Using LSTM and embedding matrix in risky choice Prediction

Yihang Luo
Department of Psychology
University of Warwick
Target Journal:NeurIPS

Abstract

Risk-based decision-making has always been an important topic in behavioural science and economics. Psychological theory models are good at explanation but lack predictive capabilities. Recently, many studies have attempted to use machine learning models combined with behavioural theory to improve the predictive capabilities of risky decision-making models and compete for model performance. This study is the first to use the neural network structure Long Short-Term Memory(LSTM) in risky decision prediction. We used the largest risky choice benchmark data set ,called 'choice 13k'[2, 28]. This study is divided into two parts; we first investigated the performance of LSTM with different data sizes. We found out it can benefit more from larger data sets and has similar performance as the previous best model on the largest full data set to date even before adjusting hyperparameters. LSTM also has a better explanatory ability than previous machine learning models. We suggest that future adjustments based on LSTM could help design better machine learning models and generate new behaviour theories in the decision-making field. Second, we investigated the numerical embedding matrices.We presented the 1 dimension embedding vector learnt from LSTM and the Principal Component Analysis(PCA)components of 20 dimension embedding vector. We reproduce the assumptions of Cumulative Prospect Theory(CPT) using the numerical embedding matrix and its PCA dimensions. We found that absolute value is also an important dimension in the risky choice context. We also discussed the limitations of this study and the possible future research.

1 Introduction and Background

1.1 Risky Choice and behaviour models

Risky decision-making is an important research area located at the crossroads of psychology and economics [14]. It covers a wide range of theoretical frameworks within both disciplines. Traditional economics assume people's decision-making is rational, and expect individuals to calculate expected utility for each option and will simply prefer options with higher expected utility[23]. However, research has revealed a set of choice behaviours that deviate from the expected utility framework. Notable examples includes Allais paradox[1] and Ellsberg paradox[5]. The discovery of these irrational behaviours in risky choice tasks has engendered many behavioural theories intended to explain the distinct deviations from Expected utility theory, led by prospect theory[18]. These models focus on different dimensions of cognitive bias and provide explanations of underlying cognitive mechanisms, significantly enhancing our understanding of human choices in risky choice situation[7].

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

1.2 The Absence of Predictive Capacity in Behavioral Models

However, many models in risky decision-making depend on retrospective analysis, they are good at explaining past behaviour in specific situations, but lack generalization ability on different data sets and the predictive capacity remains poor. Most of the phenomena behaviour theories are learnt in a specific type of experimental setting and depend on some preconditions. When a new experiment setting is introduced, the pre-conditions may no longer met and these phenomena can not be observed. Also, a new cognitive phenomenon could arise and this new phenomenon might lead to the opposite outcome. For example, CPT's two important contributions are overestimating rare events and loss aversion, but these findings can not be generalized in all task settings.

There are two main kinds of risky choice task settings. The first one is a description-based choice, which shows the probability and outcome for each gamble in two options. The second one is experience-based choice, which means the outcome and probability will not be given to participants and they need to learn information about each option by learning. Researchers observed over-estimate rare events in description-based choice and this became a part of CPT[37]. But in many experience-based decision task settings researchers observed that people underestimated the rare events[7], which is the opposite of the assumption in CPT. This is because people use very different strategies in experienced-based decision-making and description decision-making tasks. However this pre-condition was not mentioned in the original CPT paper as when they designed the CPT theory, they didn't know people's behaviour in a new experience-based choice context[37]. Also, meta-analysis studies about CPT observed no significant loss aversion effect in many participants[31]. This is also a new finding, and the pre-condition of loss aversion(why some people show loss aversion but and others do not) is still unknown. This was summarized in Erev et al.(2010)'s paper[7]. They think the reason why behaviour models lack prediction ability is most of these models focus on explaining different phenomena. There are so many cognitive phenomena with unknown pre-conditions, and these pre-conditions were not well studied as most of the behaviour theory was proposed based on several small data sets with the specific experimental setting, so it is hard to make predictions in any new content.

1.3 Risky choice competition and benchmark data set

To build theoretical models which can explain behaviour across contexts and improve predictability,recently, researchers have started to organize prediction competitions[7, 6, 29]. These competitions provide choice sets that cover a large range of cognitive bias dimension[6], focusing on comparing the prediction ability between models[11 dimension covers 14 main cognitive bias phenomenon]. These competitions introduced machine learning models in choice tasks to make better predictions[2]. Machine learning models are well known for their good prediction ability. Organizers of these competitions argue risky choice decision studies should not only focus on building behavioural models to explain the psychological phenomenon. The emergence of competition expanded behavioural decision research to the combination of machine learning and behaviour science. These competitions provide benchmarks to allow comparisons between models, so that researchers can make comparisons and build models with better prediction ability.

1.4 Choice 13k data set and Neural Network benchmark models

Among all benchmark competition data sets, the choice 13k data set[2, 28]is the current largest public human risky choice data set[30 times larger than the previous largest one: Choice Prediction Competition(CPC18)][29].CPC18 data set is much smaller than the common machine learning norms, which makes machine learning models trained on the CPC18 data set perform badly in the overall generalization test. Given the relative size of the choice 13k data set, the authors propose it as the new benchmark to test the precision of both theoretical and machine learning models. They provided a baseline score for researchers to challenge. Until now, there are three relevant papers using the Choice 13k data set and all of the optimal models in each paper are neural network models[2, 28, 33]. Neural Network models and their variants are the best prediction models in many tasks and won lots of competitions in different tasks in these years[34, 20, 38]. We summarized the best models and the key findings using Choice 13k data in the following paragraphs.

Cognitive Prior In the first choice 13k paper[2], the authors tested the various machine learning models on risky choice tasks and compared the Mean Squared Error(MSE). The optimal model is

what they call Sparse MLP with cognitive prior[MSE=0.0091]. They first used the gamble generator in Choice Prediction Competition 15[30] to get synthetic gambling sets and then used Best Estimation And Sampling Tool(BEAST)'s prediction[6] to generate the choice rate, as BEAST has the best prediction ability among all current behaviour models. They used this synthetic data in the MLP model to train the model parameter to create pre-knowledge of the model, which is called 'cognitive prior' in their paper. The model is an MLP model with a Sparse Evolutionary Training algorithm(SET)[22]. SET initialize the network as a sparse graph then randomly add new connections and prune small weights after the end of each epoch. This algorithm can improve the model performance when data is sparse. They found that using cognitive priors can improve the model performance and make the training faster, especially when the data set is sparse.

Structure constraint and data size In another later work, the same group tested the combination of neural networks and behaviour models such as neural prospect theory and neural heuristic theory[28] using another version of choice 13k data set(subset). The model with the best performance is the Mixture of Theories(MoT) neural network model with MSE=0.0113. This version used structure constraints in neural network models to force the model to have substructures such as utility function; probability weight function and heuristic. Generally, models with weaker structure constraint have better performance but it lacks psychological insight and explainability. MoT is highly flexible, using the weight of different utility functions and probability weight functions as the substructure of the neural network. These substructures were combined using the Mixture of Local Experts algorithm[16]. This paper also tested the influence of the data scale on model performance. They found out that compared with behavioural models, complex machine learning models will benefit more from the increase of data sets as these models require a large amount of data to train.

Collaboration and explainability Finally, the most recent paper using Choice 13k used BEAST-Net model[33] which is a neural network version of BEAST[6]. The best model is BEAST-Net(Manually segmented) with MSE=0.0111. This paper focused on the collaboration between machine learning models and psychological theoretical models, aiming to improve explainability. The previous papers treat machine learning models as independent entities, causing a competition between theory-based models and machine learning structures[33]. In this paper, Shoshan;Hazan and Plonsky make the behaviour model collaborate with the machine learning model, resulting in both the improvement in performance and explainability.

1.5 Using LSTM in risky choice prediction

Given the NN model's outstanding performance. It's worth trying other kinds of Neural network models on risky choice tasks as well. In this paper we use a recurrent neural network(RNN) called the Long Short-Term Memory model (LSTM) to predict choice using choice13k[15]. Sequence-based models such as LSTM perform well on natural language processing tasks[13, 12]. It is worth finding out whether these model structures can provide good model performance in another inference task Risky choice. Also, the previous paper shows data scale is an important factor for model performance. Generally more complex machine learning models will benefit more from the increase of data set[28].LSTM has a more complex structure than all previous models used in the Choice 13k dataset. We expect LSTM can benefit more than MLP/MoT neural network structure from the increasing data set size.

The design of RNN is to process each gamble outcome and the corresponding probability in sequence. The model design used in the previous paper[28, 2] used fully connected layers. They put all features in the model at the same time and all the neurons in each layer are fully connected with others, so they are harder to train and have a greater risk of overfitting. But if we simply process each gamble feature in sequence, the machine will not learn the interaction between each feature. RNN design can revisit the result of past features so each gamble feature's processed value is dependent on others. The problem with RNN is it has a huge gradient vanishing problem. Given the sequence design of RNN and the chain rule of the backpropagation algorithm, when the partial derivative is small, the product of many small values will be very close to zero. Then the network weight will not be updated efficiently.

LSTM can solve the gradient vanishing problem by inducing the gate mechanism. LSTM design added three gates(Input gate; forgot gate and output gate) into the RNN design. The input gate decides which information will be passed to the next state. The forgot gate decides which learned information

should be removed. The output gate decides the output given the current neuron state. The input gate only allows part of the output to pass to the next state. It doesn't allow a long product chain, which can prevent the gradient from vanishing. The forget gate can remove unnecessary information and reduce the risk of overfitting. The output gate will consider the current state to make sure it can filter important information to the next state. These three gate designs make LSTM suitable for processing sequence data[15].

1.6 LSTM in risky choice context and model explainability

The importance of explainability Performance is not the only thing that matters for machine learning models. Explainability is also important. Generally, neural network models are bad at explanation because the process is implicit and acts like a Black Box. This is one of the main criticisms of the NN model[3]. This is also Peterson et al.(2021) concern about the model used in choice 13k task[28]. Also, the main focus of the BEAST-Net model is not its performance, it's the improvement on explainability[33]. The Explainable AI principle(EAI) encourages researchers to build explainable models. Explainability provides the human understandable way to present the basis of the decision. This can help provide the basis for proving; verifying decisions and improving the theoretical model[24].

LSTM in risky choice LSTM's structure is similar to the dynamic decision-making process and the process is explainable. We presented an example of how the Bidirectional LSTM model processes one risky choice task(See Figure 1). Assume there is a risky choice with option A and option B, where option A has one certain outcome A1; and option B has 2 outcomes B1 B2. There are 6 gambling features. Value(A1); p(A1);value(B1); p(B1);value(B2);p(B2). The process of LSTM is shown in Figure 1. The model processes each input attribute in sequence. Each block with a red or green circle are different state of the processing. When the model is processing the V(A1). The current output of LSTM will be H[V(A1)], which represents the hidden state of V(A1). The hidden state in LSTM means the processed input feature. It is called hidden because it is an intermediate state during the whole process and will not be observed. To transform it into observable output which matches the model task (e.g. choice rate in this task), It has to be further processed by the output layer.

The bidirectional design will process the sequence from two directions: Start to End and end to Start. The special LSTM Gate design was located between each state to control which information will be passed to the next state. This is not included in Figure 1 as it will not affect the structure. We can see that finally the model will produce a Concatenated hidden state for each input attribute and each hidden state will be influenced by the process of other states, then These concatenated hidden states will be combined and put into the output layer to predict the choice rate. It is worth mentioning that it is possible to build multiple outputs for all hidden states, which are usually used in translation tasks. In this case, each output should mean the translated word. But in this risky choice task, we only have the final choice rate. So we can only extract a single output at the end of the sequence just like in sentiment analysis. The model's sequence-based design can reflect human's dynamic process of risky choice, in the dynamic risky choice model, people process each attribute step by step and transfer them to evidence, then sum up the evidence to make a choice[10]. In the LSTM model, each hidden state could be seen as evidence and the model processes the attribute step by step to collect evidence, after processing all gamble attributes, the model will combine the evidence to make a prediction.

We can also use the Attention mechanism to weigh each input's influence on the hidden state. The attention layer in LSTM gives different states different weights. For example, when processing the hidden state with the highest value in Gamble B is possible that the most important attribute is the value(B1) itself. The second important attribute is p(B1). Maybe we will not pay much attention to p(B2), as there is no strong relationship between these two attributes. The model can learn which gamble attributes are important and which are not by giving different weights for each input attribute in different process stages. This also matches the mechanism of how people process each gamble's outcome features. Eye tracking data in the dynamic study of the risky choice process shows people will process each gamble outcome in an uncertain sequence and pay more attention to some features and ignore others[10]. The LSTM model includes a large number of explainable features and can describe the process of inference in risky choice better than models such as MLP[2] and MoT[28], it has the same explainability as the CPT Neural Network[28]. So, it is possible that in the future, we can adjust the LSTM structure to create an explainable model in risky choice prediction tasks following the EAI principle.

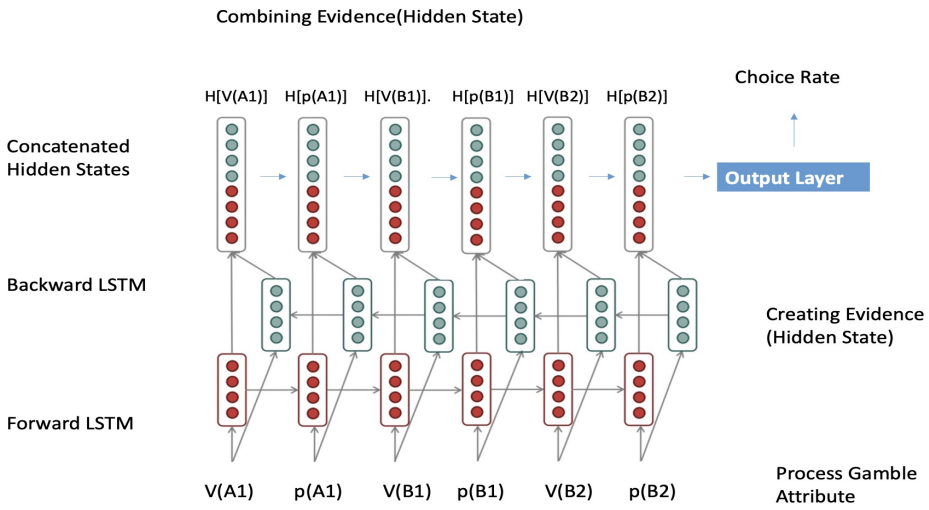


Figure 1: Example of LSTM model process for risky choice

1.7 Using Embedding matrix to explain subjective outcome and probability

Another benefit of LSTM is it can provide a new approach to describing subjective utility and subjective probability. CPT expects diminishing marginal utility and the over-estimate rare events and underestimated high probability in human risky choice[37]. Simulating the subjective function is a critical aspect of validating a model's consistency, for example, this pattern was also observed in sample-based models[35]. Peterson et al.(2021) used a neural network to infer the utility function and probability weight function and the observation is consistent with the assumption in CPT[28].

In this study, we are going to find the subjective outcome and subjective probability using what is known as the' embedding matrix', which presents a novel representation. The embedding matrix represents the meaning of each input using a vector format for each attribute. The Embedding matrix is the input value of the LSTM layer. The Embedding matrix could be both pre-trained or trained by the current model. When the Embedding matrix is pre-trained and frozen (No update), the embedding matrix is a dictionary of each attribute's value in vector format. Each attribute uses its index to find its vector value in the Embedding matrix and then this vector will be the input of the LSTM layer, which represents the target attribute. When the layer is not frozen, the Embedding matrix has the same use but it also will update the values for each vector in the matrix, which means the meaning of each attribute can be further adjusted and learned during the training process.

Encoding numbers into vectors is an existing method in Natural Language Processing(NLP) tasks. But in the NLP context, it comes with inherent challenges such as the infinite scope of numbers and the numerical meaning is highly dependent on context[17]. But in the risky choice context, these are not problems as this task has a limited numerical range and we only want to get embedding specifically in a risky choice context. Modern embedding techniques like Word2Vec and Glove are basic embedding methods to build pre-trained word vectors and embedding matrix[21, 27], but these methods can not be directly used in risky choice as they are based on co-occurrence probability. It depends on the assumption that each sentence makes sense so the co-occurrence probability is relevant to word meaning. However, co-occurrence probability doesn't have meaning in a risky choice context. In this case, we will use the same method used to get sentiment embedding[36]. We will not use the pre-trained embeddings and let the machine learn the vector of each input attribute. In this way, the generated value vector in the risky choice data set would include information about which pattern of the attribute will influence the choice rate.

There are two benefits of this method. Firstly, unlike the previous approach, which used values as inputs the machine learns the function based on input value. LSTM with an empty starting embedding matrix has zero knowledge about each number's meaning and will only learn the whole embedding vector based on the gambling choice rate. In past studies, these patterns were found by the utility and probability weight function fitting. The model gets the utility and subjective probability by

adjusting on original outcome and probability values[28]. The model only gives the utility function and probability weight function freedom to adjust the parameter. So all the bias from the expected utility theory will be forced to be mapped in the value and probability dimension. So, theoretically, we can not say the findings in CPT theory are because people underestimate the rare probability and the decreasing marginal utility. We can only say we observed these when we map the bias from the expected utility theory in the risky choice context to the utility and probability dimension.As in this design, we did not force all the biases should be mapped in utility and probability dimensions. Given the chain rule principle in the backpropagation algorithm, the weight in the embedding layer is the last one to be updated. If the bias is due to the interaction between each attribute, this bias will not be passed to the embedding layer. So If we found the same shape of subjective value and subjective probability as CPT assumed in the embedding matrices, this should support the CPT assumption more strongly. proves these biases from the expected utility theory are from utility and value dimensions and they happen before the further interaction between each attribute.

Secondly, the embedding matrix can be more than one dimension. The utility function and subjective probability are 1 dimension in previous studies. But value should not be the only dimension describing the outcome of risky choices. Other non-value features also influence an outcome. Factors such as simplicity[8]and the numerical superstitious also affect decision-making[25]. These features should not be included in the value dimension. It does not make sense To add simplicity and certainty in the subjective value or probability dimension. Because, the utility function and probability function should represent the scale of each value, simplicity and certainty mean whether a specific value is simple(59.001 and 59) or whether a specific value is certain(0.99 and 1). If we change one outcome's probability from 0.99 to 1, this could cause a huge change in preference. But this is not because 1 is 0.01 larger than 0.99. It's because of the introduction of certainty[1]. However, this information will be included in the utility function and subjective probability function after the model fitting as there is only one dimension describing value and probability. By splitting these patterns into multiple dimensions and examining the primary components of the embedding vector, we can get valuable insights into the subjective conception of value and probability in risky choices.

2 Methods

2.1 Data set

In this study, we used the Choice 13K data set[2, 28]. This data set comprises 13,006 choice tasks between Gambles A and B. The main attributes are the value and probability of each outcome and the average choice rate among all participants. To compare with a wider range of benchmark models, we tested two versions of the choice 13k data set. In Both data sets, the target is to predict the 'bRate', which is the rate of choosing option B.

Data set 1 The first version(See Figure 2 (a)) included all choice tasks[2]. In this version, all Gamble A has 2 outcomes [Ha and La] and all outcomes in Gamble B were encoded to 2 outcomes [Hb and Lb]. The original Gamble B has more than 2 outcomes[1-9 outcomes], and the model used in this version does not accept input attributes with uncertain length(e.g. MLP and Regression). So in this data set, the outcomes in gamble B were combined into 2 outcomes by calculating the expected value and the sum of probability. There are 6 features describing probability and outcome value(Ha, Hb, La, Lb,pHa,pHb). The Ha/Hb means the high value in gambles A and B. The La/Lb means the low value in gambles A and B. The pHa and pHb mean the probability of Ha and Hb. There are 4 outcome value features(Ha/Hb/La/Lb) and 2 probability features(pHa,pHb). This version also includes 6 other features describing the choice set: whether participants received feedback(Feedback 0/1); whether they can see the full description of each gamble (Amb 0/1)and a description of the original Gamble B [How many original outcomes (LotNumB 1-9)the distribution of outcome(LotShapeB 0-3], the block ID (block 1-5) and whether pay off in 2 gambles A and B are correlated(Corr -1,0,1). The key point of this version is each choice set has a set of features with the same type.

Data set 2 This version is a subset of the choice 13k data set(See Figure 2 (b)) and only includes the choice task when participants can see the feedback and the gamble in which each outcome was clearly described(feedback=1 and Amb=0 in data set1). There are a total of 9831 choice sets. This version is very simple, it only has the outcome value and the corresponding probability for each gamble[28, 33].In this version, each gamble has different numbers of outcomes and probabilities[1-9

for B and 1-2 for A]. Gamble features in Data Set 2 are the common input features of behaviour models so it is more suitable when behaviour theory is considered in the model[6]. That's why most of the recent papers are using Data Set 2 but not Data Set 1.[28, 33].

	Feedback	Block	Ha	pHa	La	Hb	pHb	Lb	LotShapeB	LotNumB	Amb	Corr	bRate
0	1	2	26	0.95	−1	23	0.05	21	0	1	0	0	0.626667
1	1	4	14	0.60	−18	8	0.25	−5	0	1	1	−1	0.493333
2	1	4	2	0.50	0	1	1.00	1	0	1	0	0	0.611765
3	1	3	37	0.05	8	87	0.25	−31	1	2	0	0	0.222222
4	0	1	26	1.00	26	45	0.75	−36	2	5	0	0	0.586667

(a) Data Set 1 Sample

	B	A
0	[[0.9500000000000001, 21.0], [0.05, 23.0]]	[[0.9500000000000001, 26.0], [0.05, −1.0]]
1	[[0.75, −5.0], [0.25, 8.0]]	[[0.6000000000000001, 14.0], [0.4, −18.0]]
2	[[1.0, 1.0]]	[[0.5, 2.0], [0.5, 0.0]]
3	[[0.75, −31.0], [0.125, 86.5], [0.125, 87.5]]	[[0.05, 37.0], [0.9500000000000001, 8.0]]
4	[[0.25, −36.0], [0.375, 41.0], [0.1875, 43.0],...	[[1.0, 26.0], [0.0, 26.0]]

(b) Data Set 2 Sample

Figure 2: Samples for features in Data Set 1 and Data Set 2

2.2 Pre-process

Encode Data Set 1 Each choice set has 12 features(see examples in Figure 1). Our initial step involved encoding all unique feature values to index. In the LSTM model, each unique input should be an index and the learnt features of each unique index will be recorded in an embedding matrix. In data set 1, there are 208 unique features. The most important two features are outcome value and probabilities. The range of value is[-50-118], there are 169 unique outcome values. The range of probabilities is [0.01-1], there are 11 unique probabilities. Other features have 28 unique values. We embedded each value with their column name to distinguish them, as 1 in Feedback 1 in probability and 1 in value have different meanings so they should not be mixed. We first recorded all unique values and the index in a unique value dictionary, then built a 208x1 and 208x20 empty embedding matrix(See Figure 3 for example).

Unique Value Dictionary

Original Attribute	Index
Outcome(Ha, Hb, La, Lb) =1	0
Outcome(Ha, Hb, La, Lb) =-10	1
Probability(pHa, pHb) =0.1	2
Probability(pHa, pHb) =1	3
Amb= 1	4
LotNumB =1	5
LotShapeB =1	6

Embedding Matrix

Index	1-d	1-d Embedding Prior	20-d
0	[0]	[1]	[0,0...,0]
1	[0]	[-10]	[0,0...,0]
2	[0]	[0.1]	[0,0...,0]
3	[0]	[1]	[0,0...,0]
4	[0]	[1]	[0,0...,0]
5	[0]	[1]	[0,0...,0]
6	[0]	[1]	[0,0...,0]

Figure 3: Unique Value Dictionary and Embedding matrix example for Data Set 1

Encode Data Set 2 We rounded probabilities to 0.01 and rounded all values to integers. This version of the data set includes values such as 53.0001. So data rounding was required. We also encoded 1 in value differently, so it will not be mixed with 1 in probability. As probability values are all between 0 and 1, we can distinguish the probabilities as outcome values are all larger than 1 or smaller than 0, and the probabilities are all between 0 and 1. There are 302 total unique values.The range of outcome values is [-50-256] , there are 258 unique outcome values. The range of probability is [0.01-1], there are 44 unique probability values. We labelled them with unique indexes and built a 302x1 dimension and 302x20 dimension empty embedding matrix.

Embedding Prior knowledge The embedding prior knowledge only appears in the 1-dimension matrix, which uses the target value's original value as the start value. In this case, the input values are the same as those used in the previous MLP model and MoT model(See Figure 2 for Example). We expect in this case, the model would have some useful start information about the meaning of each number.

EV Prior knowledge We used the gamble set in train sets to create Synthetic data. The choice rate of choosing option B('bRate') is created by the Expected Value(EV) function and exponential choice function. The formula of exponential choice function of choose option B is in (1).

$$p(B) = \frac{e^t \cdot V(B)}{e^t \cdot V(A) + e^t \cdot V(B)} \tag{1}$$

The V(A) and V(B) are expected value for option A and B. We used t=0.2, which is the common parameter for exponential choice function found in previous meta-analysis research[31].Small t means the influence of the expected value gap on the choice rate is mitigated and the choice rate for both options will be close to 0.5 unless the expected value gap is large. Then we trained the LSTM model using this data and loaded the trained EV model as the start point of the risky choice model. It's the same as the method used in the Cognitive Prior paper[2], the difference is they used BEAST model's prediction as Cognitive Prior and the Synthetic data is generated by gamble generator in CPC15 paper[6]. Our prior is Expected value and the gamble is just the same gamble in train sets. The main purpose is to make the model have some prior knowledge about each number's meaning and the approximate relationship between features and outcomes.

2.3 Model Architecture

The model structure for 2 data sets is the same. To find the model with the best performance, we should find the best hyperparameters for each model. The previous paper found hyperparameters from 20,000 combinations[2], but due to limited computation resources. We choose to use the same structure for 2 data sets and only find hyperparameters from a small range. The model structure is presented in Figure 4. The model has 4 layers, 1. Embedding layer 2. Bidirectional LSTM layer(size=64x2) 3. Attention layer: Linear layer (128,64) -tanh activation-Linear layer(64,1), followed by Drop out rate =0.2 4.Output layer: Linear layer(128,32)-PRelu activation function drop-out rate =0.5 -Linear layer (32,1) -Sigmond output.

The optimizer used is the Adam optimizer[19]. Using decreasing learning rate starts from either 0.01 or 0.001 and the decay rate is 0.9 for 0.001 and 0.7 for 0.01. Then we calculated MSE for 10%-100% data set size by splitting the original dataset into subsets with different sizes. So we can observe the influence of data size scale on model performance just as in previous papers[28]. For both data sets. The model chosen is LSTM with 1/20 dimension Embedding matrix with EV prior/ no EV prior / and 1 dimension Embedding matrix with Embedding prior. For data set1, we also replicated the MLP model[2] to make a better comparison. We tested 6 models for data set 1 and 5 models for data set 2.

2.4 Value Embedding

We extracted probabilities and outcome values in the embedding matrix vector in the trained LSTM non-prior models. As in non -the prior version, the model has zero knowledge about each attribute's meaning. It's worth exploring what will the machine learn about the meaning of each attribute in a risky choice context when it has no knowledge about each attribute. We first plotted the relationship between 1 dimension embedding and its original value for probability and value. To see if it is consistent with previous findings in[28] paper and the theoretical assumption in CPT[37]. We

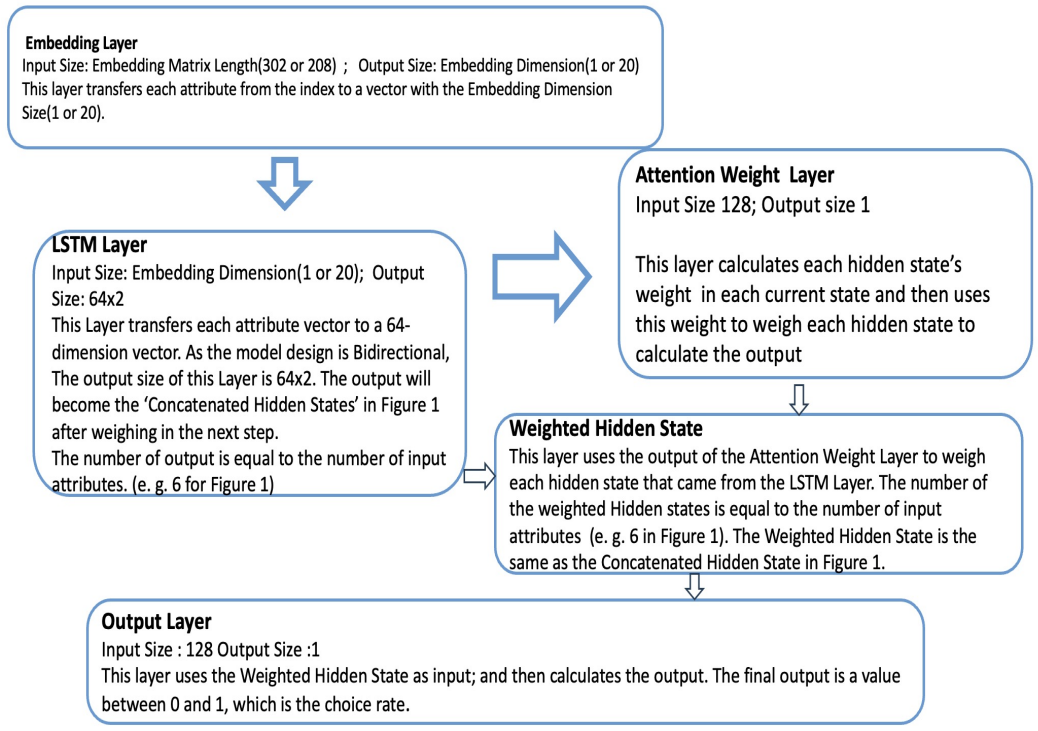


Figure 4: LSTM model Architecture

then tested 20-dimension embedding. Given that embedding vectors are trained in risky choice context, the main component of embedding vectors should reflect the subjective value and subjective probability. So we calculated the PCA component and presented the relationship between the first 2 PCA components with the original value of each unique input. We expect the first 2 PCA components can show some important features about value and probability in a risky choice context, as all previous utility functions and probability functions are limited to 1-dimension.

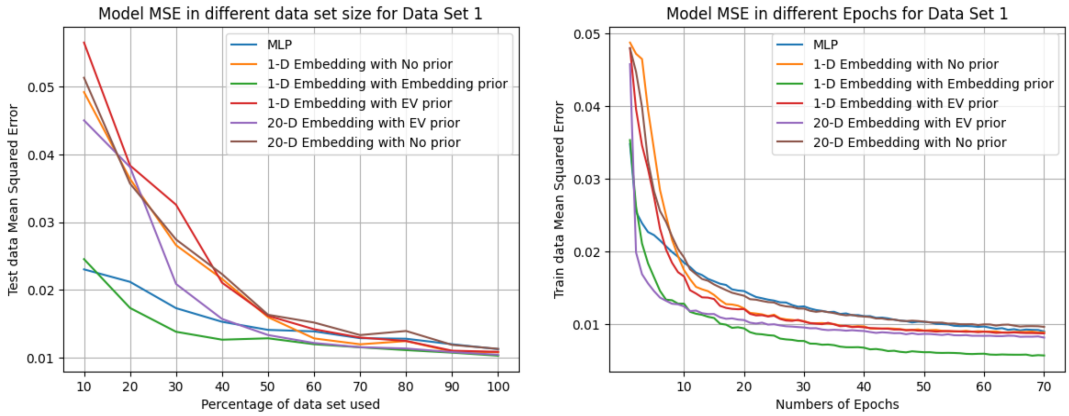
3 Results

3.1 Data Set 1

Table 1 shows the performance of our model and the benchmark models in cognitive prior paper[2]. All model result in this paper is the average MSE for 5 trained models with a different train-test split. There is no significant difference in MSE between these LSTM models except the LSTM-20 dimension without EV prior, which is a little bit worse than other LSTM models. We can see generally the models' performance is close to MLP with cognitive prior. And the best model is LSTM with Embedding prior(MSE =0.0100). This is slightly better than MLP with cognitive prior(MSE=0.0103). As the best benchmark model -Sparse MLP with cognitive prior used an extra sparse evolution method to improve the performance, but we did not use any extra tools to improve the performance. It seems reasonable to expect after some tuning, the LSTM model with embedding prior can win the current optimal MLP model in data set 1.

Model	MSE
MLP(benchmark model in [2])	0.0103
Sparse MLP(benchmark model in [2])	0.0091
LSTM 1-D non prior	0.0103
LSTM 1-D EV prior	0.0104
LSTM 1-D Embedding prior	0.0100
LSTM 20-D non prior	0.0113
LSTM 20-D EV prior	0.0104

Table 1: Test MSE on Choices13k data set 1



(a) Scale (b) Epoch

Figure 5: MSE in different Data size and Epochs for Data set 1

We tested the MSE change with the training data size,the results are shown in Figure 5(a). We can see that compared with the MLP model, the LSTM model's MSE(except LSTM with embedding prior) have a worse start but decreases fast and the final performance is close to MLP. This might be because when the data size is small there is not enough data to learn the meaning of each value, as each unique value was encoded to a vector and all embedding vectors start from 0. The LSTM model with embedding prior starts at a lower MSE when the data set is small and ends at a small MSE similar to MLP. EV prior didn't contribute to overall performance, but EV prior helped the model learn faster(Significant for the 20-dimension model and slightly for the 1-dimension model) as it already learned some information about the meanings of each value from EV synth data. (See Figure 5(b)).

3.2 Data Set 2

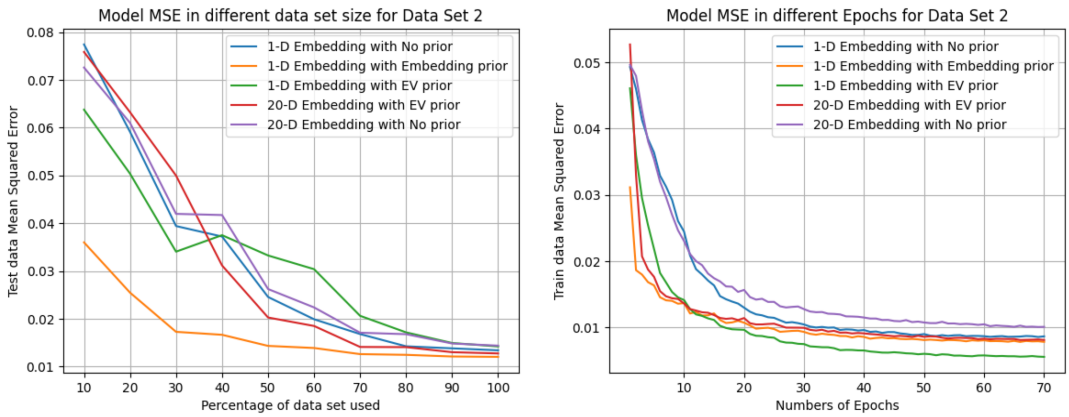
For Data Set 2, the model performance is presented with benchmark models in Table 2.Unlike models used in Data Set 1, which all have the same shape of input feature. In this version, all the models have uncertain numbers of input features, so the benchmark model in Data Set 1 such as MLP can not be used in Data Set 1 and compared with benchmark models in this part. But models in this part can be used in Data Set 1. As we mentioned before in the method part, Data Set 2 is a better version of Choice 13k and it is more suitable for Risky choice modelling and behaviour theory research. So the key studies and models about risky choice in large data sets are built and trained in this data set version. We can see that the LSTM-1 dimension models are close to the model with the best performance and perform a lot better than the CPT NN model. The best LSTM model is LSTM with embedding prior(MSE=0.0115). Generally, the LSTM-20-dimension models have worse performance than the LSTM-1-dimension model as they have a more complex structure and require more data to train, but the performance is still better than the CPT NN model.

Model	MSE
Neural Expected Utility(benchmark model in[28])	0.0217
Neural Prospect theory(benchmark model in[28])	0.0204
Neural CPT(benchmark model in[28])	0.0210
MoT (note: non-interpret able)(benchmark model in[28])	0.0113
BEAST-Net - Manually segmented(benchmark model in [33])	0.0111
LSTM 1-D non prior	0.0123
LSTM 1-D EV prior	0.0131
LSTM 1-D Embedding prior	0.0115
LSTM 20-D non prior	0.0144
LSTM 20-D EV prior	0.0126

Table 2: Test MSE on Choices13k data set 2

The influence of the data set size is shown in Figure 6 (a). We can see that at this time the LSTM with embedding prior still has a better start when the data size is small. We added the MSE Figure in different data set sizes for benchmark models in Peterson et al.(2021) paper[28] (See Figure 7). We can see that compared with benchmark models, LSTM models all have a very bad start when the data size is small (e.g. when the data set size is 50%, MSE of LSTM model are all between 0.02 and 0.04; except LSTM with embedding prior), which is a lot worse than all benchmark models in Figure 7(MSE around 0.02). But the LSTM model's MSE decreases a lot faster when the data size becomes larger. We did not find a significant influence of EV prior on model performance in all data size ranges. But the EV prior also significantly makes the model learn faster than the non-prior(See Figure 6(b)).

It's worth mentioning that though LSTM with embedding prior are best LSTM model in this stage, it has a limitation, which is the outcome value in the Embedding matrix will not be updated, as the scale of value (-50-256) is too large so they can not be updated for using small learning rate. We did not choose to normalize the value into smaller numbers as we found out this would damage the model performance. So, LSTM without embedding prior should have better potential as it can learn better subjective values. We think one important reason why the current LSTM with Embedding prior performance is better than LSTM without Embedding prior is the dataset did not provide enough samples to train a stable value vector for each input value. So we expect LSTM models without embedding prior can benefit more than previous benchmark models when the data size becomes larger and will perform better than LSTM with embedding prior. Given that LSTM with embedding prior already has close performance to the current optimal model, we expect LSTM will become the optimal model in this risky choice task after further tuning in data set 2.



(a) Scale (b) Epoch

Figure 6: MSE for different Data size and Epochs for Data Set 2

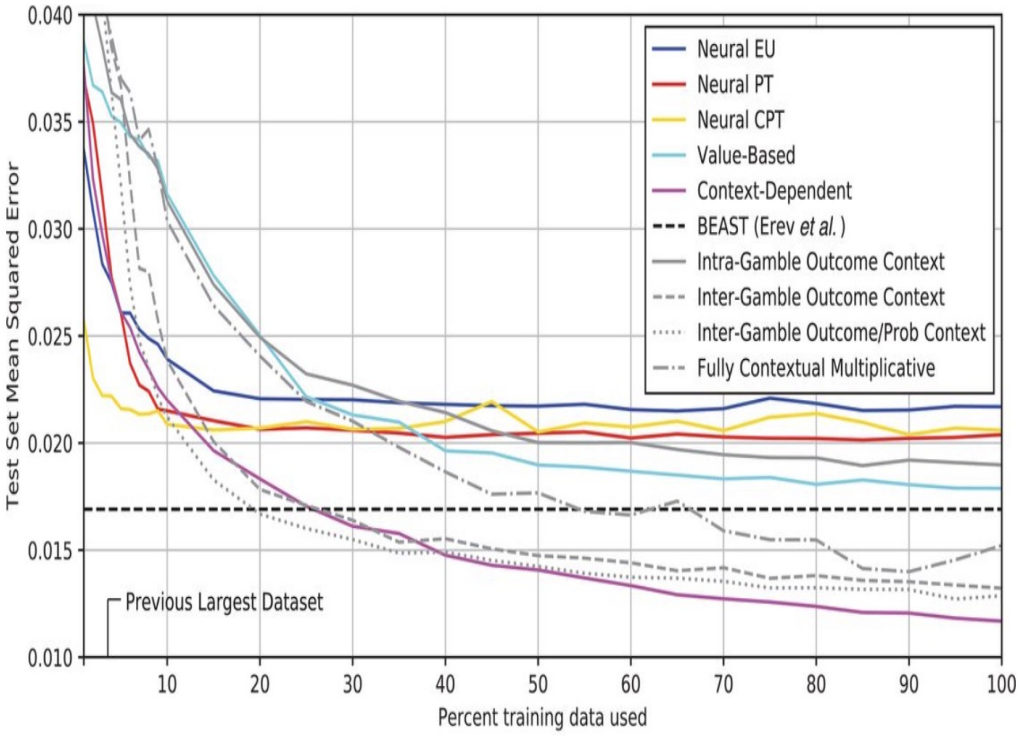


Figure 7: MSE for different Data size for benchmark models in Peterson et al.(2021) paper[28]

3.3 Value Embedding

We presented the value embedding learnt in LSTM 1-no prior and LSTM 20-no prior model in both data sets(Figure 8 for data set 1 and Figure 9 for data set 2) to see what kind of embedding vector would the model learn when the machine starts from 0 knowledge about each number's meaning. For some unique values, the number of samples is too small. So we only picked a subset of unique values with a larger number of samples. We presented outcome values from [-50-100] for both data sets. All probabilities in data set 1 and probabilities when the number of samples>1000 for data set 2.

The result shows that for both data sets. The values in 1-dimension embedding are consistent with PT theory[18]. The outcome value embedding is all S-shape, showing decreasing marginal utility for large values. As for probability, it shows overestimating small probability and underestimating large probability. It's worth mentioning that the value '1' in probability embedding is weird. That is not surprising as the PCA component of probability =1 does not only mean the value dimension. It also means the structure of the gamble is a lot different as one of the choice only have one certain outcome. 1 and 0 has special meaning and there should be a huge gap between 0 and 0.001 or 1 and 0.999. Just as the certainty effect assumes[1].

For the 20-dimension embedding, we calculated the first 2 PCA components. (Variance explained by PCA component for Value are: Data Set 1 PCA1=0.84 PCA2=0.06; Data Set 2 PCA 1 0.84 PCA2=0.06. Variances explained by the PCA component for Probability are Data Set 1 PCA1=0.88 PCA2=0.07; Data Set 2 PCA 1 0.77 PCA2=0.11.) For both data sets, the first PCA component for outcome and probability is similar to the CPT assumption just like in 1-dimension embedding. The second PCA component is all V-shaped. A natural interpretation is that the second most important attribute learnt in 20-dimension embedding is close to the absolute value of each subjective outcome and subjective probability. When we perform PCA on [X and absolute X], we will get a similar PCA component shape.For example, the two PCA components for [[1,1],[0,0],[-1,1]] is PCA1= [1,0,-1] and PCA2=[0.33,-0.66,0.33].

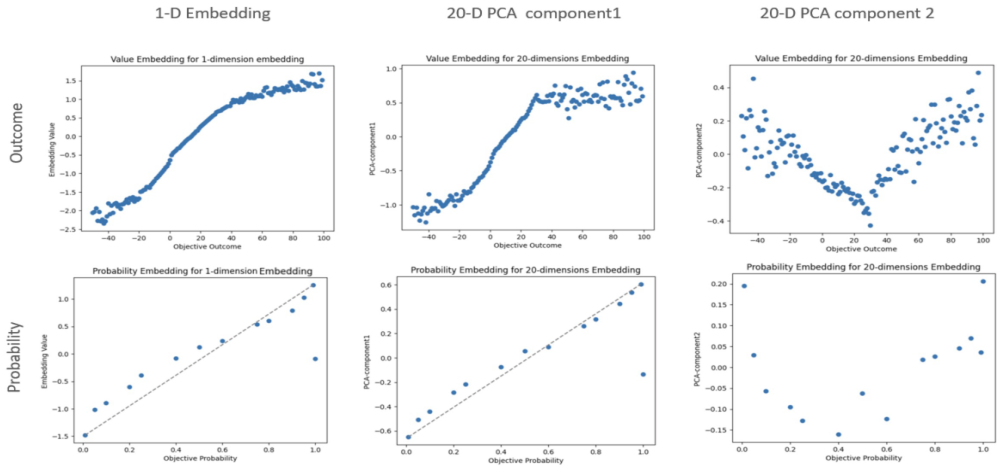


Figure 8: Embedding Results for Data set 1

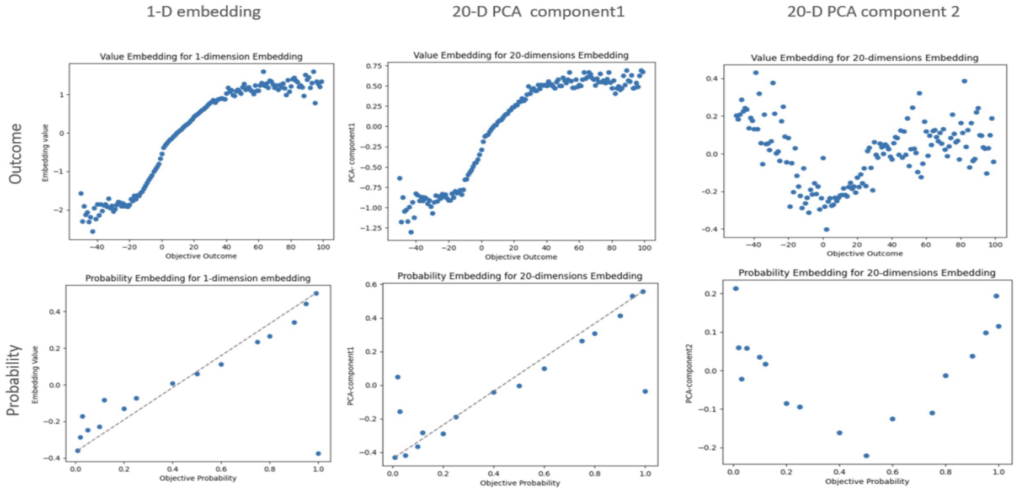


Figure 9: Embedding Results for Data set 2

4 Discussion

LSTM performance This is the first paper using LSTM in risky choice prediction and the performance is already close to the optimal performance even though we didn't adjust much structure and did not tune the model to find the best hyperparameter. Also, LSTM without embedding prior is very likely to benefit more from a larger data set and it has better potential than the current best LSTM model: LSTM with Embedding prior. This means when the risky choice set becomes larger in the future, these LSTM models are more likely to win over the current models. We expect after fine-tuning, we can build an optimal model with better performance than all current models using this approach. Apart from the benefits of LSTM design itself. Encoding unique values to index, then recording the value in the embedding matrix has potential benefits compared with using absolute value as input. Humans can only perceive the imprecise and biased representation[32]. So, when we want to combine behaviour explanation with the NN model. Using models that directly operate on outcome attributes might not be the best option. There will be a fundamental structural difference between the Neural Network model and the Biological network. The biological network in the human

brain will not treat the input gambling attribute as a precise value, it should be a feature just like words. LSTM with embedding matrix design is more close to a biological network and this may lead to better model performance.

Embedding matrix in risky choice We think the approach using an embedding matrix to describe the subjective value and subjective probability is worth further exploring. We found out the machine learnt the same subjective outcome and subjective probability shape as the theoretical assumption of CPT when it has zero prior knowledge about the meaning of each attribute. This is a strong support of CPT theory[18]. We observed the S-shape of the utility function in outcome embeddings and over-estimated rare events/ under-estimated large probabilities in probability embeddings. In this study, we trained the model to learn the embedding matrix from zero. the machine doesn't know the meaning of each number. It doesn't know if 90 is larger than 1 or 0.9 is a larger probability than 0.1. And it doesn't force the model to map the bias from the expected utility to the embedding matrix. However it still successfully learnt the subjective outcome and subjective probability patterns in the risky choice context. This means these biases from expected utility theory are very likely to come from the value dimension and probability dimension. The bias is not because of some unpredictable interactions between each attribute in later cognitive processing. Also, the PCA result of the 20-dimension embedding matrix suggests that Subjective Value should not be the only attribute for outcome and probability. Absolute values also need to be encoded in the behaviour model. We suggest trying both include value and absolute value in behaviour models, just as we both use rank and absolute value in range frequency theory[26].

Limitation The main limitation of this approach is it requires encoding unique values. So maybe it's not suitable for a very large range of input attributes. We need enough number (approximately 1000) of samples for each unique to learn a stable embedding vector matrix. the model lacks the generalization ability to new input values that are out of the train data input sample range. But the drawbacks are all relatively. This problem also happens in the traditional neural network used in previous Neural Network models using the Choice 13k dataset. consider the principle of the Linear activation function used in Fully Connected Neural Networks in these papers. Activation functions such as Relu[11] introduce non-linearity by assigning different linear local structures for each value range. When the input data is sparse, the traditional model will also lose accuracy and lack generalization ability for the same reason. So we think this is not a huge drawback as other models have a similar problem. It is possible that adding embedding prior trained in large synth data can help mitigate this limitation. Embedding prior can add a start embedding vector value close to the target embedding vector. The Embedding prior could be the original value of the target value for a 1-dimension vector or pre-trained embedding matrix using large synthetic data.After adding the embedding prior, the output embedding matrix will be stable even if there is no enough sample to train . This will add genalization ability to the model. As each value in embedding matrix start from a relatively correct position instead of zero. So , the model will be able to deal with out of sample new inputs. Also, in real work tasks. The attribute space is usually limited. Such as in the price optimization task, we usually only have very few price options in different price optimization tasks[9]. So we think the LSTM approach still have good genalization ability in many tasks.

Explainability We think changing the fundamental structure of the machine learning model is another approach to improve the explainability of the machine learning model in behaviour tasks such as risky choice prediction. Previous papers suggest that behaviour constraints on neural network structure can add explain ability[28]. However, this approach causes the performance to decrease compared with the non-constructive model with a lower explaining ability. BEAST -NeT successfully added explain ability without harm to model performance by changing the model structure[33]. So I think we don't have to treat the behaviour theory as a constraint in the machine learning model. The explained ability and performance are not a trade-off. LSTM is a representative of EAI as it increased the performance in some tasks because it learns from biological theory. We think understanding of behavioural model will help design machine learning models' fundamental structure. To build better machine learning models in behaviour tasks and better behaviour theory, we suggest we should change the fundamental structure of the current neural network under the EAI principle by combining it with elements in behaviour theory. This research shows LSTM structure is very close to the human dynamic choice model very well. So, adjusting sequence-based models such as LSTM to add explain ability and good performance in risky choice tasks could be a good start.

Future Research For future research, we need to adjust the structure and find better parameters. We expect this will make the LSTM model win all current models in performance. We should also try other sequence models, such as BERT[4]. BERT prevents embedding from learning positional information because it encodes positional information separately. Therefore it can provide better value embedding results. Moreover, its design is closer to human inference. We can further explore other properties of numerical embeddings, such as simplicity, and value ranking attributes. We hope researchers can generate a larger-scale data set than choice 13k, which includes enough samples for each value range in the future. This will benefit risky choice studies.

5 Conclusion

In conclusion, we first used the LSTM model in the risky choice task using the choice 13k dataset. The LSTM model is explainable in this context and has outstanding performance. We expect after tuning the hyperparameter and the structure it can win all current models. We also expect it will perform better in larger data sets. The main problem of this approach is the data set limitation we need a large data with enough samples of each unique value to train the model. Using embedding prior can mitigate this limitation. The value embedding study is also worth exploring. We replicated CPT assumptions of subjective value and subjective probability using an embedding matrix. We also found that absolute subjective value is important in risky choice prediction and we suggest that it should be encoded separately from the subjective value in behaviour models. We hope this study can lead to more studies using other kinds of EAI models or theories in risky choice and finally help generate explainable models with high performance and explore behaviour theory with complex structures.

References

[1] Maurice Allais. Le comportement de l'homme rationnel devant le risque: critique des postulats et axiomes de l'école américaine. *Econometrica: journal of the Econometric Society*, pages 503–546, 1953.

[2] David D Bourgin, Joshua C Peterson, Daniel Reichman, Stuart J Russell, and Thomas L Griffiths. Cognitive model priors for predicting human decisions. In *International conference on machine learning*, pages 5133–5141. PMLR, 2019.

[3] Davide Castelvecchi. Can we open the black box of ai? *Nature News*, 538(7623):20, 2016.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[5] Daniel Ellsberg. Risk, ambiguity, and the savage axioms. *The quarterly journal of economics*, 75(4):643–669, 1961.

[6] Ido Erev, Eyal Ert, Ori Plonsky, Doron Cohen, and Oded Cohen. From anomalies to forecasts: Toward a descriptive model of decisions under risk, under ambiguity, and from experience. *Psychological review*, 124(4):369, 2017.

[7] Ido Erev, Eyal Ert, Alvin E Roth, Ernan Haruvy, Stefan M Herzog, Robin Hau, Ralph Hertwig, Terrence Stewart, Robert West, and Christian Lebiere. A choice prediction competition: Choices from experience and from description. *Journal of Behavioral Decision Making*, 23(1):15–47, 2010.

[8] Ellen RK Evers, Yoel Inbar, and Marcel Zeelenberg. Set-fit effects in choice. *Journal of Experimental Psychology: General*, 143(2):504, 2014.

[9] Kris Johnson Ferreira, Bin Hong Alex Lee, and David Simchi-Levi. Analytics for an online retailer: Demand forecasting and price optimization. *Manufacturing & service operations management*, 18(1):69–88, 2016.

[10] Susann Fiedler and Andreas Glöckner. The dynamics of decision making in risky choice: An eye-tracking analysis. *Frontiers in psychology*, 3:335, 2012.

[11] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323. JMLR Workshop and Conference Proceedings, 2011.

[12] Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. Hybrid speech recognition with deep bidirectional lstm. In *2013 IEEE workshop on automatic speech recognition and understanding*, pages 273–278. IEEE, 2013.

[13] Emmanuele Grosicki and Haikal El Abed. Icdar 2009 handwriting recognition competition. In *2009 10th International Conference on Document Analysis and Recognition*, pages 1398–1402. IEEE, 2009.

[14] Reid Hastie and Robyn M Dawes. *Rational choice in an uncertain world: The psychology of judgment and decision making*. Sage Publications, 2009.

[15] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[16] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

[17] Chengyue Jiang, Zhonglin Nian, Kaihao Guo, Shanbo Chu, Yinggong Zhao, Libin Shen, and Kewei Tu. Learning numeral embeddings. *arXiv preprint arXiv:2001.00003*, 2019.

[18] Daniel Kahneman. Prospect theory: An analysis of decisions under risk. *Econometrica*, 47:278, 1979.

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

[20] Roux Ludovic, Racoceanu Daniel, Loménie Nicolas, Kulikova Maria, Irshad Humayun, Klossa Jacques, Capron Frédérique, Genestie Catherine, et al. Mitosis detection in breast cancer histological images an icpr 2012 contest. *Journal of pathology informatics*, 4(1):8, 2013.

[21] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[22] Decebal Constantin Mocanu, Elena Mocanu, Peter Stone, Phuong H Nguyen, Madeleine Gibescu, and Antonio Liotta. Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science. *Nature communications*, 9(1):2383, 2018.

[23] Philippe Mongin. Expected utility theory. 1998.

[24] Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Methods for interpreting and understanding deep neural networks. *Digital signal processing*, 73:1–15, 2018.

[25] Travis Ng, Terence Chong, and Xin Du. The value of superstitions. *Journal of Economic Psychology*, 31(3):293–309, 2010.

[26] Allen Parducci and Douglas H Wedell. The category effect with rating scales: number of categories, number of stimuli, and method of presentation. *Journal of Experimental Psychology: human perception and performance*, 12(4):496, 1986.

[27] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[28] Joshua C Peterson, David D Bourgin, Mayank Agrawal, Daniel Reichman, and Thomas L Griffiths. Using large-scale experiments and machine learning to discover theories of human decision-making. *Science*, 372(6547):1209–1214, 2021.

[29] Ori Plonsky, Reut Apel, Eyal Ert, Moshe Tennenholtz, David Bourgin, Joshua C Peterson, Daniel Reichman, Thomas L Griffiths, Stuart J Russell, Evan C Carter, et al. Predicting human decisions with behavioral theories and machine learning. *arXiv preprint arXiv:1904.06866*, 2019.

[30] Ori Plonsky, Ido Erev, Tamir Hazan, and Moshe Tennenholtz. Psychological forest: Predicting human behavior. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[31] Jörg Rieskamp. The probabilistic nature of preferential choice. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 34(6):1446, 2008.

[32] Aldo Rustichini, Katherine E Conen, Xinying Cai, and Camillo Padoa-Schioppa. Optimal coding and neuronal adaptation in economic decisions. *Nature communications*, 8(1):1208, 2017.

[33] Vered Shoshan, Tamir Hazan, and Ori Plonsky. Beast-net: Learning novel behavioral insights using a neural network adaptation of a behavioral model. Technical report, Center for Open Science, 2023.

[34] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The german traffic sign recognition benchmark: a multi-class classification competition. In *The 2011 international joint conference on neural networks*, pages 1453–1460. IEEE, 2011.

[35] Neil Stewart, Nick Chater, and Gordon DA Brown. Decision by sampling. *Cognitive psychology*, 53(1):1–26, 2006.

[36] Duyu Tang, Furu Wei, Bing Qin, Nan Yang, Ting Liu, and Ming Zhou. Sentiment embeddings with applications to sentiment analysis. *IEEE transactions on knowledge and data Engineering*, 28(2):496–509, 2015.

[37] Amos Tversky and Daniel Kahneman. Advances in prospect theory: Cumulative representation of uncertainty. *Journal of Risk and uncertainty*, 5:297–323, 1992.

[38] Fei Yin, Qiu-Feng Wang, Xu-Yao Zhang, and Cheng-Lin Liu. Icdar 2013 chinese handwriting recognition competition. In *2013 12th international conference on document analysis and recognition*, pages 1464–1470. IEEE, 2013.