allowing for two types of CE neurons will be adequate. We only use one CARA neuron because we have seen that having multiple CARA neurons does not help much. Next, additional hidden layers will be concatenated with the behavioral layer, whose number of layers and width are hyperparameters. The affine-aggregation parameters of those layers are initialized by the standard Gaussian distribution. See Fig. 4 for an illustration.

4.4. Training, regularization, and hyperparameter selection

Now we are ready to provide more details about how we estimate the logit NU model with the behavioral layer. We train the logit NU model with the behavioral layer using state-of-the-art machine-learning methods via the following neural network. Recall that each data point has two lotteries x and y. First, the neural network takes both x and y as the input. Next, it derives $U(x)$ and $U(y)$ through two *separate and identical* neural networks implied by the NU function U with the behavioral layer. Last, it uses the probability that $U(x) + \varepsilon_x > U(y) + \varepsilon_y$ as the output to predict the choice probability of x for this data point, in which again $\varepsilon_x, \varepsilon_y \overset{\text{iid}}{\sim} \text{GEV}_{\text{I}}(1)$.

We train this neural network using adaptive moment estimation (also known as Adam; see Kingma and Ba (2017)) with minibatches of size 20, which are randomly selected at each epoch. The learning rate is 0.0002 for parameters of the additional

[19] We have used λ and γ at other places but, with an abuse of notation, the ones with subscripts are reserved for RD neurons.

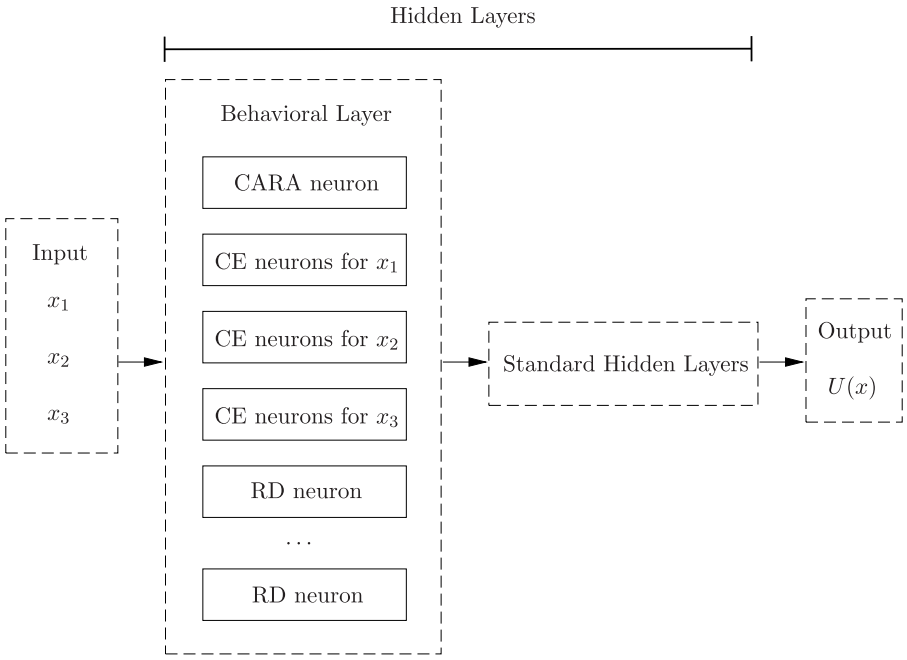


Fig. 4. In the empirical analysis, we will vary the number of standard hidden layers and the neurons we include in the behavioral layer.

hidden layers and the output layer, and 0.00002 for the parameters of the behavioral layer. To regularize the training, we use ℓ_2-norm regularization with coefficient 0.0002 (see Chapter 7 of Goodfellow et al. (2016)). Meanwhile, we stop the training after 5,000 epochs, in which each epoch goes through the 9 minibatches in a random order. Again, we use the MSE of the model as the metric to evaluate the performance of a logit NU model.

To select the hyperparameters (or equivalently, the architecture of the neural network), we divide the logit NU models into groups. For the specification of the behavioral layer, we consider CARA+RD, CARA+CE, and CARA+RD+CE. For the number of hidden layers, we consider three cases, one, two, and three/four. Note that a logit model with one hidden layer only has the behavioral layer. We divide them in this way because logit NU models with one hidden layer and logit NU models with two layers have rather different interpretations, and both are much easier to interpret than logit NU models with more layers. In total, we consider $3 \times 3 = 9$ groups of architectures.

We use LOOCV to select an architecture (i.e., a set of hyperparameters) within each group on the training dataset (of size 169). The width of the additional hidden layers (if any) concatenated with the behavioral layer may be uniformly 15, 20, or 25. The number of RD neurons may be 15, 20, or 25. For example, in the two-hidden-layer CARA+RD+CE group, there are $3 \times 3 = 9$ different architectures.

LOOCV trains each candidate model on only 168 data points and then makes a prediction on the left-out data point. Each of the 169 data points will be left out once. Then, LOOCV selects the candidate model with the least average LOOCV MSE over the 169 choices of left-out data points.

Given the selected model (set of hyperparameters), we train the logit NU model on the training dataset (of size 169), and then the trained model is taken to the testing dataset (of size 45) to compute the testing MSE. Since Adam is random in nature, this train-and-test procedure is repeated for 15 iterations and the average testing MSE is reported.[20]

4.5. Results

In Table 1, we report the selected architecture, the training MSE together with its standard deviation, and the testing MSE together with its standard deviation for each group of models.

Our first finding is that reasonably complex neural networks that have intuitive interpretations have the best performance. Measured by the testing MSE, a two-hidden-layer logit NU model (with the first hidden layer being the behavioral layer) that uses all three types of neurons in the behavioral layer—the CARA, CE, and RD neurons—has the best out-of-sample performance. Its testing error is more than one standard deviation lower than the CARA benchmark. Moreover, the architectures with width 25 in its additional layers or with 25 RD neurons are never selected in each group. While additional width or RD neurons decrease the training error, they may come at the cost of overfitting.

[20] With more repetitions the standard deviations will likely become much smaller, but this may significantly increase the computation time.

Table 1
Selection, training MSE, and testing MSE of groups of logit NU models: In the "CARA+RD" column, we require that the number of CE neurons be zero in the behavioral layer. In the "CARA+CE" column, we require that the number of RD neurons instead be zero. In the "CARA+RD+CE" column, the setup of the behavioral layer is as described in Section 4.3.

Number of Hidden Layers	CARA+RD	CARA+CE	CARA+RD+CE
1	RD: 20	–	RD: 20
2	RD: 15, Width: 15	Width: 20	RD: 20, Width: 15
> 2	RD: 15, Width: 15	Width: 15	RD: 20, Width: 15

(A): Selected architecture in each group

Number of Hidden Layers	CARA+RD	CARA+CE	CARA+RD+CE
1	2.223 (0.043)	2.327 (0.015)	2.205 (0.061)
2	1.993 (0.065)	2.243 (0.013)	1.880 (0.082)
> 2	1.473 (0.233)	2.194 (0.017)	1.680 (0.121)

(B): Training MSE×100 for the selected architecture in each group

Number of Hidden Layers	CARA+RD	CARA+CE	CARA+RD+CE
1	1.996 (0.118)	2.014 (0.012)	2.002 (0.144)
2	1.844 (0.151)	2.043 (0.040)	1.741 (0.173)
> 2	2.426 (0.870)	2.143 (0.041)	2.028 (0.407)

(C): Testing MSE×100 for the selected architecture in each group

Table 2
The two-hidden-layer CARA+RD+CE group: All architectures in the group outperform the CARA benchmark. While the minimum testing MSE is achieved with width 20 and 25 RD neurons, this architecture does not perform better than the selected architecture according to LOOCV MSE.

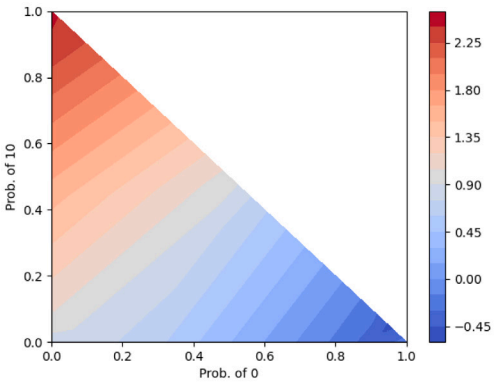
RD	Width	LOOCV MSE×100	Training MSE×100	Testing MSE×100
15	15	2.685	1.934 (0.099)	1.884 (0.131)
15	20	2.830	1.851 (0.099)	1.798 (0.172)
15	25	2.824	1.822 (0.109)	1.815 (0.174)
20	15	2.580	1.880 (0.082)	1.741 (0.173)
20	20	2.997	1.830 (0.077)	1.774 (0.240)
20	25	3.004	1.774 (0.109)	1.750 (0.253)
25	15	2.762	1.789 (0.096)	1.757 (0.232)
25	20	2.817	1.748 (0.100)	1.679 (0.231)
25	25	2.973	1.676 (0.062)	1.751 (0.264)

Our second finding is that including the CE and RD neurons can improve the performance of the logit NU model under the right architecture (recall the testing MSE×100 of the CARA benchmark is 1.98). However, including only CE neurons (together with the CARA neuron) is not very helpful. It is when both CE and RD neurons are included in the behavioral layer and one additional hidden layer is concatenated that the logit NU model starts to stand a chance to outperform the CARA benchmark significantly. This finding suggests that allowing multiple ways to aggregate the behavioral neurons in the model may better capture the decision making process. In Table 2, we report the LOOCV MSE, the training MSE, and the testing MSE for each of the 9 architectures in the two-hidden-layer CARA+RD+CE group. Every architecture in this group outperforms the CARA benchmark.

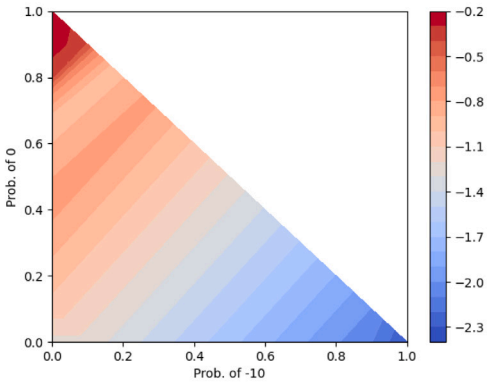
Our findings suggest that imposing assumptions on neural networks' structures based on economists' knowledge from decision theory and behavioral economics is useful in predictions. In deep learning, people have found that introducing inductive biases is extremely useful. Common forms of inductive biases include specific choices of neural network structures and optimization algorithms. For example, applying certain optimization algorithm (e.g., stochastic gradient descent) to certain neural-network models (e.g., residual neural networks) may favor simpler models over more complex ones (Neyshabur et al., 2014). Convolutional neural networks borrow ideas from how human beings understand pictures, and have been proven to be extremely useful for image recognition (see Aghdam and Heravi (2017)). Our approach introduces behavioral neurons as the inductive bias, which restricts the model class to one that better captures human behavior.[21] Consequently, we achieve significantly better performance both in sample and out of sample.

At the first glance, Table 1 may seem problematic since the testing MSEs are mostly lower than the corresponding training MSEs, and the training MSEs of some logit NU models can be higher than the CARA benchmark (2.28). This is in fact not surprising. While the CARA benchmark is obtained by minimizing the training MSE, we are not selecting architectures based on training MSE for logit NU models. Instead, we select architectures based on LOOCV MSE. Essentially, we sacrifice the fit on the training dataset in return for better out-of-sample predictions. Moreover, as we have explained before, the training dataset includes binary choice problems designed to replicate classic behavioral phenomena. By contrast, the testing dataset are mostly random binary choice problems. Thus, it seems plausible to us that the binary choice problems in the training dataset may be more prone to mistakes, which negatively affects the training MSEs.

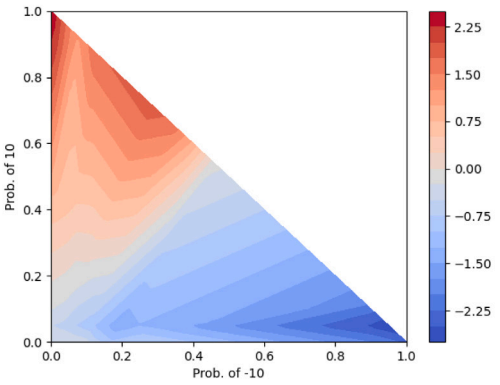
[21] The idea of constructing variables/features based on behavioral effects as the input of machine-learning models has also appeared in Erev et al. (2017) and Plonsky et al. (2017). Our approach differs from theirs: We only consider assumptions that are compatible with our axioms.



(A) Lotteries over prizes 10, 5, and 0



(B) Lotteries over prizes 0, −5, and −10



(C) Lotteries over prizes 10, 0, and −10

Fig. 5. Indifference curves of the estimated model within certain subspaces.

4.5.1. The best-performed model and mixed lotteries

Recall that we select hyperparameters by cross-validation based on the training dataset, and with the selected hyperparameters, we train the model based on the training dataset and compute its performance on the testing dataset. The reported MSEs and standard errors in Table 1c are calculated over 15 iterations of the train–test procedure. Out of the 15 iterations for what turns out to be the best-performed architecture (two-hidden-layer logit NU with CARA+RD+CE), the minimum testing MSE achieved is as little as 1.36. We report the estimated parameters in Appendix C.

Upon inspecting the estimated parameters of the logit NU model that achieved the minimum testing MSE, we find some interesting characteristics. Among others, the estimated reference points ($\frac{\gamma_i}{\lambda_i}$'s) are mostly close to zero in absolute value. As a result, for pure-gain or pure-loss lotteries, the estimated model exhibits similar patterns to the example shown in Fig. 2—the utility function is largely an expected utility one with a boost when the probability of receiving a prize is larger than the CE neurons' thresholds. In Fig. 5, we plot the indifference curves of the model within various subspaces over different prize combinations.

Within subspaces that involve both gains and losses, however, as can be seen from Fig. 5(c), the indifference curves exhibit distinct patterns for lotteries that have positive expected values and those that have negative expected values, potentially reflecting the difficulty of evaluating such lotteries for the decision maker. Given that the best-performed logit NU model behaves similarly to expected utility for pure-gain and pure-loss lotteries, it is likely that the binary choice problems that involve lotteries with both gains and losses (called *mixed lotteries*) are those in which the logit NU model performs the best relative to the CARA benchmark.

To investigate this, we compute the MSE for the best-performed logit NU model and the previously estimated CARA expected utility benchmark, respectively, on such binary choice problems in the testing dataset. Among 45 binary choice problems in the

testing dataset, 25 of them involve mixed lotteries. We find that the testing MSE×100 of logit NU is 1.16 and that of the CARA benchmark is 2.24, which confirms our hypothesis.[22]

Several recent studies (Blavatskyy et al. (2022); Bouchouicha et al. (2023); and Dembo et al. (2024)) have found that expected utility theory performs well empirically. However, these studies mainly focus on pure-gain lotteries. Our observation that the expected utility model does not perform well over mixed lotteries is consistent with experimental evidence in Chew et al. (2022), which documents that participants' choices are significantly more stochastic over mixed lotteries than their choices over pure-gain/pure-loss lotteries. Thus, while expected utility theory explains choices over pure lotteries well, non-expected utility models may be more effective in dealing with mixed lotteries.

It is worth noting that the kinks in the indifference curves generally face toward or away from the origin. This indicates that the ratio of the probability of winning to the probability of losing may be a significant factor in the decision-making process of the subjects.[23] The estimated level of utility tends to increase when the ratio of winning to losing is either high or relatively low. Furthermore, when the ratio exceeds one but is not excessively high, the estimated level of utility remains relatively stable while holding the probability of receiving the intermediate prize constant. This non-monotonic pattern is prevalent in subspaces that involve both positive and negative prizes, indicating the need for further investigation.

5. Axiomatic characterization

In this section, we introduce axioms to be imposed on the SCF and characterize the Luce NU model and the logit NU model. The role of the axioms is explained at the end of the Introduction. The first two axioms are from McFadden (1973).

Axiom 1 (*Positivity*). *For any* $A \in \mathcal{M}$ *and* $x \in A$, $\rho(x, A) > 0$.

Axiom 2 (*Independence of Irrelevant Alternatives*). *For any* $x, y \in X$ *and* $A \in \mathcal{M}$ *such that* $x, y \in A$, $\rho(x, A)\rho(y, \{x, y\}) = \rho(y, A)\rho(x, \{x, y\})$.

McFadden (1973) argues that a zero probability is empirically indistinguishable from a positive but small probability. Therefore, there is little loss of generality from imposing positivity. Independence of irrelevant alternatives (IIA) is a more controversial axiom, but the reason it is violated is not crucial for our analysis in this paper. McFadden shows that these two axioms characterize the logit model, which we will formally define later. In Section 6, we will briefly discuss a potential nested logit generalization of the logit NU model.

Following Block and Marschak (1960), we use the SCF to define a binary relation on X called the *stochastic preference*.

Definition 5. For any $x, y \in X$, we say that x is stochastically preferred to y, denoted by $x \succsim y$, if $\rho(x, \{x, y\}) \geqslant \rho(y, \{x, y\})$.

We use $\succ$ and $\sim$ to denote $\succsim$'s asymmetric and symmetric parts, respectively. The axiom below requires that the SCF be continuous.

Axiom 3 (*Continuity*). *For any sequence* (x_n) *in* X *whose limit is* $x \in X$, $\lambda \in [0, 1]$, *and* $y \in X$ *distinct from* (x_n) *and* x, *if* $\rho(x_n, \{x_n, y\}) \geqslant (\leqslant)\lambda$ *for all* n, *then* $\rho(x, \{x, y\}) \geqslant (\leqslant)\lambda$.

The next axiom is one of the main axioms from Ke and Zhao (2024). A subset of X is said to be a neighborhood of an alternative x if it is open and convex and contains x.

Axiom 4 (*Weak Local Bi-independence*). *Any* $z, \tilde{z} \in X$ *with* $z \sim \tilde{z}$ *have neighborhoods* L *and* $\tilde{L}$, *respectively, such that for any* $x \in L$, $\tilde{x} \in \tilde{L}$, *and* $\lambda \in (0, 1)$, *we have* $x \succsim \tilde{x} \Leftrightarrow \lambda x z \succsim \lambda \tilde{x} \tilde{z}$.

This axiom weakens a condition that appears in expected utility theory. In expected utility theory, the independence axiom characterizes linear/expected utility functions, and is equivalent to the following condition called bi-independence in Ke and Zhao (2024): For any $\lambda \in (0, 1)$ and lotteries $x, \tilde{x}, z, \tilde{z}$ such that $z \sim \tilde{z}$, we have $x \succsim \tilde{x} \Leftrightarrow \lambda x z \succsim \lambda \tilde{x} \tilde{z}$.

Motivated by the fact that decision makers' preferences often cannot be represented by linear functions due to violations of (bi-)independence, but may exhibit some form of linearity locally, Ke and Zhao (2024) propose weak local bi-independence to weaken bi-independence. Under weak local bi-independence, fixing any z and $\tilde{z}$, locally around z and $\tilde{z}$, alternatives that are mixed with z and $\tilde{z}$, respectively, must satisfy the property stated in bi-independence.

To understand the motivation for weak local bi-independence, consider a well-known violation of independence, the Allais paradox. Given the following two pairs of lotteries, most decision makers choose the left-hand lottery from the first pair and the right-hand lottery from the second:

[22] Note that allowing for probability weighting, loss aversion, and risk seeking over losses does not help much. In our estimation, CPT's out-of-sample performance exhibit similar patterns as CARA expected utility. Over mixed lotteries, its testing MSE×100 is 2.28; over pure-gain/pure-loss lotteries, its testing MSE×100 is 1.61.

[23] This pattern is robust across subspaces that involve both positive and negative prizes, regardless of their magnitudes. See Appendix C for additional indifference maps in various subspaces.

First pair		Second pair	
100%: \$1M	3%: \$0 87%: \$1M 10%: \$1.5M	87%: \$0 13%: \$1M	90%: \$0 10%: \$1.5M

Such choices imply a violation of independence. To see this, let δ_r be the degenerate lottery that pays \$$r$ for sure. Let $x = \delta_{1\text{M}}$, $y = \frac{10}{13}\delta_{1.5\text{M}} + \frac{3}{13}\delta_0$, $z = \delta_{1\text{M}}$, and $z' = \delta_0$. Under these notations, the first pair of lotteries becomes $0.13xz$ and $0.13yz$, and the second pair becomes $0.13xz'$ and $0.13yz'$. Independence implies that $0.13xz \succsim 0.13yz \iff 0.13xz' \succsim 0.13yz'$. Therefore, the Allais paradox violates independence and hence bi-independence. The interpretation of this violation is that decision makers tend to prefer risk-free alternatives, compared to what expected utility theory predicts, which is sometimes called the certainty effect.

Ke and Zhao (2024) point out that if the two pairs of lotteries are reconstructed so that the right-hand lottery in the first pair becomes almost risk-free, then the decision maker may not be attracted by $\delta_{1\text{M}}$ in the first pair nearly as much, and hence the certainty effect may not be that strong to trigger violations of (bi-)independence. To see this more concretely, consider the example below with $0.013xz$ and $0.013y^*z$ in the first pair and $0.013xz^*$ and $0.013y^*z^*$ in the second, in which $y^* = \frac{10}{13}\delta_{1.5M} + \frac{3}{13}\delta_{0.5M}$ and $z^* = \delta_{0.5\text{M}}$:

First pair		Second pair	
100%: \$1M	0.3%: \$0.5M 98.7%: \$1M 1%: \$1.5M	98.7%: \$0.5M 1.3%: \$1M	99%: \$0.5M 1%: \$1.5M

Intuitively, the certainty effect in the above example should be much weaker, and therefore it seems more plausible to require that some notion of independence hold locally rather than globally.

Ke and Zhao (2024) show that a complete, transitive, and continuous binary relation satisfies weak local bi-independence if and only if the binary relation can be represented by an NU function. Before defining the NU function, we introduce our final axiom. As will be seen in Theorem 1, weak local bi-independence is too weak to yield the logit NU model. The following axiom strengthens weak local bi-independence.

Axiom 5 (*Weak Local Bi-invariance*). *Any $z, \tilde{z} \in X$ with $z \sim \tilde{z}$ have neighborhoods L and $\tilde{L}$, respectively, such that for any $x \in L$, $\tilde{x} \in \tilde{L}$, $\alpha, \beta \in [0,1]$, and $\lambda \in (0,1)$, we have $x \succsim \tilde{x} \Leftrightarrow \rho(\lambda x(\alpha xz), \{\lambda x(\alpha xz), \lambda z(\alpha xz)\}) \geqslant \rho(\lambda\tilde{x}(\beta\tilde{x}\tilde{z}), \{\lambda\tilde{x}(\beta\tilde{x}\tilde{z}), \lambda\tilde{z}(\beta\tilde{x}\tilde{z})\})$.*

Weak local bi-invariance combines weak local bi-independence with the idea of an axiom in Ke (2018) that is crucial to the characterization of the logit expected utility model. That axiom essentially requires that for any $x, y, z, \tilde{x}, \tilde{y}, \tilde{z} \in X$ such that $z \sim \tilde{z}$, $x \sim \tilde{x}$, and $\lambda \in (0,1)$, we have $\rho(\lambda xy, \{\lambda xy, \lambda zy\}) = \rho(\lambda\tilde{x}\tilde{y}, \{\lambda\tilde{x}\tilde{y}, \lambda\tilde{z}\tilde{y}\})$. Note that in weak local bi-invariance, we pick $y = \alpha xz$ and $\tilde{y} = \beta\tilde{x}\tilde{z}$, so that all lotteries involved in L are mixtures of x and z, and all lotteries involved in $\tilde{L}$ are mixtures of $\tilde{x}$ and $\tilde{z}$, which captures the same idea as how weak local bi-independence weakens bi-independence. We show in the Appendix that, given the other axioms, weak local bi-invariance implies weak local bi-independence.

Our main theoretical result is the following.

Theorem 1. *The SCF ρ has a Luce NU representation if and only if ρ satisfies positivity, IIA, continuity, and weak local bi-independence. The SCF ρ has a logit NU representation if and only if ρ satisfies positivity, IIA, continuity, and weak local bi-invariance.*

The first statement of the theorem is straightforward. Under positivity and IIA, there must exist a function $V : X \to \mathbb{R}_{++}$ such that for any menu A and $x \in A$,

$$\rho(x, A) = \frac{V(x)}{\sum_{y \in A} V(y)}.$$

This is called a Luce rule (Luce, 1959). Next, it can be shown that the stochastic preference is complete and transitive. Together with continuity and weak local bi-independence, from Ke and Zhao (2024), we know that the stochastic preference can be represented by an NU function U. Moreover, it can be shown that both U and V represent the stochastic preference. Therefore, there exists a strictly increasing function that transforms U into V. Denoting that function as ϕ, we obtain the first statement of the theorem.

According to Ke and Zhao (2024), an NU function U must be a continuous finite piecewise linear function, and a continuous finite piecewise linear representation of a binary relation is unique up to a strictly increasing continuous finite piecewise linear transformation. It is well known that V in the Luce rule is unique up to a positive scalar multiplication. Hence, uniqueness results of ϕ can be easily obtained, although it is not important to our analysis.

The Luce NU representation becomes a logit NU representation when the function ϕ is exponential, which is not true in general. When it is, the standard random utility formulation of the logit model applies. See Ke (2018) for examples of non-exponential ϕ functions and discussion of the distinction between Luce rules and logit models.

The second statement of Theorem 1 says that if we replace weak local bi-independence with weak local bi-invariance, we can ensure that the function ϕ in the Luce NU representation is exponential. The proof of the theorem can be found in Appendix A.

6. Concluding remarks

In this paper, we introduce, empirically analyze, and characterize the logit neural-network utility model. In the context of decision-making under risk, the model feeds the probability distribution of a lottery to a neural network and essentially outputs the Luce value of the lottery. We show how to use behavioral neurons to capture behavioral patterns and mitigate overfitting. We find that simple logit NU models with behavioral neurons predict better than expected utility theory and cumulative prospect theory out of sample. In particular, the logit NU models perform well in binary choice problems involving lotteries with both positive and negative prizes.

There are many alternative ways to apply machine learning techniques to predict choices under uncertainty. For example, using the same dataset, Plonsky et al. (2017) and Plonsky et al. (2019) estimate mappings from binary choice problems to choice probabilities using random forests enhanced with behavioral features. Although these models achieve higher predictive accuracy than ours, their underlying axiomatic foundations remain unclear at this stage. In contrast, our analysis deliberately focuses on special cases of the logit NU model, which has clearly defined and economically meaningful behavioral axioms. Moreover, it is straightforward to see how our axioms generalize or differ from those of classic economic models. Put differently, our primary objective is not to maximize predictive accuracy per se, but to optimize predictive power within a theoretically grounded economic framework.

One advantage of our approach is that the behavioral properties of our model are better understood. For instance, our axiomatic characterization demonstrates that the logit NU model satisfies a form of stochastic transitivity, a property that is generally considered desirable (see He and Natenzon (2024)). Therefore, all special cases of the logit NU model used in our empirical analysis satisfy this property. By contrast, the models employed by Plonsky et al. (2017) and Plonsky et al. (2019) are likely to violate this property, and it is unclear what forms of stochastic transitivity these alternative models may satisfy.

Our empirical strategy differs from the standard approach in the non-expected utility literature—we evaluate models by their our-of-sample predictive power. In particular, we take the separation of the training dataset and the testing dataset as given and have no control in this regard. Such a clean separation of training and testing datasets is a critical aspect of our approach.

In our model, the NU function is paired with the logit model. It is well known that the logit model is subject to the critique of the red-bus-blue-bus problem (see Debreu (1960)). One solution to the red-bus-blue-bus problem is called the nested logit model (see Train (2003)). It is straightforward to write down a nested logit version of our logit NU model so that our model can also avoid the red-bus-blue-bus problem. What might be more challenging is the generalization of our axiomatic characterization. Fortunately, recent work—by Kovach and Tserenjigmid (2022a), for example—has shown how to relax IIA to characterize the nested logit model. Therefore, it is possible to relax our axioms in a way similar to how Kovach and Tserenjigmid relax IIA to derive a characterization of the nested logit version of our model.

There are additional limitations of the logit model worth mentioning. For example, Apesteguia and Ballester (2018) show that the logit model may violate an intuitive monotonicity property, and Lu and Saito (2022) establish that the logit model—and indeed almost all mixed logit models—violates a natural behavioral condition known as convex substitutability. Our paper is among the first to integrate machine learning methods into economics through an axiomatic approach. Combining the NU function with the logit model is a natural first step in this direction. We anticipate that future research will explore alternative machine learning methods and pair them with stochastic choice models that exhibit more desirable behavioral properties.

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.

Appendix A. Proof of Theorem 1

Proof. For any $L \subseteq X$, let $\text{int}(L)$, $\text{cl}(L)$, ∂L, $\text{aff}(L)$, $\dim(L)$ denote the interior, closure, boundary, affine hull, and the dimension of the affine hull of L, respectively, in $\mathbb{R}^N$. For any $x \in X$ and $\varepsilon > 0$, let $B_\varepsilon(x)$ denote the open ball centered at x with radius ε. For any finite set of choice alternatives $\{x^1, \dots, x^m\}$, let $\overline{x^1 \dots x^m} := \text{co}(\{x^1, \dots, x^m\})$ be the convex hull of $\{x^1, \dots, x^m\}$.

The first statement is an immediate implication of Ke and Zhao (2024) and McFadden (1973). We will focus on the second statement. We will first show the sufficiency of the axioms and then the necessity. In this part of the proof, we will maintain the assumption that ρ satisfies, positivity, IIA, continuity, and local mixture bi-invariance.

By McFadden (1973), positivity and IIA imply that there exists a positive-valued function $V : X \to \mathbb{R}$ such that for any menu A and $x \in A$, $\rho(x, A) = V(x)/\sum_{y \in A} V(y)$. Because $\rho(x, \{x, y\}) = V(x)/(V(x) + V(y))$, we also know that $x \gtrsim y$ if and only if $V(x) \geqslant V(y)$. In other words, $\gtrsim$ can be represented by V. Therefore, $\gtrsim$ is complete and transitive.

The first lemma shows that under positivity and IIA, if ρ satisfies weak local bi-invariance, then $\gtrsim$ satisfies weak local bi-independence.

Lemma 1. *If ρ satisfies positivity, IIA and weak local bi-invariance, then $\gtrsim$ satisfies weak local bi-independence.*

Proof. Given $z \sim \tilde{z}$, let the neighborhoods in weak local bi-invariance be L and $\tilde{L}$, respectively. Pick any $x \in L$ and $\tilde{x} \in \tilde{L}$, and let $\alpha = \beta = 0$. Then weak local bi-invariance implies that for all $\lambda \in (0,1)$, $x \succsim \tilde{x} \Leftrightarrow \rho(\lambda xz, \{\lambda xz, z\}) \geqslant, \rho(\lambda\tilde{x}\tilde{z}, \{\lambda\tilde{x}\tilde{z}, \tilde{z}\})$. Note that under positivity and IIA, $z \sim \tilde{z}$ implies $V(z) = V(\tilde{z})$. Thus, $\rho(\lambda xz, \{\lambda xz, z\}) \geqslant \rho(\lambda\tilde{x}\tilde{z}, \{\lambda\tilde{x}\tilde{z}, \tilde{z}\})$ if and only if $V(\lambda xz) \geqslant V(\lambda\tilde{x}\tilde{z})$, which is equivalent to $\lambda xz \succsim \lambda\tilde{x}\tilde{z}$. Thus, $\succsim$ satisfies weak local bi-independence. □

Hence ρ has a Luce NU representation; that is, there exists an NU function $U : X \to \mathbb{R}$ and a strictly increasing continuous function $\phi : U(x) \to \mathbb{R}_{++}$ such that $V(x) = \phi(U(x))$. By Corollary 1 in Ke and Zhao (2024), there exists a finite collection of regular closed subsets, denoted as $X_1, X_2, \ldots, X_n$, whose union is X, such that U is affine on each of those subsets. Without loss of generality, we can assume each X_k is a convex polytope—the bounded intersection of finitely many closed half-spaces in $\mathbb{R}^N$. Clearly, each neuron in the first hidden layer defines a closed half-space; each neuron in the later layers, combined with its parent (and grand-parent, etc.) neurons, defines a finite collection of closed half-spaces.

Lemma 2. *There exists $m \in \mathbb{N}_+$ and $u_1, \ldots, u_m$ such that $\min_{x\in X} U(x) = u_1 < u_2 < \cdots < u_m = \max_{x\in X} U(x)$, and for all $i \in \{1, \ldots, m\}$ and $u \in [u_i, u_{i+1}]$,*

$$\phi(u) = c_i e^{u/a_i}$$

for some $a_i, c_i > 0$. Furthermore, $c_i e^{u_{i+1}/a_i} = c_{i+1} e^{u_{i+1}/a_{i+1}}$ for all $i \in \{1, \ldots, m-1\}$.

Proof. It suffices to show the claim with X replaced by X_k for an arbitrary $k \in \{1, 2, \ldots, n\}$. We will focus on the case in which U is not constant on X_k, since the claim is trivial otherwise. Pick $z^h = \arg\max_{x\in X_k} U(x)$ and $z^l = \arg\min_{x\in X_k} U(x)$. We can find such lotteries since U is affine on X_k.

For all $z \in \overline{z^h z^l}$, by weak local bi-invariance with $\tilde{z} = z$ and $\tilde{x} = x$, there exists $\varepsilon_z > 0$ such that for all $x \in B_{\varepsilon_z}(z)$, $\alpha, \beta \in [0,1]$ and $\lambda \in (0,1)$, we have

$$\rho(\lambda x(\alpha xz), \{\lambda x(\alpha xz), \lambda z(\alpha xz)\}) = \rho(\lambda x(\beta xz), \{\lambda x(\beta xz), \lambda z(\beta xz)\}),$$

which implies for all $u, u' \in [\min\{U(x), U(z)\}, \max\{U(x), U(z)\}]$ and $\lambda \in (0,1)$,

$$\frac{\phi(\lambda U(x) + (1-\lambda)u)}{\phi(\lambda U(z) + (1-\lambda)u)} = \frac{\phi(\lambda U(x) + (1-\lambda)u')}{\phi(\lambda U(z) + (1-\lambda)u')}.$$

Pick arbitrary $x \in B_{\varepsilon_z}(z)$ such that $U(x) > U(z)$. We have for all $\tilde{\lambda} \in (0, U(x) - U(z))$ and $\gamma, \delta \in [0, U(x) - U(z) - \tilde{\lambda}]$,

$$\frac{\phi_{x,z}(\tilde{\lambda} + \gamma)}{\phi_{x,z}(\gamma)} = \frac{\phi_{x,z}(\tilde{\lambda} + \delta)}{\phi_{x,z}(\delta)},$$

in which $\phi_{x,z}(\tilde{u}) = \phi(U(z) + \tilde{u})$ for all $\tilde{u} \in [0, U(x) - U(z)]$. We also have

$$\phi^*_{x,z}(\tilde{\lambda} + \gamma) = \phi^*_{x,z}(\tilde{\lambda})\phi^*_{x,z}(\gamma) \tag{3}$$

for all $\tilde{\lambda}, \gamma \in \mathbb{R}_{++}$ such that $\tilde{\lambda} + \gamma < U(x) - U(z)$, in which $\phi^*_{x,z} = \phi_{x,z}/\phi_{x,z}(0)$.

Eq. (3) is the multiplicative form of Cauchy's functional equation.[24] Since $\phi^*_{x,z}$ is continuous and strictly positive, it is clear that there exists $a_{x,z} > 0$ such that $\phi^*_{x,z}(u) = e^{u/a_{x,z}}$ for all $u \in (0,1)$. It then follows that for all $u \in (U(z), U(x))$,

$$\phi(u)/\phi(U(z)) = \phi_{x,z}(u - U(z))/\phi_{x,z}(0) = e^{\frac{1}{a_{x,z}}(u - U(z))}.$$

Thus, it is easy to see that for all $x \in B_{\varepsilon_z}(z)$ such that $U(x) > U(z)$, $a_{x,z} \equiv a_z$. Thus, for all $x \in B_{\varepsilon_z}(z)$ such that $U(x) > U(z)$,

$$\phi(U(x)) = \phi(U(z)) e^{\frac{1}{a_z}(U(x) - U(z))}.$$

By a symmetric argument, there exists $b_z > 0$ such that for all $x \in B_{\varepsilon_z}(z)$ such that $U(x) < U(z)$,

$$\phi(U(x)) = \phi(U(z)) e^{\frac{1}{b_z}(U(x) - U(z))}.$$

Note that $\{B_{\varepsilon_z}(z) : z \in \overline{z^h z^l}\}$ forms an open cover of $\overline{z^h z^l}$ and thus has a finite sub-cover. Combining this observation with the solutions to the Cauchy equations, we conclude that there exist $z^1, \ldots, z^m$ such that $U(z^l) = U(z^1) < U(z^2) < \cdots < U(z^m) = U(z^h)$ such that for all $i \in \{1, \ldots, m\}$ and $u \in [U(z^i), U(z^{i+1})]$,

$$\phi(u) = c_i e^{u/a_i}$$

for some $a_i, c_i > 0$. In addition, continuity of ϕ requires that $c_i e^{U(z^{i+1})/a_i} = c_{i+1} e^{U(z^{i+1})/a_{i+1}}$ for all $i \in \{1, \ldots, m-1\}$. Let $u_i = U(z^i)$ and we are done. □

The final step is to perform a continuous finite piecewise linear (CFPL) transformation of U and normalize ϕ to the exponential function.

[24] See Aczél (1966) for a thorough treatment of Cauchy's equations.

Lemma 3. *There exists an NU function $\tilde{U}$ that represents $\succsim$ and $\phi \circ U = e^{\tilde{U}}$.*

Proof. Let u_i, a_i, c_i be given by Lemma 2 for $i = 1, \dots, m$. Define $f : U(X) \to \mathbb{R}$ as follows: $f(u) = u/a_i + \ln c_i$ for $u \in [u_i, u_{i+1}]$ for $i = 1, \dots, m-1$. The second statement in Lemma 2 ensures that f is continuous. Thus f is a strictly increasing CFPL transformation. By Proposition 2 and Corollary 1 in Ke and Zhao (2024), $\tilde{U} := f \circ U$ is also an NU representation of $\succsim$. It is easy to see that $\phi \circ U = e^{\tilde{U}}$. □

Now we show the necessity of the axioms for the second statement. We focus on weak local bi-invariance since the necessity of the other axioms are trivial. Suppose ρ has a logit NU representation; that is, there is an NU function U such that

$$\rho(x, A) = \frac{e^{U(x)}}{\sum_{y \in A} e^{U(y)}}$$

for all $A \in \mathcal{M}$ and $x \in A$.

Clearly, U is also a CFPL function. By Theorem 2.1 in Ovchinnikov (2002), there exists distinct affine functions $U_1, \dots, U_n$ and index sets $I_1, \dots, I_m$ such that

$$x \succsim y \iff \max_{1 \leqslant j \leqslant m} \min_{i \in I_j} U_i(x) \geqslant \max_{1 \leqslant j \leqslant m} \min_{i \in I_j} U_i(y).$$

Since $U_1, \dots, U_n$ are distinct, for each $i \neq j$, $\mathrm{aff}(\{x \in X : U_i(x) = U_j(x)\})$ is either empty or defines an affine hyperplane in $\mathbb{R}^N$. Let $\mathcal{A}$ be the collection of these affine hyperplanes. Thus, $\mathcal{A}$ is an arrangement of hyperplanes in $\mathbb{R}^N$. A *region* of $\mathcal{A}$ in X is a connected component of $X \backslash (\bigcup_{H \in \mathcal{A}} H)$. Let $\mathcal{R}(\mathcal{A})$ be the collection of regions of $\mathcal{A}$ in X. For each $L \in \mathcal{R}(\mathcal{A})$, it is easy to see that L is nonempty, open, and $\mathrm{cl}(L)$ is a polytope. Let $\mathcal{P}(\mathcal{A}) := \{\mathrm{cl}(L) : L \in \mathcal{R}(\mathcal{A})\}$. Since $\mathcal{A}$ is finite, $\mathcal{P}(\mathcal{A})$ must be finite. Clearly $\bigcup_{P \in \mathcal{P}(\mathcal{A})} P = X$, and for any $P \in \mathcal{P}(\mathcal{A})$ there exists k such that $\max_{1 \leqslant j \leqslant m} \min_{i \in I_j} U_i(x) = U_k(x)$ for every $x \in P$.

For any $x \in X$, let $\mathcal{A}(x) := \{H \in \mathcal{A} : x \in H\}$ and consider $\mathcal{A}' = \mathcal{A} \backslash \mathcal{A}(x)$. Basically, we remove all the hyperplanes that contains x, if any. Clearly, there exists $L_x \in \mathcal{R}(\mathcal{A}')$ such that $x \in L_x$. It is clear that $x \in \bigcap \{P \in \mathcal{P}(\mathcal{A}) : x \in P\}$.

Next, we show that $L_x = \mathrm{int}\left(\bigcup \{P \in \mathcal{P}(\mathcal{A}) : x \in P\}\right)$. The claim is trivially true if $\mathcal{A}(x) = \supsetneq$. If $\mathcal{A}(x) \neq \supsetneq$, then by construction $\mathcal{A}(x)$ is an arrangement of hyperplanes in $\mathbb{R}^N$. Moreover, $x \in \bigcap_{H \in \mathcal{A}(x)} H$. It follows that x is in every closed half-spaces defined by hyperplanes in $\mathcal{A}(x)$. Thus, $x \in P'$ for every $P' \in \mathcal{P}(\mathcal{A}(x))$. Since $x \in L_x$, we have that $x \in P' \cap L_x$ for every $P' \in \mathcal{P}(\mathcal{A}(x))$. It is clear that

$$\{L' \cap L_x : L' \in \mathcal{R}(\mathcal{A}(x))\} = \{L \in \mathcal{R}(\mathcal{A}) : L \subseteq L_x\}.$$

It follows that $x \in P$ for every $P \in \mathcal{P}(\mathcal{A})$ such that $P \subseteq \mathrm{cl}(L_x)$. Since $x \in L_x$, we have $x \notin P$ if $P \nsubseteq \mathrm{cl}(L_x)$. Hence,

$$\begin{aligned}
\mathrm{cl}(L_x) &= \mathrm{cl}\left(\bigcup \{L' \cap L_x : L' \in \mathcal{R}(\mathcal{A}(x))\}\right) \\
&= \mathrm{cl}\left(\bigcup \{L \in \mathcal{R}(\mathcal{A}) : L \subseteq L_x\}\right) \\
&= \bigcup \{P \in \mathcal{P}(\mathcal{A}) : P \subseteq \mathrm{cl}(L_x)\} \\
&= \bigcup \{P \in \mathcal{P}(\mathcal{A}) : x \in P\}.
\end{aligned}$$

Note that since L_x is the interior of a polytope, it is regular open. Thus, $L_x = \mathrm{int}(\mathrm{cl}(L_x))$ and we are done with this step.

The last step is to show that this L_x construction is exactly what we want for weak local bi-invariance. Given $z, \tilde{z} \in X$ with $z \sim \tilde{z}$, by the convexity of each $P \in \mathcal{P}(\mathcal{A})$, it is clear that for any $x \in L_z$ and $\tilde{x} \in L_{\tilde{z}}$, $\overline{xz} \subseteq P$ and $\overline{\tilde{x}\tilde{z}} \subseteq P'$ for some $P, P' \in \mathcal{P}(\mathcal{A})$. Since U coincides with an affine function within P and P', we have for all $x \in L_z$, $\tilde{x} \in L_{\tilde{z}}$, $\alpha, \beta \in [0, 1]$, $\lambda \in (0, 1)$,

$$\rho(\lambda x(\alpha xz), \{\lambda x(\alpha xz), \lambda z(\alpha xz)\}) = \frac{e^{\lambda U(x) + (1-\lambda) U(\alpha xz)}}{e^{\lambda U(x) + (1-\lambda) U(\alpha xz)} + e^{\lambda U(z) + (1-\lambda) U(\alpha xz)}} = \frac{e^{\lambda U(x)}}{e^{\lambda U(x)} + e^{\lambda U(z)}},$$

and

$$\rho(\lambda \tilde{x}(\alpha \tilde{x}\tilde{z}), \{\lambda \tilde{x}(\alpha \tilde{x}\tilde{z}), \lambda \tilde{z}(\alpha \tilde{x}\tilde{z})\}) = \frac{e^{\lambda U(\tilde{x}) + (1-\lambda) U(\alpha \tilde{x}\tilde{z})}}{e^{\lambda U(\tilde{x}) + (1-\lambda) U(\alpha \tilde{x}\tilde{z})} + e^{\lambda U(\tilde{z}) + (1-\lambda) U(\alpha \tilde{x}\tilde{z})}} = \frac{e^{\lambda U(\tilde{x})}}{e^{\lambda U(\tilde{x})} + e^{\lambda U(\tilde{z})}}.$$

Since $U(z) = U(\tilde{z})$, we conclude that weak local bi-invariance holds. □

Appendix B. Sample choice problems in Plonsky et al. (2019)

See Fig. 6.

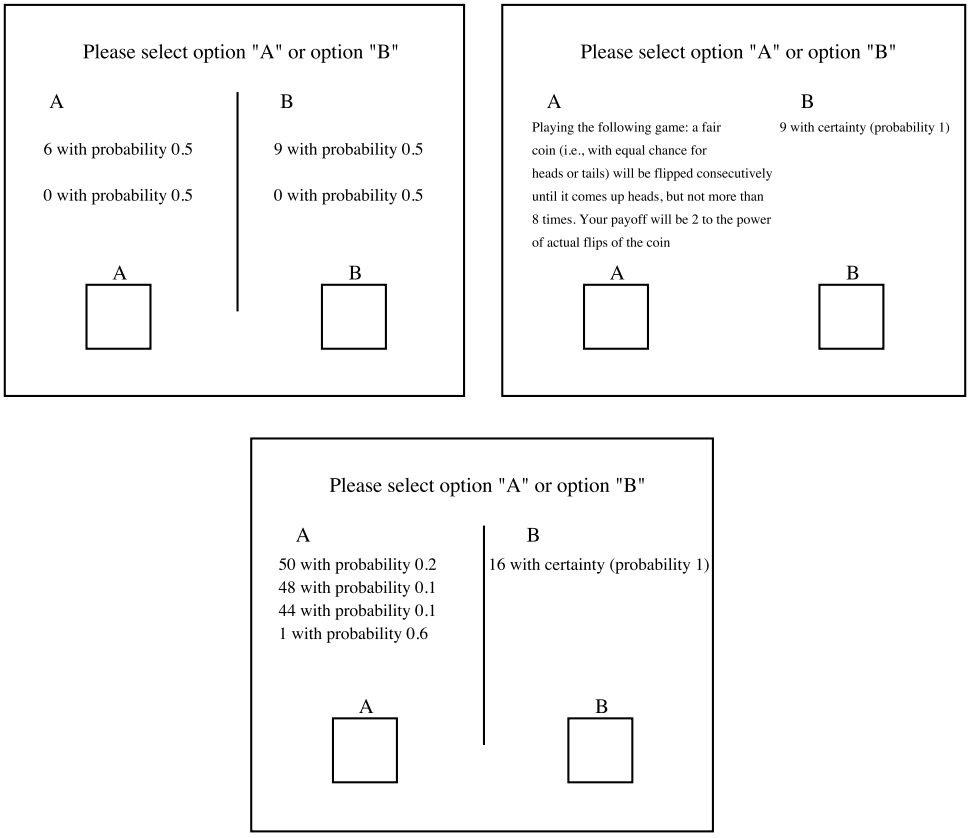


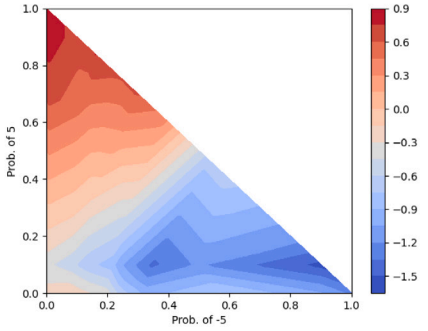
Fig. 6. Examples of the binary choice problems over lotteries in the experiments of Plonsky et al. (2019).

Appendix C. The best-performed model

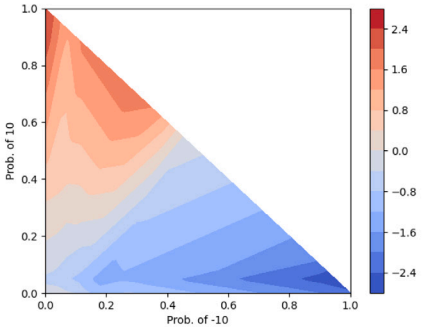
In this section, we report the best-performed model in our estimation. To describe the neural network, we will first present the indifference maps within various subspaces and then the exact estimates of the parameters.

C.1. Indifference maps

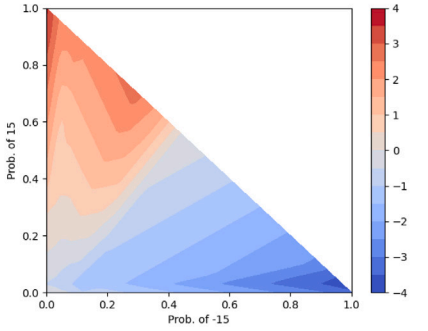
C.1.1. Subspaces with both positive and negative prizes



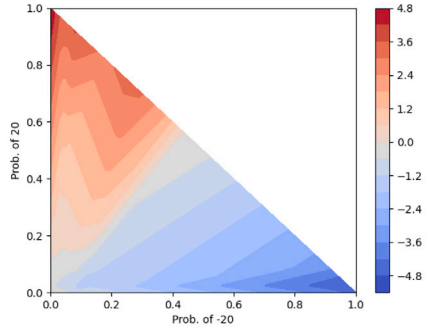
(A) Lotteries over prizes 5, 0, and −5



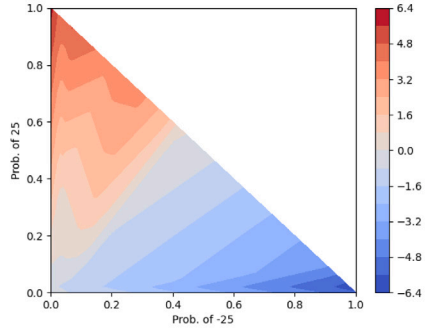
(B) Lotteries over prizes 10, 0, and −10



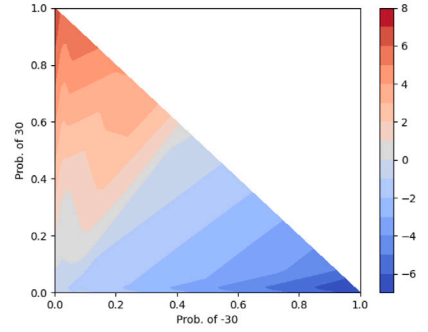
(C) Lotteries over prizes 15, 0, and −15



(D) Lotteries over prizes 20, 0, and −20

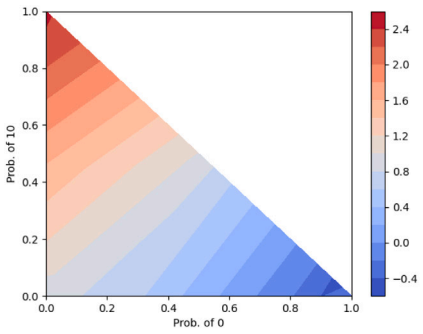


(E) Lotteries over prizes 25, 0, and −25

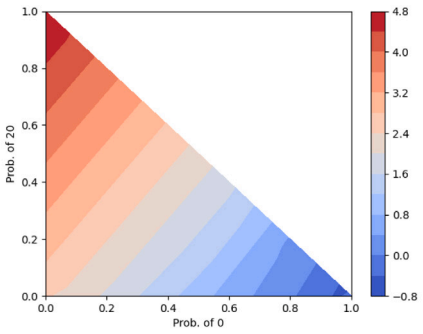


(F) Lotteries over prizes 30, 0, and −30

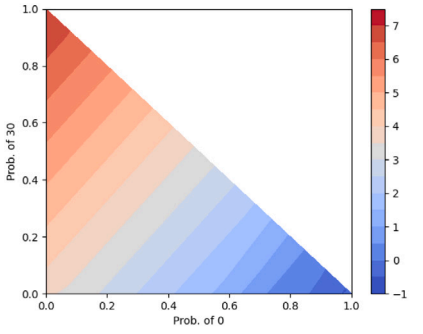
C.1.2. Subspaces with nonnegative prizes



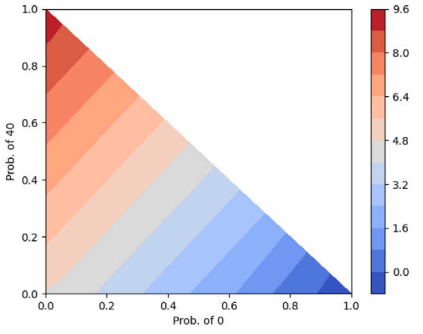
(A) Lotteries over prizes 10, 5, and 0



(B) Lotteries over prizes 20, 10, and 0

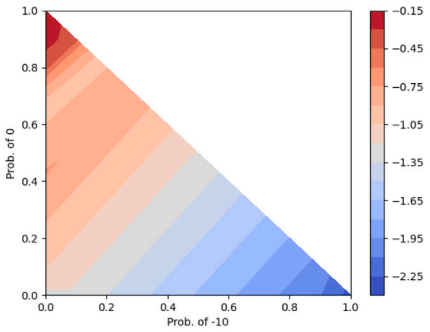


(C) Lotteries over prizes 30, 15, and 0

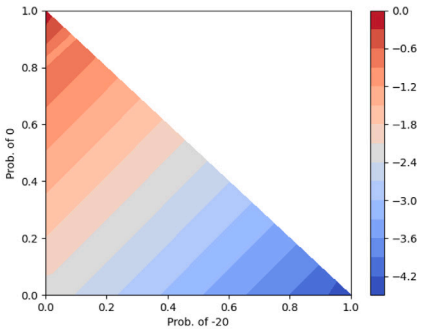


(D) Lotteries over prizes 40, 20, and 0

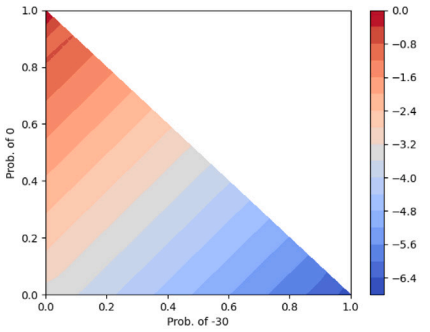
C.1.3. Subspaces with nonpositive prizes



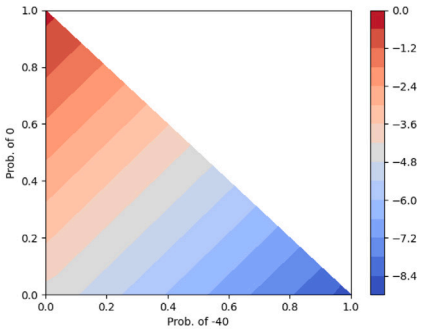
(A) Lotteries over prizes 0, −5, and −10



(B) Lotteries over prizes 0, −10, and −20



(C) Lotteries over prizes 0, −15, and −30



(D) Lotteries over prizes 0, −20, and −40

C.2. Estimated neural network

C.2.1. The parameters of the CARA neuron and the CE neurons

Recall that the selected architecture has one CARA neuron and two CE neurons in the behavioral layer. In the best-performed model, the CARA neuron has a reasonable level of α and the two CE thresholds coincide with each other.

α	β	η_1	η_2
2.456	1.268	0.900	0.900

C.2.2. The parameters of the RD neurons

Recall that the selected architecture has 20 RD neurons in the behavioral layer.

λ_i				
−0.2423	0.8690	0.9490	−1.1009	−1.8825
0.2890	−0.0348	−0.0213	−1.5899	0.3629
0.1764	−1.3350	−0.4814	0.2009	0.2528
0.3216	1.7527	−1.5367	−2.3024	0.1011
γ_i				
0.4846	0.5749	−1.0015	−0.7963	−0.1861
1.3081	−0.3849	0.6741	1.3114	−1.2115
−0.3528	0.5260	0.0001	1.4063	−0.3760
0.3216	0.5472	−0.2428	0.3262	0.5538
κ_i				
−0.9536	−1.3531	0.9147	1.3647	0.1489
0.7659	−0.6733	1.6707	−0.6014	−1.9913
−0.3619	1.5037	0.0000	−0.5946	−0.0216
−0.0675	0.7970	0.8573	1.5972	0.4117

C.2.3. The parameters of the additional hidden layer

Recall that there are 15 neurons in the additional hidden layer. Each neuron takes a weighted average of the outputs from the 23 neurons in the behavioral layer (1 CARA, 2 CE, 20 RD), adds a bias to the weighted average, compares the result with 0, and outputs the result if it is positive and zero otherwise. Finally, the neural network takes a weighted average of the outputs of the additional hidden layer to compute the utility.

(A) Weights of the neurons in the additional hidden layer

Neurons	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
CARA	0.0805	0.0414	0.0051	0.0527	−0.1095	0.1405	0.1098	−0.1145	0.0837	−0.0355	−0.0297	−0.1028	0.0359	−0.0062	0.0000
CE 1	0.1907	0.0580	−0.0504	0.1176	−0.2673	−0.0421	−0.0122	−0.0275	−0.1076	−0.0665	0.1114	−0.2454	−0.0114	−0.0124	0.0000
CE 2	0.1847	0.0574	−0.0493	0.1149	−0.2449	−0.0404	−0.0160	−0.0317	−0.0919	−0.0658	0.1103	−0.2542	−0.0114	−0.0124	0.0000
RD 1	0.3060	0.0119	−0.3884	−0.0148	0.4107	−0.0220	0.3257	0.1411	0.1138	0.2519	0.0831	0.1326	0.1811	0.0185	0.0000
RD 2	0.4945	−0.3647	0.4452	0.1405	−0.4141	−0.3331	0.0378	−0.1788	−0.3489	0.2901	0.3230	−0.3738	0.1355	−0.1132	0.0000
RD 3	0.5147	−0.0757	−0.2326	−0.3663	0.2318	0.5072	0.1885	−0.5404	0.5467	−0.2331	−0.3339	0.4594	−0.4719	0.0010	0.0000
RD 4	0.4272	−0.2009	−0.3472	0.1000	−0.0009	−0.5134	−0.0386	0.2090	−0.2290	0.0450	0.0782	−0.0684	−0.2758	−0.0157	0.0000
RD 5	0.4118	−0.1189	−0.2059	0.2274	0.2030	0.0292	0.1535	−0.5111	−0.2729	0.4797	0.0740	−0.4599	0.5172	−0.1880	0.0000
RD 6	0.0629	0.3150	0.2950	0.1520	0.1116	0.6450	−0.1722	0.4354	−0.1203	−0.4955	−0.4202	0.3021	−0.2969	−0.0494	0.0000
RD 7	0.1735	0.0578	0.1231	0.0855	−0.1555	0.2508	−0.0098	−0.1270	0.0291	−0.0099	−0.0649	−0.3075	−0.1023	−0.0103	0.0000
RD 8	0.0000	0.0438	−0.5856	0.0000	0.0000	0.1528	−0.1843	−0.3301	−0.3625	0.1492	0.0248	0.0000	−0.3362	0.0000	0.0000
RD 9	−0.2390	−0.3206	−0.0025	0.3392	0.2215	0.4892	−0.0435	−0.3375	0.5832	0.1674	0.3253	0.4327	0.0028	0.3023	0.0000
RD 10	0.0558	0.3048	0.1432	−0.2104	−0.0156	0.0192	0.1154	−0.3112	0.3688	0.0263	0.2398	−0.4356	−0.4952	−0.1173	0.0000
RD 11	−0.2496	0.0348	−0.2004	0.0988	0.2898	−0.1560	−0.0383	0.0583	−0.2610	0.1706	0.0692	0.1066	−0.3755	0.0178	0.0000
RD 12	0.1143	−0.1725	0.1802	0.3767	0.3508	0.4574	−0.1628	0.2614	0.2114	−0.2173	0.2202	0.4930	−0.4325	0.2204	0.0000
RD 13	0.1515	−0.0584	0.2740	−0.1107	0.2683	0.1448	0.0574	−0.2725	0.0123	0.3538	0.0159	0.2801	−0.4718	0.1183	0.0000
RD 14	0.1238	0.0914	0.4155	0.2640	0.2400	0.3715	0.2359	−0.2959	−0.2420	−0.0165	−0.3749	−0.2453	−0.2369	−0.0658	0.0000
RD 15	0.2558	0.0768	0.3759	0.3532	0.1872	−0.1390	0.4280	−0.2774	0.0919	0.2347	−0.0629	−0.3067	−0.1911	0.0350	0.0000
RD 16	−0.2149	0.0427	−0.2219	0.1392	−0.3732	0.3467	0.2825	0.4568	−0.1552	0.0120	−0.1138	0.1416	0.3476	0.0390	0.0000
RD 17	−0.4582	0.2030	−0.2305	0.0908	0.3685	0.0066	0.2739	0.5102	−0.0925	0.5253	−0.3564	0.4927	−0.1798	0.2292	0.0000
RD 18	−0.0832	0.4517	0.3768	0.4306	−0.4791	−0.2881	0.0849	0.1906	−0.3216	0.1398	0.1886	0.4300	0.2875	−0.0902	0.0000
RD 19	0.1987	0.0837	0.2144	0.5113	0.1070	−0.5514	−0.1198	0.0139	−0.3803	−0.1791	0.2126	−0.0962	0.4534	0.0605	0.0000
RD 20	0.1188	0.1721	−0.3292	−0.0625	−0.4791	0.0943	0.0926	−0.1723	−0.0756	0.0901	−0.2825	0.1629	0.2708	0.0054	0.0000

(B) Biases of the neurons in the additional hidden layer

Neurons	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Biases	0.0000	0.0099	−0.3195	0.0000	0.0000	−0.0448	−0.1017	−0.0994	0.0630	0.0886	−0.0056	0.0000	−0.3886	0.0000	0.0000

(C) Utility weights of the neurons in the additional hidden layer

Neurons	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Utility weights	−0.5437	−0.2091	−0.5713	−0.3341	0.7878	−0.6671	−0.7185	0.8277	−1.0413	0.2155	−0.2630	0.7289	0.6113	0.0344	0.0000

In the following table, we first report the weights each neuron assigns to the behavioral neurons, then the biases of each neuron, finally the weights the NU function assigns to each neuron in the additional hidden layer. Note that the Neuron 15 is never activated in the estimation process and thus redundant.

Data availability

Data will be made available on request.

References

Aczél, J., 1966. Lectures on Functional Equations and Their Applications. Academic Press, New York.

Aghdam, H.H., Heravi, E.J., 2017. Guide to Convolutional Neural Networks. Springer, New York.

Apesteguia, J., Ballester, M., 2018. Monotone stochastic choice models: The case of risk and time preferences. J. Political Econ. 126 (1), 74–106.

Battalio, R., Kagel, J., Jiranyakul, K., 1990. Testing between alternative models of choice under uncertainty: Some initial results. J. Risk Uncertain. 3 (1), 25–50.

Bernheim, B.D., Sprenger, C., 2020. On the empirical validity of cumulative prospect theory: Experimental evidence of rank-independent probability weighting. Econometrica 88 (4), 1363–1409.

Blavatskyy, P., Ortmann, A., Panchenko, V., 2022. On the experimental robustness of the Allais Paradox. Am. Econ. J.: Microeconomics 14 (1), 143–163.

Block, H.D., Marschak, J., 1960. Random orderings and stochastic theories of responses. In: Olkin, I., Ghurye, S., Hoeffding, W., Madow, W., Mann, H. (Eds.), Contributions to Probability and Statistics. Stanford University Press, pp. 97–132.

Bouchouicha, R., Wu, J., Vieider, F.M., 2023. Choice lists and 'standard patterns' of risk-taking. Working Paper.

Caplin, A., Martin, D., Marx, P., 2022. Modeling machine learning. Working Paper.

Chew, S.H., Miao, B., Shen, Q., Zhong, S., 2022. Multiple-switching behavior in choice-list elicitation of risk preference. J. Econom. Theory 204, 105510.

Chew, S.H., Waller, W.S., 1986. Empirical tests of weighted utility theory. J. Math. Psych. 30 (1), 55–72.

Cho, I.-K., Libgober, J.A., 2021. Algorithm games and rational play with strategic inference. Working Paper.

Choi, S., Fisman, R., Gale, D., Kariv, S., 2007. Consistency and heterogeneity of individual behavior under uncertainty. Am. Econ. Rev. 97 (5), 1921–1938.

Debreu, G., 1960. Review of "Individual choice behavior: A theoretical analysis" by R.D. Luce. Am. Econ. Rev. 50 (1), 186–188.

Dembo, A., Kariv, S., Polisson, M., Quah, J.K.-H., 2024. Ever since allais. Working Paper.

Echenique, F., Saito, K., 2019. General Luce model. Econom. Theory 68 (4), 811–826.

Erev, I., Ert, E., Plonsky, O., Cohen, D., Cohen, O., 2017. From anomalies to forecasts: Toward a descriptive model of decisions under risk, under ambiguity, and from experience. Psychol Rev 124 (4), 369–409.

Ert, E., Erev, I., 2013. On the descriptive value of loss aversion in decisions under risk: Six clarifications. Judgm. Decis. Mak. 8 (3), 214–235.

Fudenberg, D., Liang, A., 2019. Predicting and understanding initial play. Am. Econ. Rev. 109 (12), 4112–4141.

Fudenberg, D., Strzalecki, T., 2015. Dynamic logit with choice aversion. Econometrica 83 (2), 651–691.

Goldblum, M., Finzi, M., Rowan, K., Wilson, A.G., 2023. The No Free Lunch Theorem, Kolmogorov complexity, and the role of inductive biases in machine learning. arXiv preprint arXiv:2304.05366.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press, Cambridge, MA.

Goyal, A., Bengio, Y., 2022. Inductive biases for deep learning of higher-level cognition. Proc. R. Soc. A 478 (2266), 20210068.

Hahnloser, R., Sarpeshkar, R., Mahowald, M., Douglas, R., Seung, H., 2000. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. Nature 405, 947–951.

Harless, D., Camerer, C., 1994. The predictive utility of generalized expected utility theories. Econometrica 62 (6), 1251–1289.

Harrison, G.W., List, J.A., Towe, C., 2007. Naturally occurring preferences and exogenous laboratory experiments: A case study of risk aversion. Econometrica 75 (2), 433–458.

Hastie, T., Tibshirani, R., Friedman, J., 2009. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, New York.

He, J., Natenzon, P., 2024. Moderate utility. Am. Econ. Rev.: Insights 6 (2), 176–195.
Hertwig, R., Erev, I., 2009. The description–experience gap in risky choice. Trends Cogn. Sci. 13 (12), 517–523.
Kahneman, D., Tversky, A., 1979. Prospect theory: An analysis of decision under risk. Econometrica 47 (2), 263–292.
Ke, S., 2018. Rational expectation of mistakes and a measure of error-proneness. Theor. Econ. 13 (2), 527–552.
Ke, S., Wu, B., Zhao, C., 2024. Learning from a black box. J. Econom. Theory 221, 105886.
Ke, S., Zhao, C., 2024. From local utility to neural networks. J. Math. Econom. 113, 103003.
Kingma, D., Ba, J., 2017. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980v9.
Kovach, M., Tserenjigmid, G., 2022a. Behavioral foundations of nested stochastic choice and nested logit. J. Political Econ. 130 (9), 2411–2461.
Kovach, M., Tserenjigmid, G., 2022b. The focal Luce model. Am. Econ. J.: Microeconomics 14 (3), 378–413.
LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521, 436–444.
Lu, J., Saito, K., 2022. Mixed logit and pure characteristics models. Working Paper.
Luce, R.D., 1959. Individual Choice Behavior: A Theoretical Analysis. Wiley, New York.
McFadden, D., 1973. In: Zarembka, P. (Ed.), Frontiers in Econometrics. Academic Press, New York, pp. 105–142.
McGranaghan, C., Nielsen, K., O'Donoghue, T., Somerville, J., Sprenger, C.D., 2024. Distinguishing common ratio preferences from common ratio effects using paired valuation tasks. Am. Econ. Rev. 114 (2), 307–347.
Murdoch, W., Singh, C., Kumbier, K., Abbasi-Asl, R., Yu, B., 2019. Definitions, methods, and applications in interpretable machine learning. Proc. Natl. Acad. Sci. 116 (44), 22071–22080.
Neyshabur, B., Tomioka, R., Srebro, N., 2014. In search of the real inductive bias: On the role of implicit regularization in deep learning. arXiv preprint arXiv:1412.6614.
Noussair, C.N., Trautmann, S.T., van de Kuilen, G., 2014. Higher order risk attitudes, demographics, and financial decisions. Rev. Econ. Stud. 81 (1), 325–355.
Ovchinnikov, S., 2002. Max-min representation of piecewise linear functions. Contrib. Algebra Geom. 43 (1), 297–302.
Peterson, J.C., Bourgin, D.D., Agrawal, M., Reichman, D., Griffiths, T.L., 2021. Using large-scale experiments and machine learning to discover theories of human decision-making. Science 372 (6547), 1209–1214.
Plonsky, O., Apel, R., Ert, E., Tennenholtz, M., Bourgin, D., Peterson, J.C., Reichman, D., Griffiths, T.L., Russell, S.J., Carter, E.C., Cavanagh, J.F., Erev, I., 2019. Predicting human decisions with behavioral theories and machine learning. arXiv preprint arXiv:1904.06866.
Plonsky, O., Erev, I., Hazan, T., Tennenholtz, M., 2017. Psychological forest: Predicting human behavior. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence. AAAI-17, pp. 656–662.
Saito, K., 2018. Axiomatizations of the mixed logit model. Working Paper.
Schmidt, U., 1998. Axiomatic Utility Theory Under Risk: Non-Archimedean Representations and Application to Insurance Economics. Springer-Verlag Berlin Heidelberg.
Starmer, C., 2000. Developments in non-expected utility theory: The hunt for a descriptive theory of choice under risk. J. Econ. Lit. 38 (2), 332–382.
Train, K., 2003. Discrete Choice Methods with Simulation. Cambridge University Press.
Tversky, A., Kahneman, D., 1992. Advances in prospect theory: Cumulative representation of uncertainty. J. Risk Uncertain. 5 (4), 297–323.
Wu, G., Zhang, J., Abdellaoui, M., 2005. Testing prospect theories using probability tradeoff consistency. J. Risk Uncertain. 30 (2), 107–131.