Deep Neural Networks for Choice Analysis

by

Shenhao Wang

Submitted to the Department of Urban Studies and Planning
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Computer and Urban Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Feburary 2020

Author .....
Department of Urban Studies and Planning
September 13, 2019

Signed by .....
Jinhua Zhao
Edward H. and Joyce Linde Associate Professor
Department of Urban Studies and Planning
Supervisor

Accepted by .....
Jinhua Zhao
Edward H. and Joyce Linde Associate Professor
PhD Committee Chair
Department of Urban Studies and Planning

Deep Neural Networks for Choice Analysis

by

Shenhao Wang

Submitted to the Department of Urban Studies and Planning
on September 13, 2019, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Computer and Urban Science

Abstract

As deep neural networks (DNNs) outperform classical discrete choice models (DCMs) in many empirical studies, one pressing question is how to reconcile them in the context of choice analysis. So far researchers mainly compare their prediction accuracy, treating them as completely different modeling methods. However, DNNs and classical choice models are closely related and even complementary. This dissertation seeks to lay out a new foundation of using DNNs for choice analysis. It consists of three essays, which respectively tackle the issues of economic interpretation, architectural design, and robustness of DNNs by using classical utility theories.

Essay 1 demonstrates that DNNs can provide economic information as complete as the classical DCMs. The economic information includes choice predictions, choice probabilities, market shares, substitution patterns of alternatives, social welfare, probability derivatives, elasticities, marginal rates of substitution (MRS), and heterogeneous values of time (VOT). Unlike DCMs, DNNs can automatically learn the utility function and reveal behavioral patterns that are not prespecified by modelers. However, the economic information from DNNs can be unreliable because the automatic learning capacity is associated with three challenges: high sensitivity to hyperparameters, model non-identification, and local irregularity. To demonstrate the strength of DNNs as well as the three issues, I conduct an empirical experiment by applying the DNNs to a stated preference survey and discuss successively the full list of economic information extracted from the DNNs.

Essay 2 designs a particular DNN architecture with alternative-specific utility functions (ASU-DNN) by using prior behavioral knowledge. Theoretically, ASU-DNN reduces the estimation error of fully connected DNN (F-DNN) because of its lighter architecture and sparser connectivity, although the constraint of alternative-specific utility could cause ASU-DNN to exhibit a larger approximation error. Both ASU-DNN and F-DNN can be treated as special cases of DNN architecture design guided by utility connectivity graph (UCG). Empirically, ASU-DNN has 2-3% higher prediction accuracy than F-DNN. The alternative-specific connectivity constraint, as a domain-knowledge-based regularization method, is more effective than other regularization methods. This essay demonstrates that prior behavioral knowledge can be used to

guide the architecture design of DNN, to function as an effective domain-knowledge-based regularization method, and to improve both the interpretability and predictive power of DNNs in choice analysis.

Essay 3 designs a theory-based residual neural network (TB-ResNet) with a two-stage training procedure, which synthesizes decision-making theories and DNNs in a linear manner. Three instances of TB-ResNets based on choice modeling (CM-ResNets), prospect theory (PT-ResNets), and hyperbolic discounting (HD-ResNets) are designed. Empirically, compared to the decision-making theories, the three instances of TB-ResNets predict significantly better in the out-of-sample test and become more interpretable owing to the rich utility function augmented by DNNs. Compared to the DNNs, the TB-ResNets predict better because the decision-making theories aid in localizing and regularizing the DNN models. TB-ResNets also become more robust than DNNs because the decision-making theories stablize the local utility function and the input gradients. This essay demonstrates that it is both feasible and desirable to combine the handcrafted utility theory and automatic utility specification, with joint improvement in prediction, interpretation, and robustness.

Committee member: Stefanie Jegelka
Title: X-Consortium Career Development Associate Professor
Electrical Engineering and Computer Science

Committee member: Drazen Prelec
Title: Professor of Management Science and Economics
Sloan Management School, Department of Economics, and Department of Brain and Cognitive Science.

Supervisor: Jinhua Zhao
Title: Edward H. and Joyce Linde Associate Professor
Department of Urban Studies and Planning

Acknowledgments

I feel a great gratitude towards my advisor Jinhua Zhao. He guided me through every step in this PhD program. Jinhua provides me the freedom to explore different ideas. My research directions had two dramatic changes in the past five years: from a policy focus about five years ago, to a behavioral focus about two years ago, and to this current dissertation that strongly relies on machine learning models. This large interest shift would be impossible without Jinhua's full support. Jinhua is not only a great advisor academically, but also taught me the skills in management and communication. It is very rare for any supervisor to provide the consistent and reliable support as Jinhua does. Words are not enough to express my gratitude towards Jinhua's helps.

I am grateful for the concrete suggestions that Stefanie Jegelka gave for my dissertation. Stefanie suggested me to learn statistical learning theory, nonlinear and robust optimization about two years ago, taught me the network perspective in her class, and tested my skills about the DNN-related topics in my qualifying exams. Stefanie helped me to explore the machine learning path I did not know much about two and a half years ago. I feel so grateful that the best suggestions were given at an early stage of my dissertation writing.

I am grateful for Drazen Prelec's helps. As a behavioral economist, Drazen helps me with the behavioral perspectives in the dissertation. I took Drazen's behavioral economics class first time in 2015 and second time in 2017. It is the only class I took twice in MIT. Drazen helped me with a paper about risk and AV adoption, which I did not incorporate in this dissertation. I wish I could have incorporated more papers into this dissertation to show my gratefulness. It is always a pleasure to work with him.

I feel so lucky to be in MIT, with its nearly unique institutional culture encouraging creativity and interdisciplinary collaborations. This dissertation is a synthesis of diverse perspectives, and I fully understand that this synthesis would be impossible in most of the other institutions in the world. Besides the three committee members,

the faculties in MIT taught me great knowledge that contributes significantly to this dissertation. I learnt econometrics models from Jerry Hausman, demand modeling from Moshe Ben-Akiva, machine learning skills from the machine learning group in computer science, optimization skills from Bart Parys, statistical learning theory from Sasha Rakhlin, and a lot of knowledge in many other classes.

The micro-environment in JTL built by Jinhua is amazing. JTL members always provide very valuable and constructive comments on my studies. For my dissertation, I would like to thank Qingyi Wang for her consistent and effective modeling supports, particularly for the first essay. I am also grateful for the great comments, suggestions, and the proofreading helps from Nate Bailey, Joanna Moody, Mary Rose Fissinger, Jintai Li, Jeff Rosenblum, Fiona Tanuwidjaja, and many others I cannot name one by one.

The academic environment in DUSP provides a nearly unique opportunity for the birth of this dissertation because it allows the interaction of many diverse and even opposite views to interplay in one issue. I can also get great career, research, and life suggestions from the faculties in DUSP: Larry Susskind, David Hsu, and Larry Vale. I am also grateful for the daily supports from my classmates: Elise Harrington, Nick Kelly, and many others.

Lastly, I need to thank my mother and grandmother for their supports on my life. It is great that my mother did not force me to get married or to make big money in the past few years, but largely kept silence to allow me to pursue my interests. I know how difficult it is to refrain from taking actions, which is the virtue of self-control. The greatest thank-you is devoted to my father, who nurtured me to be a person with decent characters. I wish he can feel calm and ease in the heaven.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Choice Analysis. Choice analysis has been an enduring question in various fields of social science. In economics, individual choice based on utility theories functions as the foundation of micro- and behavioral economics [85, 49, 130, 94]. In marketing, consumers' individual choices constitute the revenues of all the companies. In the realm of policy analysis, a massive number of individual choices jointly contribute to the final political decisions through the modern political institution. In the field of urban transportation, there has been a long tradition of using choice modeling tools to analyze how individuals make decisions of travel modes, travel frequency, travel scheduling, destination and origin, and routing [119, 32, 11]. Traditionally, researchers examine these individual choices by relying on discrete choice models (DCMs) and broadly utility theories. With these methods, the research communities have obtained tremendous insights into how individual decision-making.

DNNs. Recently deep neural networks (DNNs) as one particular machine learning (ML) method have demonstrated their high prediction accuracy in many empirical studies, as well as their flexibility of accommodating various data types. Convolutional neural networks (CNNs) and recurrent neural networks (RNNs), as two spe-

cific types of DNNs, have been applied to image recognition, classification, natural language processing, machine translation, health care, or sentimental classification, showing higher model performance than traditional models [73, 76, 43]. The DNN models are gradually expanding its applications from the computer vision or pattern recognition that traditionally belong to the field of computer science to many fields in social science [61, 22, 24, 40, 102]. In the transportation field, DNNs have also been applied to traffic operations, infrastructure management and maintenance, transportation planning, environment and transport, safety and human behavior, and air, transit, rail, and freight operations [65]. Typically, researchers found that DNNs can outperform classical methods in terms of their empirical prediction accuracy [106, 98, 113, 48, 26]. Unlike the classical DCMs, DNNs can be built in a generic way without involving much domain-specific knowledge that has been accumulated in each research community for decades. However, even without much prior human knowledge inputs, the generic-purpose DNNs can still outperform domain-specific models in a massive number of applications.

DNNs' Power. The high prediction accuracy of DNNs is partially caused by their high approximation power. NNs with even one hidden layer are known as *universal approximator*, implying that even a shallow neural networks (SNNs) is asymptotically a universal approximator when the width becomes infinite [30, 59, 58]. Recently, this asymptotic perspective leads to a more non-asymptotic question, asking why depth is necessary for NNs as even SNNs are already universal approximators. It turns out that DNNs can approximate functions with an exponentially smaller number of neurons than SNNs in many settings [29, 109, 103]. With this extraordinary approximation power, DNNs can automatically learn effective representations without relying on domain-specific knowledge [76]. This capacity of automatic feature learning is in sharp contrast to the domain-specific methods that typically rely on domain experts' knowledge, which can be incomplete and lead to large misspecification errors.

DNNs' Weaknesses. However, to fully take advantage of the DNNs' approximation power in choice analysis, researchers have to face at least three different challenges, including the lack of interpretability, the choice of regularization and DNN architectures, and the lack of robustness. First, it is unclear how to interpret DNNs for behavioral and policy analysis, as typically done by the classical choice modeling methods. Compared to the prevalent focus on DNNs' prediction power, the interpretability of DNNs is relatively understudied. However, it would be unfair to claim DNNs as being "black-box" models, since recent studies have improved DNNs interpretability by using a sequential dual training method [55], instance-based methods [36, 117, 68, 88], gradient-based methods [107, 154, 5, 120], or simply visualizing the hidden layers of DNNs [150, 147]. In terms of choice analysis, researchers used DNNs to predict demand in many studies, but the connection between DNNs and economic information is very weak with only several studies touching upon the issue of computing elasticities and market shares by using DNNs [107, 15]. Second, to interpret DNNs for choice analysis, regularization and sparse architectural design of DNNs is an inevitable challenge, owing to the potentially large estimation error of DNNs caused by their high model complexity. The architectural design perspective presents both the strength and the weakness of DNNs. On the one side, most recent studies made progresses in prediction accuracy by creating innovative DNN architectures, such as AlexNet [73], GoogleNet [125], and ResNet [52] in computer vision. On the other side, the extraordinary flexibility in DNNs' architectural design creates difficulty for researchers to choose an architecture for a specific task at hand. As a result, many studies have developed methods to facilitate the architectural design [152, 7, 153] and the hyperparameter searching [16, 17]. In the choice analysis setting, while it is possible to apply many generic methods to DNNs, it is unclear how to adopt this architectural design perspective in a way related to utility theory. Third, another challenge to DNNs' interpretability is related to their robustness. DNNs have been found to be a "brittle" system since it is easy to create adversarial examples around some local points of DNNs [126]. Recently, researchers developed many effective adversarial attacks, including fast gradient sign methods, one-step target class methods,

and many others [44, 74]. To improve the robustness of DNN models, researchers also develop many adversarial training methods to address the challenges posed by the adversarial attacks [100, 74, 84]. Robustness is important for economic interpretation. It is because economic analysis in models typically relies on local information at certain small region or even one point of the whole input space.

Summary. Overall, DNNs present new opportunities for researchers in the choice analysis. The power of DNNs comes from its approximation power and its automatic representation learning capacity. However, the interpretability of DNNs in the context of choice analysis is relatively understudied. To make the economic information extracted from DNNs reliable, the modeling process inevitably involves the architectural design and the robustness of DNNs. As a comparison, the classical choice models do not face these challenges since the implicit architecture of classical models is simple and the local information is typically regular. Therefore, to fully take advantage of the high approximation power of DNNs, it is inevitable for us to provide perspectives into these challenges.

1.2 Dissertation Overview

Research Question. This dissertation discusses how to interpret DNNs for economic information in the context of choice analysis with a three-essay structure. Overall, the key research question in this dissertation is

- How to improve the interpretability and robustness of DNN-based choice analysis by using the perspective of utility theory?

Under this umbrella question, three essays have slightly different focuses. The first essay answers the question How to extract from DNNs a full list of economic information as obtained from the classical discrete choice models. Note that essay 1 targets the interpretability of DNNs, and I focus on only narrowly the economic informa-

tion, thus avoiding the generic discussion about interpretability [1]. Essay 1 illustrates the similarity between DNNs and classical DCMs, and based on this similarity, the essay demonstrates that it is feasible to derive a complete list of economic information from DNNs. However at the same time, I will elaborate on three challenges, including the high sensitivity to hyperparameters, model non-identification, and the local irregularity, involved in interpreting DNNs for economic information. The model non-identification issue is treated less a problem recently since local minima can still provide high prediction accuracy in the out-of-sample context. The first and the third challenges lead to the questions in Essays 2 and 3. Essay 2 explores the question how to design an interpretable DNN architecture by using utility theory, and essay 3 examines the question how to address the local irregularity of DNNs by using utility theory. Essay 2 and 3 relate to the generic discussions about DNNs' architectural design and robustness. In all three essays, I emphasize the interdisciplinary perspective between DNNs and utility theory, rather than simply using generic ML methods such as L_1 and L_2 penalties to improve DNNs. A generic model cannot work well for domain-specific problems, as powerfully demonstrated by No Free Lunch Theorem. Then the critical aspect is how to impose effective constraints on the generic-purpose models to improve the model performance. In choice analysis, the rich utility theory becomes a natural choice for this purpose. The synthesis of DNNs and utility theory is also quite convenient owing to the high similarity between the random utility maximization (RUM) framework and the DNNs. Specifically, while DNN models do not have clear structures and lack robustness, classical utility theories typically have well-defined structure, are highly interpretable and robust to adversarial attacks. Therefore, the synthesis of DNN perspectives and classical theories can provide mutual benefits to each other.

Urban Transportation. In all three essays, urban transportation cases are used as running examples. This is because the urban transportation field has a long tradition

[1] I made this choice owing to the ambiguity of the concept interpretability. As illustrated by Lipton (2016) [80], interpretability has many aspects, which could lead to inconsistent evaluations on the same model.

of using choice modeling to examine the important questions such as travel mode choice. The question of travel mode choice is important because the result is often the foundation for predicting the overall performance of the transportation system. In essay 3, I also incorporate prospect theory and hyperbolic discounting models, both of which are important behavioral models. While concrete transportation applications and behavioral models are used as examples in the three essays, the insights from the three essays are very generic to any choice analysis.

Knowledge Generation. In a broader sense, this dissertation tackles the issue of knowledge generation, concerning the tension between theory-driven and data-driven methods, or equivalently, domain-specific and generic-purpose knowledge. This tension sometimes leads to the question whether we really need domain-specific knowledge for prediction, and sometimes a reversed one how to use prediction-driven ML models to generate more insights for policy intervention. Classical methods heavily rely on experts' knowledge, researchers start with domain-specific knowledge and end with prediction. On the contrary, the other direction seems also possible. Researchers argue that a model that constantly predicts accurately must have captured something [135]. If so, researchers should learn from machines to provide insights into the questions we seek to answer. The tension between prediction-driven and theory-driven methods is not a zero-one dichotomy. In fact, domain-specific knowledge is always involved at even the starting point of DNN modeling. Convolutional layers and max pooling layers in CNNs are designed since modelers know the conditional independence properties of pixels in images; RNNs are designed since modelers know the time series structure. While many researchers praise the power of full automatic learning in DNNs [76, 14], some studies argue that models need to be handcrafted to certain extent and then let the model to learn from the data [78]. This synthetic perspective is adopted in this dissertation to enable a more dynamic interaction between generic-purpose DNNs and domain-specific knowledge, achieving better prediction performance, model interpretation, and robustness.

Chapter 2

Essay 1: Extracting Complete Economic Information for Interpretation

2.1 Introduction

Discrete choice models (DCMs) have been used to examine individual decision making for decades with wide applications to economics, marketing, and transportation [13, 130]. Recently, however, there is an emerging trend of using machine learning models, particularly deep neural networks (DNNs), to analyze individual decisions. DNNs have shown its predictive power across the broad fields of computer vision, natural language processing, and healthcare [76]. In the transportation field, DNNs also perform better than DCMs in predicting travel mode choice, automobile ownership, route choice, and many other specific tasks [95, 107, 144, 26, 27, 63]. However, the interpretability of DNNs is relatively understudied despite the recent progress. [108, 34, 150]. It remains unclear how to obtain reliable economic information from the DNNs in the context of travel choice analysis.

This study demonstrates that DNNs can provide economic information as complete as the classical DCMs, including choice predictions, choice probabilities, market

share, substitution patterns of alternatives, social welfare, probability derivatives, elasticities, marginal rates of substitution (MRS), and heterogeneous values of time (VOT). Using the estimated utility and choice probability functions in DNNs, we can compute choice probabilities, market share, substitution patterns of alternatives, and social welfare. Using the input gradients of choice probability functions, we can compute probability derivatives, elasticities, marginal rates of substitution (MRS), and heterogeneous values of time (VOT). The process of interpreting DNN for economic information is significantly different from the process of interpreting classical DCMs. The DNN interpretation relies on the *function* estimation of choice probabilities, rather than the *parameter* estimation as in classical DCMs. With the accurate estimation of choice probability functions in DNNs, it proves unnecessary to delve into individual parameters in order to extract the commonly used economic information. Moreover, DNNs can automatically learn utility functions and identify behavioral patterns that are not foreseen by modelers. Hence the DNN interpretation does not rely on the completeness of experts' prior knowledge, thus avoiding the misspecification problem. We demonstrated this method using one stated preference (SP) dataset of travel mode choice in Singapore, and this process of interpreting DNN for economic information can be applied to the other choice analysis contexts.

However, DNNs' power of automatic utility learning comes with three challenges: (1) high sensitivity to hyperparameters, (2) model non-identification, and (3) local irregularity. The first refers to the fact that the estimated DNNs are highly sensitive to the selection of hyperparameters that control the DNN complexity. The second refers to the fact that the optimization in the DNN training often identifies the local minima or saddle points rather than the global optimum, depending on the initialization of the DNN parameters. The third refers to the fact that DNNs have locally irregular patterns such as exploding gradients and the lack of monotonicity to the extent that certain choice behavior revealed by DNNs is not reasonable. The three challenges are embedded respectively in the statistical, optimization, and robustness discussions about DNNs. While all three challenges create difficulties in interpreting DNN models for economic information, our empirical experiment shows that even

simple hyperparameter searching and information aggregation can partially mitigate these issues. We present additional approaches to address these challenges by using better regularizations and DNN architectures, better optimization algorithms, and robust DNN training methods in the discussions section.

This study makes the following contributions. While some studies touched upon the issue of interpreting DNNs for economic information in the past, this study is the first to systematically discuss the complete list of economic information that can be obtained from DNNs. We point out the three challenges involved in this process and tie the three challenges to their theoretical roots. While we cannot fully address the three challenges in this study, we demonstrate the importance of using hyperparameter searching, repeated trainings, and information aggregation to improve the reliability of the economic information extracted from DNNs. The paper can be valuable practical guidance for transportation modelers and provides useful methodological benchmarks for future researchers to compare and improve.

The paper is structured as follows. Section 2 reviews the studies about DCMs, and DNNs concerning prediction, interpretability, sensitivity to hyperparameters, model non-identification, and local irregularity. Section 3 introduces the theory, models, and methods of computing economic information. Section 4 sets up the experiments, and Section 5 discusses the list of economic information obtained from the DNNs. Section 6 discusses potential solutions to the three challenges, and Section 7 concludes.

2.2 Literature Review

DCMs have been used for decades to analyze the choice of travel modes, travel frequency, travel scheduling, destination and origin, travel route, activities, location, car ownership, and many other decisions in the transportation field [12, 26, 11, 119, 32, 2]. While demand forecasting is important in these applications, all the economic information provides insights to guide policy interventions. For example, market shares can be computed from the DCMs to understand the market power of competing industries [130]. Elasticities of travel demand describe how effective it is to influence

travel behavior through the change of tolls or subsidies [118, 53]. VOT, as one important instance of MRS, can be used to measure the monetary gain of saved time after the improvement of a transportation system in a benefit-cost analysis [118, 119].

Recently researchers started to use machine learning models to analyze individual decisions. Karlaftis and Vlahogianni (2011) [65] summarized 86 studies in six transportation fields in which DNNs were applied. Researchers used DNNs to predict travel mode choice [26], car ownership [101], travel accidents [149], travelers' decision rules [132], driving behaviors [60], trip distribution [89], and traffic flows [104, 82, 142]. DNNs are also used to complement the smartphone-based survey [143], improve survey efficiency [115], and impute survey data [35]. In the studies that focus on prediction accuracy, researchers often compare many classifiers, including DNNs, support vector machines (SVM), decision trees (DT), random forests (RF), and DCMs, typically yielding the finding that DNNs and RF perform better than the classical DCMs [106, 98, 113, 48, 26]. In other fields, researchers also found the superior performance of DNNs in prediction compared to all the other machine learning (ML) classifiers [38, 72]. Besides high prediction power, DNNs are powerful due to its versatility, as they are able to accommodate various information formats such as images, videos, and text [76, 73, 61].

Since DNNs are often criticized as a "black-box" model, many resent studies have investigated how to improve its interpretability [34]. Researchers distilled knowledge from DNNs by re-training an interpretable model to fit the predicted soft labels of a DNN [55], visualizing hidden layers in convolutional neural networks [150, 147], using salience or attention maps to identify important inputs [80], computing input gradients with sensitivity analysis [5, 114, 120, 36], using instance-based methods to identify representative individuals for each class [1, 36, 117], or locally approximating functions to make models more interpretable [108]. In the transportation field, only a very small number of studies touched upon the interpretability issue of DNNs for the choice analysis. For example, researchers extracted the elasticity values from DNNs [107], ranked the importance of DNN input variables [48], or visualized the input-output relationship to improve the understanding of DNN models [15]. However, no

study has discussed systematically how to compute all the economic information from DNNs, and none have demonstrated the practical and theoretical challenges in the process of interpreting DNNs for economic information.

First, DNN performance is highly sensitive to the choice of hyperparameters, and choosing hyperparameters is essentially a statistical challenge of balancing approximation and estimation errors. The hyperparameters include architectural and regularization hyperparameters. For a standard feedforward DNN, the architectural hyperparameters include depth and width, and the regularization hyperparameters include the L_1 and L_2 penalty constants, training iterations, minibatch sizes, data augmentation, dropouts, early stopping, and others [43, 19, 73, 137, 148]. Both architectural and regularization hyperparameters control the complexity of DNNs: a DNN becomes more complex with deeper architectures and weaker regularizations, and becomes simpler with shallower architectures and stronger regularizations. From a statistical perspective, the model complexity is the key factor to balance the approximation and estimation errors. A complex model tends to have larger estimation errors and smaller approximation errors, and a simple model is the opposite. DNNs have very small approximation errors because it has been proven to be a *universal approximator* [59, 58, 30], which also leads to the large estimation error as an issue. The large estimation error in DNNs can be formally examined by using statistical learning theory [20, 138, 134, 139, 136]. Formally, the model complexity can be measured by the Vapnik-Chervonenkis (VC) dimension (v), which provides an upper bound on DNNs' estimation error (proof is available in Appendix I). Recently, progress has been made to provide a tighter upper bound on the estimation error of DNNs by using other methods [10, 3, 92, 42]. While the theoretical discussion is slightly involved, it is crucial to understand that selecting DNNs' hyperparameters is the same as selecting DNNs' model complexity, which balances between approximation and estimation errors. When either the approximation errors or the estimation errors are high, the overall DNN performance is low. In practice, it indicates that certain hyperparameter tuning is needed to select the DNN with low overall prediction error, which is the sum of the approximation and estimation errors.

Second, DNN models are not identifiable, because the empirical risk minimization (ERM) is non-convex with high dimensionality. Given the ERM being non-convex, the DNN training is highly sensitive to the initialization [51, 41]. With different initializations, the DNN model can end with local minima or saddle points, rather than the global optimum [43, 31]. For comparison, this issue does not happen in the classical multinomial logit (MNL) models, because the ERM of the MNL models is globally convex [21]. Decades ago, model non-identification was one reason why DNNs were discarded [76]. However, these days, researchers argue that some high quality local minima are also acceptable, and the global minimum in the training may be irrelevant since the global minimum tends to overfit [28]. Intuitively, this problem of model non-identification indicates that each training of DNNs can lead to very different models, even conditioned on the fixed hyperparameters and training samples. Interestingly, these trained DNNs may have very similar prediction performance, creating difficulties for researchers to choose the final model for interpretation.

Third, the choice probability functions in DNNs are locally irregular because their gradients can be exploding or the functions themselves are non-monotonic, both of which are discussed in the robust DNN framework. When the gradients of choice probability functions are exploding, it is very simple to find an adversarial input x', which is ϵ-close to the initial x ($||x' - x||_p \leq \epsilon$) but is wrongly predicted to be a label different from the initial x with high confidence. This type of system is not robust because they can be easily fooled by the adversarial example x'. In fact, it has been found that DNNs lack robustness [93, 126]. With even a small ϵ perturbation introduced to an input image x, DNNs label newly generated image x' to the wrong category with extremely high confidence, when the correct label should be the same as the initial input image x [126, 44]. Therefore, the lack of robustness in DNNs implies the locally irregular patterns of the choice probability functions and the gradients, which are the key information for DNN interpretation.

2.3 Model

2.3.1 DNNs for Choice Analysis

DNNs can be applied to choice analysis. Formally, let $s_k^*(x_i)$ denote the true probability of individual i choosing alternative k out of $[1, 2, ..., K]$ alternatives, with x_i denoting the input variables: $s_k^*(x_i) : R^d \rightarrow [0, 1]$. Individual i's choice $y_i \in \{0, 1\}^K$ is sampled from a multinomial random variable with $s_k^*(x_i)$ probability of choosing k. With DNNs applied to choice analysis, the choice probability function is:

$$s_k(x_i) = \frac{e^{V_{ik}}}{\sum_j e^{V_{ij}}} \tag{2.1}$$

in which V_{ij} and V_{ik} are the jth and kth inputs into the Softmax activation function of DNNs. V_{ik} takes the layer-by-layer form:

$$V_{ik} = (g_m^k \circ g_{m-1} ... \circ g_2 \circ g_1)(x_i) \tag{2.2}$$

where each $g_l(x) = ReLU(W_l x + b_l)$ is the composition of linear and rectified linear unit (ReLU) transformation; g_m^k represents the transformation of the last hidden layer into the utility of alternative k; and m is the total number of layers in a DNN. Figure 2-1 visualizes a DNN architecture with 25 input variables, 5 output alternatives, and 7 hidden layers. The grey nodes represent the input variables; the blue ones represent the hidden layers; and the red ones represent the Softmax activation function. The layer-by-layer architecture in Figure 2-1 reflects the compositional structure of Equation 2.2.

The inputs into the Softmax layers in DNNs can be treated as utilities, the same as those in the classical DCMs. This utility interpretation in DNNs is actually shown by the Lemma 2 in McFadden (1974) [85], which implies that the Softmax activation function is equivalent to a random utility term with Gumbel distribution under the random utility maximization (RUM) framework. Hence DNNs and MNL models are both under the RUM framework, and their difference only resides in the utility

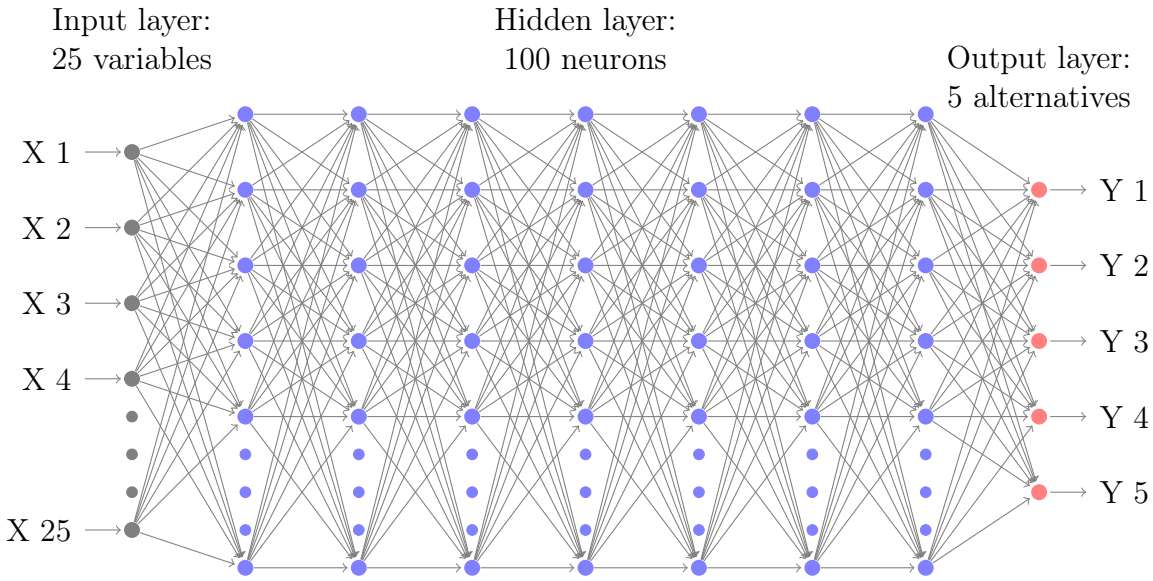


Figure 2-1: A DNN architecture (7 hidden layers * 100 neurons)

specifications. In other words, the inputs into the last Softmax activation function of DNNs can be interpreted as utilities; the outputs from the Softmax activation function are choice probabilities; the transformation before this Softmax function can be seen as a process of specifying utility functions; and the Softmax activation function can be seen as a process of comparing utility values.

DNNs are a much more generic model family than MNL models, and this relationship can be understood from various mathematical perspectives. The *universal approximator theorem* developed in the 1990s indicates that a neural network with only one hidden layer is asymptotically a universal approximator when the width becomes infinite [30, 59, 58]. Recently this asymptotic perspective leads to a more non-asymptotic question, asking why depth is necessary when a wide and shallow neural network is powerful enough. It has been shown that DNNs can approximate functions with an exponentially smaller number of neurons than a shallow neural network in many settings [29, 109, 103]. In other words, DNNs can be treated as an efficient universal approximator, thus being much more generic than the MNL model, which is a shallow neural network with zero hidden layers. However, from the perspective of statistical learning theory, a more generic model family leads to both smaller approximation errors and large estimation errors. Since the out-of-sample

prediction error equals to the sum of the approximation and estimation errors, DNNs do not necessarily outperform MNL models from a theoretical perspective. The major challenge of DNNs is its large estimation error, which is caused by its extraordinary approximation power. A brief theoretical proof about the large estimation error of DNNs is available in Appendix I. The proof is Appendix I uses Dudley integral and covering argument, which establishes the key intuition about the tradeoff between model complexity and sample size. I also provide a second proof in Appendix II, which focuses on the estimation error of the choice probability functions. The proof in Appendix II uses contraction inequality to generates tighter bound. Appendix II shows how the choice probability functions behave in a non-asymptotic manner. More detailed discussions are available in the recent studies from the field of statistical learning theory [136, 139, 42, 92, 10, 77, 9]. For the purpose of this study, it is important to know that the hyperparameter searching is essentially about the control of model complexity, which balances the approximation and estimation errors. This tradeoff between the approximation and estimation errors has a deep foundation in the statistical learning theory discussions.

2.3.2 Computing Economic Information From DNNs

The utility interpretation in DNNs enables us to derive all the economic information traditionally obtained from DCMs. With $\hat{V}_k(x_i)$ denoting the estimated utility of alternative k and $\hat{s}_k(x_i)$ the estimated choice probability function, Table 2.1 summarizes the formula of computing the economic information, which is sorted into two categories. Choice probabilities, choice predictions, market share, substitution patterns, and social welfare are derived by using functions (either choice probability or utility functions). Probability derivatives, elasticities, MRS, and VOTs are derived from the gradients of choice probability functions. This differentiation is owing to the the different theoretical properties between functions and their gradients [1]. The two categories also relate to different generic methods of interpreting DNNs, as discussed

[1]The uniform convergence proof is possible for the estimated functions, while it is much harder for the gradients because the estimated functions may not be even differentiable.

in our results section.

Economic Information	**Formula in DNNs**	**Categories**
Choice probability	$\hat{s}_k(x_i)$	F
Choice prediction	$\underset{k}{\text{argmax}}\ \hat{s}_k(x_i)$	F
Market share	$\sum_i \hat{s}_k(x_i)$	F
Substitution pattern between alternatives k_1 and k_2	$\hat{s}_{k_1}(x_i)/\hat{s}_{k_2}(x_i)$	F
Social welfare	$\sum_i \frac{1}{\alpha_i} \log(\sum_{j=1}^{J} e^{\hat{V}_{ij}}) + C$	F
Change of social welfare	$\sum_i \frac{1}{\alpha_i} \big[\log(\sum_{j=1}^{J} e^{\hat{V}^1_{ij}}) - \log(\sum_{j=1}^{J} e^{\hat{V}^0_{ij}}) \big]$	F
Probability derivative of alternative k w.r.t. x_{ij}	$\partial \hat{s}_k(x_i)/\partial x_{ij}$	GF
Elasticity of alternative k w.r.t. x_{ij}	$\partial \hat{s}_k(x_i)/\partial x_{ij} \times x_{ij}/\hat{s}_k(x_i)$	GF
Marginal rate of substitution between x_{ij_1} and x_{ij_2}	$-\frac{\partial \hat{s}_k(x_i)/\partial x_{ij_1}}{\partial \hat{s}_k(x_i)/\partial x_{ij_2}}$	GF
VOT (x_{ij_1} is time and x_{ij_2} is monetary value)	$-\frac{\partial \hat{s}_k(x_i)/\partial x_{ij_1}}{\partial \hat{s}_k(x_i)/\partial x_{ij_2}}$	GF

Table 2.1: Formula to compute economic information from DNNs; F stands for function, GF stands for the gradients of functions.

This process of interpreting economic information from DNNs is significantly different from the classical DCMs for the following reasons. In DNNs, the economic information is directly computed by using *functions* $\hat{s}_k(x_i)$ and $\hat{V}_k(x_i)$, rather than individual *parameters* $\hat{w}$. This focus on functions rather than individual parameters is inevitable owing to the fact that a simple DNN can easily have thousands of individual parameters. This focus is also consistent with the interpretation studies about DNNs: a large number of recent studies used the function estimators for interpretation, while none focused on individual neurons/parameters [88, 55, 5, 110]. In other words, the DNN interpretation can be seen as an end-to-end mechanism without involving the individual parameters as an intermediate process. In addition, the interpretation of DNNs is a prediction-driven process: the economic information is generated in a post-hoc manner after a model is trained to be highly predictive. This prediction-driven interpretation takes advantage of DNNs' capacity of automatic feature learning, and it is also in contrast to the classical DCMs that rely on hand-

crafted utility functions. This prediction-driven interpretation is based on the belief that "when predictive quality is (consistently) high, some structure must have been found" [90].

2.4 Setup of Experiments

2.4.1 Hyperparameter Training

Random searching is used to explore a pre-specified hyperparameter space to identify the DNN hyperparameters with the highest prediction accuracy [16]. The hyperparameter space consists of the architectural hyperparameters, including the depth and width of DNNs; and the regularization hyperparameters, including L_1 and L_2 penalty constants, and dropout rates. 100 sets of hyperparameters are randomly generated for comparison. The details of the hyperparameter space is available in Appendix III. Besides the hyperparameters varying across the 100 models, all the DNN models share certain fixed components, including ReLU activation functions in the hidden layers, Softmax activation function in the last layer, Gloret initialization, and Adam optimization, following the standard practice [43, 47]. Formally, the hyperparameter searching is formulated as

$$\hat{w}_h = \underset{w_h \in \{w_h^{(1)}, w_h^{(2)}, \ldots, w_h^{(S)}\}}{\operatorname{argmin}} \underset{w}{\operatorname{argmin}}\ L(w, w_h) \tag{2.3}$$

where $L(w, w_h)$ is the empirical risk function that the DNN training aims to minimize, w represents the parameters in a DNN architecture, w_h represents the hyperparameter, $w_h^{(s)}$ represents one group of hyperparameters randomly sampled from the hyperparameter space, and $\hat{w}_h$ is the chosen hyperparameter used for in-depth economic interpretation. Besides the random searching, other approaches can be used for hyperparameter training, such as reinforcement learning or Bayesian methods, [122, 152], which are beyond the scope of our study.

2.4.2 Training with Fixed Hyperparameters

After the hyperparameter searching, we examine one group of hyperparameters that lead to the highest prediction accuracy. Then by using the same training set and the fixed group of hyperparameters, we train the DNN models another 100 times to observe whether different trainings lead to differences in choice probability functions and other economic information. Note that the 100 hyperparameter searches introduced in the previous subsection provide evidence about the sensitivity of DNNs to hyperparameters, while the 100 trainings here conditioned on the fixed hyperparameters are designed to demonstrate the model non-identification challenge. Each training seeks to minimize the empirical risk conditioned on the fixed hyperparameters, formulated as following.

$$\min_{w} L(w, \hat{w}_h) = \min_{w} -\frac{1}{N}\sum_{i=1}^{N} l(y_i, s_k(x_i; w, \hat{w}_h)) + \gamma||w||_p \tag{2.4}$$

where w represents the parameters; $\hat{w}_h$ represents the best hyperparameters; $l()$ is the loss function, typically the cross-entropy loss function; and N is the sample size. $\gamma||w||_p$ represents L_p penalty ($||w||_p = (\sum_j (w_j)^p)^{\frac{1}{p}}$), and L_1 (LASSO) and L_2 (Ridge) penalties are the two specific cases of L_p penalties. Note that DNNs have the model non-identification challenge because the objective function in Equation 2.4 is not globally convex. DNNs have the local irregularity challenge because this optimization over the *global* prediction risks is insufficient to guarantee the *local* fidelity. The two issues are caused by related but slightly different reasons.

2.4.3 Dataset

Our experiments use a stated preference (SP) survey conducted in Singapore in July 2017. In total, $2,073$ respondents participated, and each responded to seven choice scenarios that varied in the availability and attributes of the travel mode alternatives. The final dataset with a complete set of alternatives included $8,418$ observations. The choice variable y is travel mode choice, including five alternatives: walking,

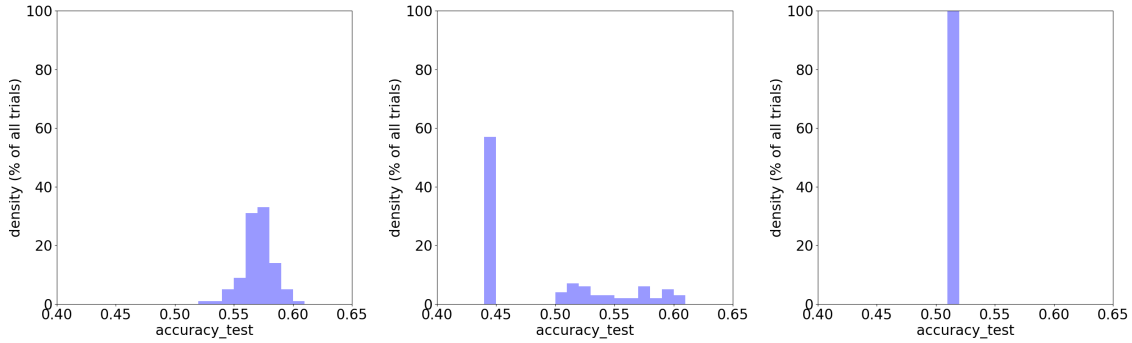
taking public transit, ride sharing, using an autonomous vehicle, and driving. The explanatory variables include 25 individual-specific and alternative-specific variables, such as income, education, gender, driving costs, and driving time. The dataset is split into the training, validation, and testing sets, with a ratio of 6:2:2, associated with 5,050:1,684:1,684 observations for each. The training set was used for training individual models; the validation set for selecting hyperparameters; the testing set for the final analysis of economic information.

2.5 Experimental Results

This section shows that it is feasible to extract all the economic information from DNNs without involving individual parameters, and that by using the hyperparameter searching and ensemble methods, it is possible to partially mitigate the three problems involved in the DNN interpretation. We will first present the results about prediction accuracy, then the function-based interpretation for choice probabilities, substitution patterns of alternatives, market share, and social welfare, and lastly the gradient-based interpretation for probability derivatives, elasticities, VOT, and heterogeneous preferences. This section focuses on one group of DNN models with five hidden layers and fixed hyperparameters (5L-DNNs), chosen from the hyperparameter searching thanks to their highest prediction accuracy. Note that the 5L-DNNs are chosen based on our hyperparameter searching results using this particular dataset, and this does not at all suggest that this specific architecture is generally the best in the other cases. The 5L-DNNs are compared to two benchmark model groups: (1) the 100 DNN models randomly searched in the pre-specified hyperparameter space (HP-DNNs), and (2) the classical MNL models with linear utility specifications. While it is possible to enrich the linear specifications in the MNL model, it is beyond the scope of this study to explore the different types of MNL models.

2.5.1 Prediction Accuracy of Three Model Groups

The comparison of the three model groups in Figure 2-2 reveals two findings. First, 5L-DNNs on average outperform the MNL models by about 5-8 percentage points in terms of the prediction accuracy, as shown by the difference between Figure 2-2a and 2-2c. This result that DNNs outperform MNL models is consistent with previous studies [107, 95, 65]. Second, choosing the correct hyperparameter plays a critical role in improving the model performance of DNNs, as shown by the higher prediction accuracy of the 5L-DNNs than the HP-DNNs. With higher predictive performance, the 5L-DNNs are more likely to reveal valuable economic information than the MNL models and the HP-DNNs.



(a) 5L-DNNs (b) HP-DNNs (c) MNL

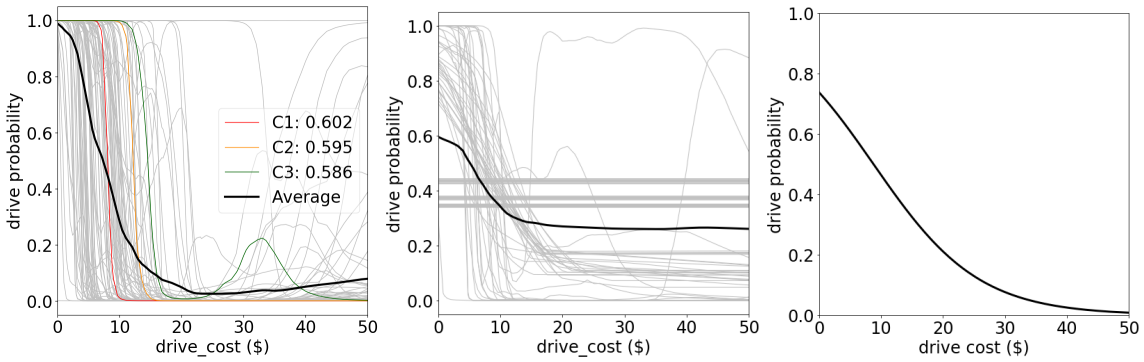
Figure 2-2: Histograms of the prediction accuracy of three model groups (100 trainings for each model group)

2.5.2 Function-Based Interpretation

Choice Probability Functions

The choice probability functions of the three model groups are visualized in Figure 2-3. Since the inputs of the choice probability functions $s(x)$ have high dimensions, the $s(x)$ is visualized by computing the driving probability with varying only the driving cost, holding all the other variables constant at the sample mean. Each light grey curve in Figures 2-3a-2-3b represents one individual training result, and the dark

curve is the ensemble of all 100 models. In Figure 2-3c, only one training result is visualized because the MNL training has no variation.



(a) 5L-DNNs (b) HP-DNNs (c) MNL

Figure 2-3: Driving probability functions with driving costs (100 trainings for each model group)

The results of the 5L-DNNs in Figure 2-3a demonstrate the power of DNNs being able to automatically learn the choice probability functions. From a behavioral perspective, the majority of the choice probability functions in Figure 2-3a are reasonable. In comparison to the choice probability functions of MNL (Figure 2-3c), the choice probability functions of the 5L-DNNs are richer and more flexible. The caveat is that the DNN choice probability functions may be too flexible to reflect the true behavioral mechanisms, owing to three theoretical challenges.

First, the large variation of individual models in Figure 2-3b reveal that DNN models are sensitive to the choice of hyperparameters. With different hyperparameters, some of HP-DNNs' choice probability functions are simply flat without revealing any useful information, while others are similar to 5L-DNNs with reasonable patterns. This challenge can be mitigated by hyperparameter searching and model ensemble. For example, the 5L-DNNs can reveal more reasonable economic information than the HP-DNNs because the 5L-DNNs use specific architectural and regularization hyperparameters, chosen from the results of hyperparameter searching based on their high prediction accuracy. In addition, as shown in Figure 2-3a, the choice probability function aggregated over models retains more smoothness and monotonicity than individual ones. The average choice probability function predicts that the driving

probability decreases the most when the driving cost increases from about $5 to $20, which is reasonable. Averaging models is an effective way of regularizing models because it reduces the large variance of the models with high complexity, such as DNNs [19].

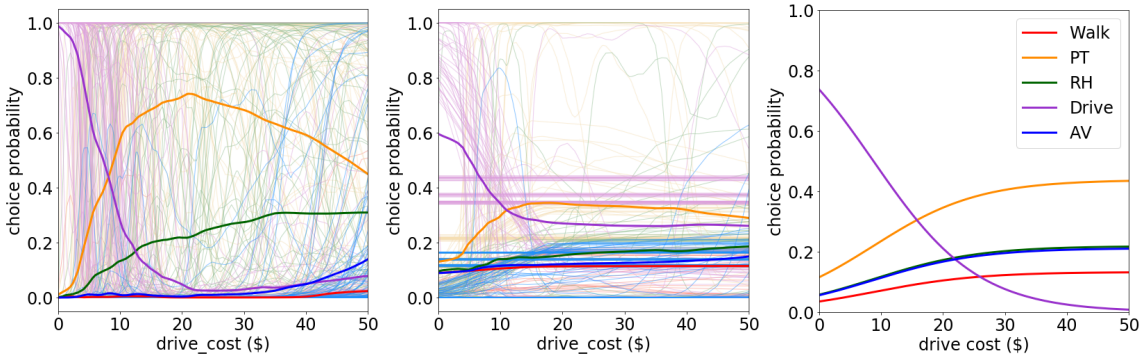
Second, the large variation of the individual 5L-DNN trainings (Figure 2-3a) reveal the challenge of model non-identification. Given that the 100 trainings are conditioned on the same training data and the same hyperparameters, the variation across the 5L-DNNs in Figure 2-3a is attributable to the model non-identification issue, or more specifically, the optimization difficulties in minimizing the non-convex risk function of DNNs. As DNNs' risk function is non-convex, different model trainings can converge to very different local minima or saddle points. Whereas these local minima have similar prediction accuracy, it brings difficulties to the model interpretation since the functions learnt from different local minima are different. For example, the three individual training results (C1, C2, and C3) have very similar out-of-sample prediction accuracy (60.2%, 59.5%, and 58.6%); however, their corresponding choice probability functions are very different. In fact, the majority of the 100 individual trainings have quite similarly high prediction accuracy, whereas their choice probability functions differ from each other. On the other side, the choice probability function averaged over the 100 trainings of the 5L-DNNs is more stable than individual ones. In practice, averaging over models is one effective way to provide a stable and reasonable choice probability function for interpretation.

Third, the shapes of the individual curves in Figure 2-3a show the local irregularity of the choice probability functions in certain regions of the input domain. First, some choice probability functions can be sensitive to the small change of input values; for example, the probability of choosing driving in C1 drops from 96.6% to 7.8% as the driving cost increases from $7 to $9, indicating a locally exploding gradient. This phenomenon of exploding gradients is acknowledged in the robust DNN discussions, because exploding gradients render a system vulnerable [111, 110]. Second, many training results present a non-monotonic pattern. For example, C3 represents a counter-intuitive case where the probability of driving starts to increase dramatically

as the driving costs are larger than $25. The local irregularity only exists in a limited region of the input domain: the driving probability becomes increasing when the cost is larger than $25, where the training sample is sparse. As a comparison, the average choice probability function of the 5L-DNNs has only a slight increasing trend when the driving cost is larger than $25, mitigating the local irregularity issue.

Substitution Pattern of Alternatives

The substitution pattern of the alternatives is of both practical and theoretical importance in choice analysis. In practice, researchers need to understand how market shares vary with input variables; in theory, the substitution pattern constitutes the critical difference between multinomial logit, nested logit, and mixed logit models. Figure 2-4 visualizes how the probability functions of the five alternatives vary as the driving cost increases. By visualizing the choice probabilities of all five alternatives, Figure 2-4 is an one-step extension of Figure 2-3.



(a) 5L-DNNs (b) HP-DNNs (c) MNL

Figure 2-4: Substitution patterns of five alternatives with varying driving costs

The substitution pattern of the 5L-DNNs is more flexible than that of the MNL models and more reasonable than that of the HP-DNNs. When the driving cost is smaller than $20, the substitution pattern of the 5L-DNNs aggregated over the 100 models illustrates that the five alternatives are substitute to each other, since the driving probability is decreasing while others are increasing. When the driving cost is larger than $20, the substitution pattern between walking, ridesharing, driving,

and using an AV still reveals the substitute nature. In a choice modeling setting, the alternatives in a choice set are typically substitutes: people are expected to switch from driving to other travel modes, as the driving cost increases. Therefore, the aggregated substitution pattern has mostly reflected the correct relationship of the five alternatives. However, the three theoretical challenges also permeate into the substitution patterns. The large variation in Figure 2-4b illustrates the high sensitivity to hyperparameters; the large variation in Figure 2-4a illustrates the model non-identification; and the individual curves in Figure 2-4a reveal the local irregularity. Even the model ensemble cannot solve all the problems. When the driving cost is larger than \$20, the average substitution pattern of the 5L-DNNs indicate that people are less likely to choose the public transit as the driving cost increases. This phenomenon seems unlikely because driving and public transit are supposed to be substitute to each other. As a comparison, the substitution pattern in Figure 2-4c, although perhaps exceedingly restrictive, reflects the travel mode alternatives being substitute goods. Therefore, DNNs can overall reveal a flexible substitution pattern of alternatives, although the pattern can be counter-intuitive in certain local regions of the input space.

Market Shares

Table 2.2 summarizes the market shares predicted by the three model groups. Each entry represents the average value of the market share over 100 trainings, and the number in the parenthesis is the standard deviation. Whereas the choice probability functions of 5L-DNNs can be unreasonable locally as discussed in section 2.5.2, the aggregated market share of 5L-DNNs are very close to the true market share, and it is more accurate than the HP-DNNs and the MNL models. It appears that the three challenges do not emerge in this discussion about market shares. The local irregularity could be cancelled out owing to the aggregation over the sample; the model non-identification appears less a problem when the market shares across the 5L-DNN trainings are very stable, as shown by the small standard deviations in the parenthesis; and the high sensitivity to hyperparameters is addressed by the selection

of the 5L-DNNs from the hyperparameter searching process, as the market shares of the 5L-DNNs are much more accurate than the HP-DNNs.

	5L-DNNs	HP-DNNs	MNL	True Market Share
Walk	8.98% (1.3%)	2.05% (3.6%)	4.78%	9.48%
Public Transit	23.4% (2.1%)	12.6% (15.1%)	23.1%	23.9%
Ride Hail	10.2% (1.2%)	2.17% (4.1%)	1.28%	10.8%
Drive	46.9% (1.8%)	80.4% (23.3%)	68.6%	44.5%
AV	10.5% (1.3%)	2.80% (4.5%)	2.2%	11.2%

Table 2.2: Market share of five travel modes (testing)

Social Welfare

Since DNNs have an implicit utility interpretation, we can observe how social welfare changes as action variables change the values. To demonstrate this process, we simulate one dollar decrease of the driving cost, and calculate the average social welfare change in the 5L-DNNs. We found that the social welfare increases by about \$520 in the 5L-DNN models after averaging over all 100 trainings. Interestingly, the magnitude of this social welfare change (\$520) is very intuitive and consistent with the one computed from MNL models, which is \$491 dollars. In the process of computing the social welfare change, we used the α_i averaged across 100 trainings as the individual i's marginal value of utility. Without using average α_i, individuals' marginal value of utility can take unreasonable values, caused by local irregularity and model non-identification. The problem associated with the individuals' gradient information will be discussed in details in the following section.

Interpretation Methods

The four subsections above interpret DNNs by using choice probability and utility functions. Both are widely used in the generic studies about DNN interpretation, although usually referred to by different names. For example, researchers interpret DNNs by identifying the representative observation for each class. The

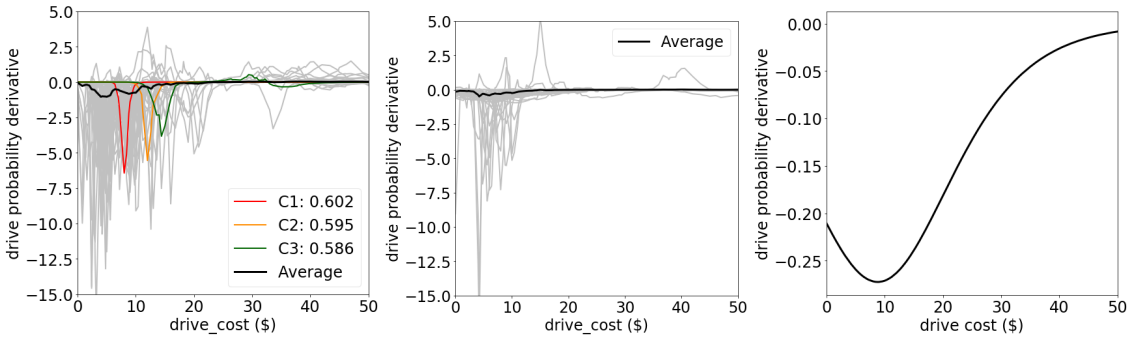
method is called activation maximization (AM) $\hat{x}_k = \underset{x}{\text{argmax}} \ \log P(y = k|x)$, which maximizes the conditional probability density function with respect to the input x [36, 117, 88, 67]. The choice probabilities are also referred to as soft labels, used to distill knowledge by retraining a simple model to fit a complicated DNN [55]. Researchers in the computer vision field interpret DNN results by mapping the neurons of the hidden layers to the input space [147] or visualizing the activation maps in the last layer [151]. Since utilities are just the activation maps of the last layer, our interpretation approach is similar to those used in computer vision. In these generic discussions about DNN interpretation, the differentiation between the utility function and the choice probability functions is weak, since their mapping is monotonic and the function properties are similar.

2.5.3 Gradient-Based Interpretation

Gradients of Choice Probability Functions

The gradient of choice probability functions offers opportunities to extract more important economic information. Since researchers often seek to understand how to intervene to trigger behavioral changes, the most relevant information is the partial derivative of the choice probability function with respect to a targeting input variable. Figure 2-5 visualizes the corresponding probability derivatives of the choice probability functions in Figure 2-3. As shown below, both the strength and the challenges identified in the choice probability functions are retained in the properties of the probability derivatives.

In Figure 2-5a, the majority of the 5L-DNNs, such as the three curves (C1, C2, and C3), take negative values and have inverse bell shapes. This inverse bell shaped curve is intuitive because people are not as sensitive to price changes when price is close to zero or infinity, but are more sensitive when price is close to a certain tipping point. The shapes revealed by 5L-DNNs are similar to the MNL models. The probability derivative of MNL models is $\partial s(x)/\partial x = s(x)(1-s(x)) \times (\partial V(x)/\partial x)$, which is mostly negative and take a very regular inverse bell shape, as shown in Figure 2-5c.



(a) 5L-DNNs (b) HP-DNNs (c) MNL

Figure 2-5: Probability derivatives of choosing driving with varying driving costs

The sensitivity to hyperparameters, the model non-identification, and the local irregularity are also shown in Figure 2-5, similar to the discussions in Figure 2-3. HP-DNNs reveal more unreasonable behavioral patterns than 5L-DNNs, as many of the input gradients are flat on zero, demonstrating the importance of selecting correct hyperparameters. The variation of individual trainings in Figure 2-5a demonstrates the challenge of model non-identification. With fixed training samples and hyperparameters, the DNN trainings can lead to different training results, thus creating difficulty for researchers to choose a final model for interpretation. The exploding gradients and the non-monotonicity issues, as the two indicators of local irregularity, are also clearly illustrated in the individual trainings in Figure 2-5a. The absolute values of many probability derivatives are of large magnitude; for example, at the peak of the C1 curve, \$1 cost increase leads to about 6.5% change in choice probability [2], which is much larger than the MNL models. Similar to the previous discussions, hyperparameter searching and information aggregation can mitigate these issues.

Elasticities

To compare across input variables, researchers often compute elasticities because the elasticities are standardized derivatives. Given that DNNs provide choice probability derivatives, it is straightforward to compute the elasticities from DNNs. Table 2.3

[2]This 6.5% appears much smaller than the values in Figure 2-3. It is because of the difference between arc and point elasticities.

presents the elasticities of travel mode choices with respect to input variables. Each entry represents the average elasticity across the 100 trainings of the 5L-DNNs, and the value in the parenthesis is the standard deviation of the elasticities across the 100 trainings. Unlike a regression table, the standard deviation in Table 2.3 is *not* caused by the sampling randomness, but by the non-identification of models.

	Walk	Public Transit	Ride Hailing	Driving	AV
Walk time	**-5.308(6.9)**	0.399(5.9)	-0.119(7.1)	-0.030(4.6)	-1.360(6.8)
Public transit cost	-1.585(9.6)	**-4.336(9.6)**	-1.648(11.1)	1.081(5.9)	1.292(9.5)
Public transit walk time	0.123(6.9)	**-1.707(6.5)**	0.047(7.3)	0.621(4.7)	0.844(6.7)
public transit wait time	0.985(8.7)	**-2.520(8.9)**	-0.518(9.1)	0.092(5.8)	0.366(8.8)
Public transit in-vehicle time	0.057(9.0)	**-1.608(9.0)**	0.484(9.4)	0.778(5.8)	1.273(8.9)
Ride hail cost	-2.353(7.6)	0.005(6.9)	**-4.498(8.9)**	0.304(5.6)	-0.243(9.0)
Ride hail wait time	0.234(8.8)	1.471(8.3)	**-2.536(10.1)**	-0.253(5.7)	-0.228(8.8)
Ride hail in-vehicle time	0.299(7.8)	-0.224(7.4)	**-5.890(9.4)**	0.740(5.4)	0.739(7.6)
Drive cost	1.124(6.6)	2.545(5.9)	3.760(6.8)	**-1.886(5.0)**	2.273(6.9)
Drive walk time	2.033(5.3)	0.552(5.0)	2.503(5.6)	**-0.412(3.8)**	1.787(5.4)
Drive in-vehicle time	1.824(9.0)	4.163(8.2)	3.640(9.9)	**-3.199(7.4)**	3.268(9.1)
AV cost	-0.562(6.5)	-0.198(6.2)	0.819(6.9)	0.337(4.6)	**-4.289(7.6)**
AV wait time	-0.068(7.9)	-0.695(7.4)	2.400(8.4)	0.284(4.6)	**-1.591(7.8)**
AV in-vehicle time	-0.784(6.2)	0.221(5.6)	0.955(7.1)	0.079(4.3)	**-4.534(6.8)**
Age	-1.003(18.7)	2.502(18.4)	-4.385(20.0)	0.949(13.7)	-1.936(18.6)
Income	1.127(10.7)	0.727(10.5)	0.957(11.9)	-0.002(6.7)	2.539(10.8)

Table 2.3: Elasticities of five travel modes with respect to input variables

The average elasticities of the 5L-DNNs are reasonable in terms of both the signs and magnitudes. We highlight the elasticities that relate the travel modes to their own alternative-specific variables. These highlighted elasticities are all negative, which is very reasonable since higher travel cost and time should lead to lower probability of adopting the corresponding travel mode. The magnitudes are higher than the typical results from the MNL models. For example, Table 2.3 indicates that 1% increase in public transit cost, walking time, waiting time, and in-vehicle travel time leads to the decrease of 4.3%, 1.7%, 2.5%, and 1.6% probability in using public transit. In addition, the highlighted elasticities are overall of a larger magnitude than others, which is also reasonable since the self-elasticity values are typically larger than cross-elasticity values. Therefore, as the elasticity values are aggregated over the trainings and the sample, these values are quite reasonable.

Model non-identification is revealed here by the large standard deviations of the elasticities. For example, as the walking elasticity regarding walking time is -5.3 on average, its standard deviation is 6.9. This large standard deviation is caused by model non-identification, as every training leads to a different model and a different elasticity. The high sensitivity and the local irregularity issues are not present in the process of computing the average elasticities, because the 5L-DNNs are trained by the same hyperparameter and the local irregularity is partially mitigated by the aggregation over the sample.

Marginal Rates of Substitution: Values of Time

VOT, as one example of MRS, is one of the most important pieces of economic information obtained from choice models, since the monetary gain from time saving is the most prevalent benefit from the improvement of any transportation system. As VOT is computed as the ratio of two parameters in a MNL model, the ratio of two probability derivatives represents the VOT in the DNN setting. Figure 2-6 presents the distribution of the VOTs of the 5L-DNNs. The distribution has a very large dispersion and even some negative values, caused by the model non-identification issue.

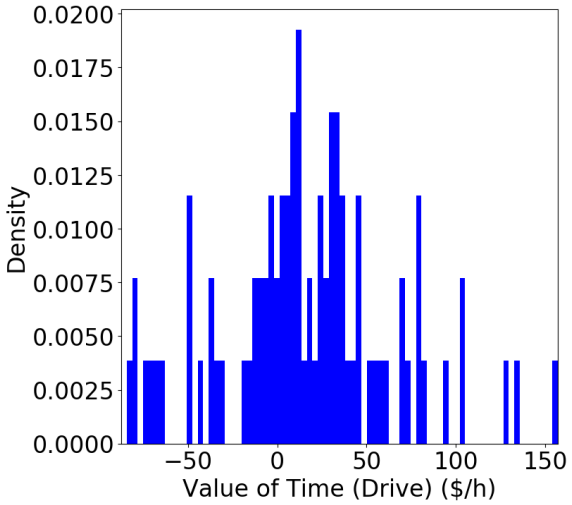
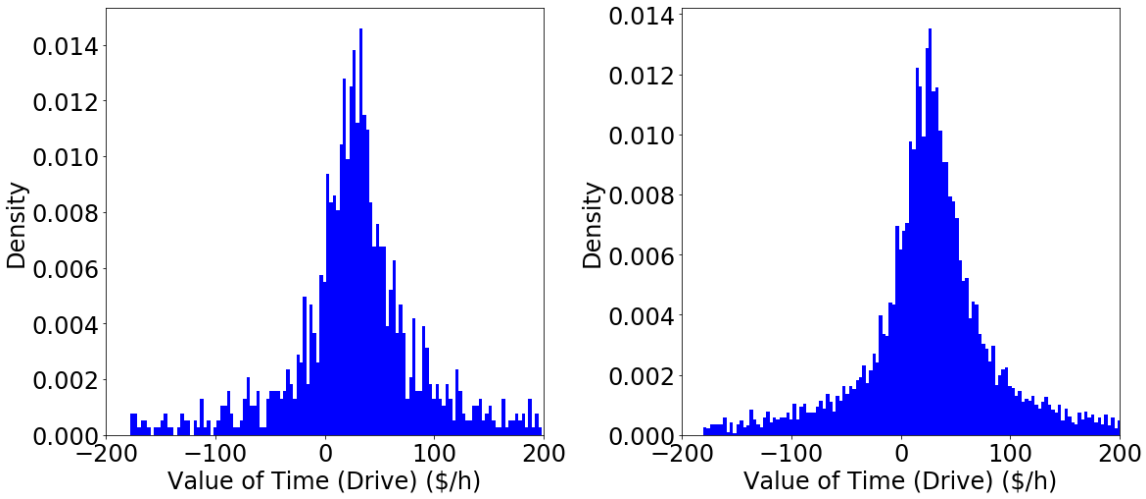


Figure 2-6: Values of time (5L-DNNs with 100 model trainings)

Heterogeneity of Preference: VOT

Since different people often have different VOT, Figure 2-7 shows the distribution of the heterogeneous VOT of the individuals in the training and testing sets. The distribution of the VOT in Figure 2-7 is the individuals' VOT in one specific 5L-DNN model run, different from the distribution of the VOT in Figure 2-6, which represents the distribution of the VOT across the 100 5L-DNNs model runs. As shown by Figure 2-7, heterogeneous VOT can be automatically identified from the DNN models, and the median VOT in the training and testing sets are respectively \$26.8/h and \$27.8/h. The VOT distribution is highly concentrated around its mean value, resembling the shape of a Gaussian distribution.



(a) 5L-DNNs (Testing) (b) 5L-DNNs (Training)

Figure 2-7: Heterogeneous values of time in the training and testing sets (one model training)

The median \$27/h VOT in Figure 2-7 is consistent with previous studies. In previous studies, VOT has been found to be between \$7.8/h and \$30.3/h for various travel modes [57]. VOT has also been found to be between 21% and 254% of the hourly wage in a review paper [146]. By using the average hourly wage (\$27.16/h) of the U.S workers in 2018, we would expect the VOT here to be between \$5.7/h and \$70.0/h. Our VOT obtained from DNNs is about in the middle of this range. Intuitively, the VOT should be of the same magnitude as the hourly wages, and

$\$27/h$ is very close to the average hourly wage. However, on the other hand, the VOT obtained from DNNs can be unreasonable for certain individuals. It is highly unlikely for VOT to be negative, while DNNs detect a sizeable portion of people whose VOT are negative. This counter-intuitive result is caused by the local irregularity of the probability derivatives. As the VOT equals the ratio of two derivatives, VOT can become abnormal when any one of the two derivatives takes abnormal values.

Interpretation Methods

The four subsections above interpret DNNs by using the input gradients, a commonly used approach in the generic DNN interpretation literature. It is often referred to by different names such as sensitivity analysis, saliency, or attribution maps in computer vision [117, 67], or attention mechanism in natural language processing [145]. In the transportation field, some studies used input gradients to describe the relationship between inputs and outputs in DNNs [15, 107, 48]. Recently, researchers in the ML community are increasingly focusing their attention on the properties of DNNs' input gradients, owing to their importance in DNN interpretation [120, 114, 123, 110].

2.6 Discussions: Towards Reliable Economic Information from DNNs

There should be little doubt that DNNs can provide a rich set of economic information. The challenge, however, is how to make the economic information from DNNs more *reliable*. This study has demonstrated the importance of using hyperparameter searching, repeated trainings conditioned on the fixed hyperparameters, and aggregation over models and population to improve the reliability of the economic information. Specifically, we found that the *aggregated* economic information, whether over the trainings or the sample, becomes more reliable than the *disaggregate* economic information. The average choice probability function, average probability derivatives, and average VOT are all more reliable than the corresponding results

of single trainings, individuals, and the specific regions in the input domain. This result is intuitive since model ensemble can be seen as a regularization method and the summation over the sample may cancel out the individual irregularities. Recent studies have provided other methods of improving the reliability of economic information extracted from DNNs and addressing the three challenges which are related to three broad research fields in the ML community.

With better regularization methods, DNN architectures, hyperparameter tuning algorithms, statistical theoretical understanding, or larger sample sizes, DNNs can control its large estimation error, thus providing more reliable economic information for interpretation. Researchers have explored a massive number of regularization methods, such as domain constraints, Bayesian priors, model ensemble [73], data augmentation [18], dropouts [56], early stopping, sparse connectivity, and many others that influence the DNN models through the computational process [43, 86]. Researchers also identified an extremely large number of more effective DNN architectures, such as AlexNet [73], GoogleNet [125], and ResNet [52] in the computer vision field. The process of selecting hyperparameters can also be automatically addressed by using Gaussian process, Bayesian neural networks [121, 122], or reinforcement learning [152, 153, 7], much richer than a simple random searching [17, 16]. Theoretically, statisticians have provided tighter bounds on the estimation errors of DNNs than the classical VC dimension bound [133, 10, 8, 92, 42]. In addition, even simply increasing the sample size can improve DNN model performance because of the tighter control on its large estimation errors (Appendix I).

With better optimization algorithms, DNN models can mitigate the model non-identification issue. In fact, the optimization algorithm has been refined significantly in the past years to the extent that it converges to the simple first order stochastic gradient descent with momentum [69] and specific initialization methods [41, 51]. However, model non-identification is viewed differently from the other two issues, because researchers tend to believe it is no longer a problem. Local minima can still provide high-quality predictions, and global minimum might even overfit the training set, leading to the low performance in the testing set [28].

With robust training methods and monotonicity constraints, the DNN models can mitigate the local irregularity, becoming more economically interpretable. To formally measure local irregularity, researchers evaluated the model performance on adversarial examples [44, 74, 75]. To defend against the adversarial attacks, researchers designed the adversarial training with adversarial examples [74], defensive knowledge distillation [99], mini-max robust training [84], and even simple gradient regularization [110]. To address the non-monotonicity issue, researchers developed various types of constraints to guarantee its monotonicity [46].

2.7 Conclusion

This study aims to interpret DNN models in the context of choice analysis and extract economic information as complete as obtained from classical DCMs. The economic information includes a complete list of choice predictions, choice probabilities, market share, substitution patterns of alternatives, social welfare, probability derivatives, elasticity, marginal rates of substitution (MRS), and heterogeneous values of time (VOT). The process of interpreting DNN models is different from classical DCMs because DNNs are a very flexible model family, capable of automatically learning more flexible behavioral patterns than the regular patterns pre-specified by domain experts in the classical DCMs. As a result, we found that most economic information extracted from DNN is reasonable and more flexible than the MNL models. However, the economic information automatically learnt by DNNs is sometimes unreliable, caused by three challenges: high sensitivity to hyperparameters, model non-identification, and local irregularity. Owing to the high sensitivity to hyperparameters, the DNN models without appropriate regularizations or architectures cannot provide valuable economic information. Owing to the model non-identification, researchers cannot obtain a definitive function estimate for economic interpretation. Owing to the local irregularity, DNN models reveal unreasonable local behavioral patterns. These three problems can be partially addressed by using simple random hyperparameter searching, repeated trainings on fixed hyperparameters, and infor-

mation aggregation. Particularly, the economic information aggregated over trainings or the sample, such as the average choice probability function, average probability derivatives, market shares, average social welfare change, average elasticities, and the median VOT, are mostly consistent with our behavioral intuition and previous studies.

Beyond the methods used in this study, each challenge can be addressed in many other ways. To address the high sensitivity issue, researchers need to choose better regularizations, DNN architectures, or more automatic algorithms for hyperparameter searching. To address model non-identification, researchers can use better optimization algorithms or initialization procedures. To address local irregularity, researchers can use robust DNN training methods. In each of these directions, future studies can explore the established methods in the ML community or create more domain-specific solutions for choice analysis.

Appendix I: Estimation Error of Prediction Accuracy in DNNs

Definition 1 *Excess error of $\hat{f}$ is defined as*

$$\mathbb{E}_S[L(\hat{f}) - L(f^*)] \tag{2.5}$$

which is the same as estimation error when no approximation error exists.

$L(\hat{f})$ is the population error of the estimator; $L(f^*)$ is the population error of the true model; $L = \mathbb{E}_{x,y}[l(y, f(x))]$ and $l(y, f(x))$ is the loss function. Excess error measures to what extent the error of the estimator deviates from the true model, averaged over random sampling S.

Proposition 1 *The estimation error of $\hat{f}$ can be bounded by VC dimension*

$$\mathbb{E}_S[L_{0/1}(\hat{f}) - L_{0/1}(f^*)] \lesssim O(\frac{v}{N}) \tag{2.6}$$

in which v is the VC dimension of function class $\mathcal{F}$; N is the sample size; $L_{0/1}$ is the binary prediction error.

Proof. When no misspecification error exists, estimation error can be further decomposed as three terms

$$\begin{aligned}
\mathbb{E}_S[L(\hat{f}) - L(f^*)] &= \mathbb{E}_S[L(\hat{f}) - \hat{L}(\hat{f}) + \hat{L}(\hat{f}) - \hat{L}(f^*) + \hat{L}(f^*) - L(f^*)] && (2.7)\\
&\leq \mathbb{E}_S[L(\hat{f}) - \hat{L}(\hat{f})] && (2.8)\\
&\leq \mathbb{E}_S \sup_{f \in \mathcal{F}} [L(f) - \hat{L}(f)] && (2.9)
\end{aligned}$$

in which $\hat{L}(f) := \frac{1}{N}\sum_i l(y_i, f^*(x_i))$; the first inequality holds because $\mathbb{E}_S[\hat{L}(\hat{f}) - \hat{L}(f^*)] \leq 0$ based on the definition of $\hat{f} := \operatorname{argmin} \hat{L}(f)$ and $\mathbb{E}_S[\hat{L}(f^*) - L(f^*)] = 0$ based on the law of large numbers; the second inequality holds due to the sup operator.

Equation 2.9 can be bounded.

$$\mathbb{E}_S \sup_{f\in\mathcal{F}} [L(f) - \hat{L}(f)] \leq 2\mathbb{E}_{S,\epsilon} \sup_f \frac{1}{N}\sum_i l(f(x_i), y_i)\epsilon_i \tag{2.10}$$

This proof relies on the technique called symmetrization, as shown in the proof of Theorem 4.10 in [139]. Note that for prediction error, the loss function $l(f(x_i), y_i) = \mathbb{1}\{f(x_i) \neq y_i\} = y_i + (1-2y_i)f(x_i)$, as $y_i \in \{0,1\}$ and $f(x_i) \in \{0,1\}$. By applying contraction inequality to Equation 2.10,

$$2\mathbb{E}_{S,\epsilon} \sup_f \frac{1}{N}\sum_i l(f(x_i), y_i)\epsilon_i = 2\mathbb{E}_{S,\epsilon} \sup_f \frac{1}{N}\sum_i (y_i + (1-2y_i)f(x_i)) \times \epsilon_i \tag{2.11}$$

$$\leq 2\mathbb{E}_{S,\epsilon} \sup_f \frac{1}{N}\sum_i f(x_i)\epsilon_i \tag{2.12}$$

$$= 2\mathbb{E}_S \hat{\mathcal{R}}_N(\mathcal{F}\,|_S) \tag{2.13}$$

in which the second line uses the contraction inequality [77] and the third uses the definition of Rademacher complexity. Basically the question about the upper bound of estimation error is turned to the question about the complexity of function class of DNN $\mathcal{F}$. There are many ways to derive an upper bound on Rademacher complexity [10]. To obtain the v/N result, Dudley integral and chaining techniques are useful. Let $Z_f := \frac{1}{\sqrt{N}}\sum_i \epsilon_i f(x_i)$ and $Z_g := \frac{1}{\sqrt{N}}\sum_i \epsilon_i g(x_i)$, in which $f, g \in \mathcal{F}$. Based on Theorem 5.22 Dudley's entropy integral bound in [139],

$$\mathbb{E}_S \left[\sup_{f\in\mathcal{F}} Z_f\right] \leq \mathbb{E}_S \left[\sup_{f,g\in\mathcal{F}} Z_f - Z_g\right] \tag{2.14}$$

$$\leq 2\mathbb{E}_S\Big[\sup_{f',g'\in\mathcal{F};\rho_x(f',g')\leq\delta} Z_{f'} - Z_{g'}\Big] + 32\int_{\delta/4}^{D}\sqrt{\log N_x(u;\mathcal{F})}du \tag{2.15}$$

in which $\rho_x^2(f',g') = \frac{1}{N}\sum_{i=1}^N (f'(x_i) - g'(x_i))^2$; f' and g' are the components around the δ distance of one element in the δ cover of function class $\mathcal{F}$; D is the diameter of the function class $\mathcal{F}$ projected to dataset S, defined as $D := \sup_{f,g\in\mathcal{F}} \rho_x(f,g) \leq 1$; δ is any positive value in $[0, D]$. Equation 2.15 holds for any δ. The first term in Equation 2.15 measures the local complexity of DNN and the second term measures

the error caused by discretization of the function space. The two terms could be bounded separately. For the first term,

$$\mathbb{E}_S\Big[\sup_{f',g'\in\mathcal{F};\rho_x(f',g')\leq\delta} Z_{f'} - Z_{g'}\Big] = \mathbb{E}_S\Big[\sup_{\rho_x(f',g')\leq\delta} \frac{1}{\sqrt{N}}\sum_i \epsilon_i(f'(x_i) - g'(x_i))\Big] \tag{2.16}$$

$$= \delta\mathbb{E}_S||\epsilon||_2 \tag{2.17}$$

$$\leq \delta\sqrt{\mathbb{E}\sum_i \epsilon_i^2} \tag{2.18}$$

$$\leq \delta\sqrt{N} \tag{2.19}$$

in which the second line uses the dual norm; the third line uses the fact that ϵ_i is a 1 sub-Gaussian random variable. For the second term in Equation 2.15, we need to use the Haussler fact [50] that

$$N_x(u;\mathcal{F}) \leq Cv(16e)^v(\frac{1}{u})^v$$

It implies

$$32\int_{\delta/4}^{D}\sqrt{\log N_x(u;\mathcal{F})}du \leq 32\int_{\delta/4}^{D}\sqrt{\log\left[Cv(16e)^v(\frac{1}{u})^v\right]}du \tag{2.20}$$

$$= 32\int_{\delta/4}^{D}\sqrt{\log C + \log v + v\log 16e + v\log\frac{1}{u}}\,du \tag{2.21}$$

$$\leq c_0\sqrt{v}\int_{\delta/4}^{D}\sqrt{\log\frac{1}{u}}\,du \tag{2.22}$$

$$\leq c_0\sqrt{v}\int_{0}^{D}\sqrt{\log\frac{1}{u}}\,du \tag{2.23}$$

$$\leq c_0'\sqrt{v} \tag{2.24}$$

By plugging in the upper bounds on the two terms back to Equation 2.15 and dividing

both side by $\sqrt{N}$, it implies

$$\mathbb{E}_S \sup_{f\in\mathcal{F}} \frac{1}{N}\sum_i \epsilon_i f(x_i) \leq \inf_\delta \Big[\delta + c_0'\sqrt{\frac{v}{N}}\Big] \tag{2.25}$$

$$= c_0'\sqrt{\frac{v}{N}} \tag{2.26}$$

Therefore, the estimation error can be bounded:

$$\mathbb{E}_S[L(\hat{f}) - L(f^*)] \lesssim O(\sqrt{\frac{v}{N}}) \tag{2.27}$$

Remarks. Intuitively, v/N describes the tradeoff between model complexity and sample size. In a typical MNL model, v is of the same scale as the number of parameters and the input dimension d; on the contrary, DNN is a much more complex nonlinear model with much larger v. As proved by Bartlett (2017) [8], DNN with W denoting the number of weights and L denoting the depth has VC dimension $O(WL\log(W))$. For instance, when a dataset has 25 input variables, the VC dimension of a simple DNN with 8 layers and 100 neurons as its width is about $320,000$, as opposed to $v = 25$ as the VC dimension of MNL. Theorefore, the theoretical upper bound of DNN on its estimation error is much larger than MNL model.

Statistical learning theory is a very broad field that can be used to prove the upper bound on the estimation error [136, 139]. Proposition 1 is limited to the binary discrete output, although its extension to multiple classes and continuous output is also possible. The theoretically optimum upper bound on DNN's estimation error is still an ongoing research field. Statisticians have been exploring different methods to bound DNN, and the methods based on empirical process theory and the contraction inequaility could provide the tightest upper bound so far [42, 92, 10, 77]. To understand the property of the choice probability functions, I provide a second proof in Appendix II.

Appendix II: Estimation Error of Choice Probability Functions in DNNs

Proposition 2 *The estimation error of $\hat{f}$ can be upper bounded by the Rademacher complexity*

$$\mathbb{E}_S[L(\hat{f}) - L(f_F^*)] \leq 2\mathbb{E}_S\hat{\mathcal{R}}_n(l \circ \mathcal{F}|_S) \tag{2.28}$$

Proof. Estimation error can be decomposed:

$$\begin{aligned}
\mathbb{E}_S[L(\hat{f}) - L(f_F^*)] &= \mathbb{E}_S[L(\hat{f}) - \hat{L}(\hat{f}) + \hat{L}(\hat{f}) - \hat{L}(f_F^*) + \hat{L}(f_F^*) - L(f_F^*)] && (2.29)\\
&\leq \mathbb{E}_S[L(\hat{f}) - \hat{L}(\hat{f})] && (2.30)\\
&\leq \mathbb{E}_S[\sup_{f \in F} |L(f) - \hat{L}(f)|] && (2.31)
\end{aligned}$$

The first inequality holds since (1) $\hat{L}(\hat{f}) - \hat{L}(f_F^*) \leq 0$ due to the definition of $\hat{f}$ and (2) $\mathbb{E}_S[\hat{L}(f_F^*) - L(f_F^*)] = 0$ due to law of large numbers; the second inequality holds since $\hat{f}$ is only one function in F ($\mathcal{F}_0$ or $\mathcal{F}_1$ in this study). The right hand side of Equation 2.31 above can be further upper bounded by using a technique called symmetrization. Formally, suppose another set of $\{x_i'\}_1^N$ is also generated, following

the same distribution as $\{x_i\}_1^N$. Then

$$\mathbb{E}_S\Big[\sup_{f\in F}\Big|L(f)-\hat{L}(f)\Big|\Big] = \mathbb{E}_S\Big[\sup_{f\in F}\Big|\mathbb{E}_{x,y}[l(y,f(x))]-\frac{1}{N}\sum_{i=1}^{N}l(y_i,f(x_i))\Big|\Big] \tag{2.32}$$

$$= \mathbb{E}_S\Big[\sup_{f\in F}\Big|\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_{x'}l(y,f(x_i'))-\frac{1}{N}\sum_{i=1}^{N}l(y_i,f(x_i))\Big|\Big] \tag{2.33}$$

$$\leq \mathbb{E}_{S,S'}\Big[\sup_{f\in F}\Big|\frac{1}{N}\sum_{i=1}^{N}l(y,f(x_i'))-\frac{1}{N}\sum_{i=1}^{N}l(y_i,f(x_i))\Big|\Big] \tag{2.34}$$

$$= \mathbb{E}_{S,S'}\Big[\sup_{f\in F}\Big|\frac{1}{N}\sum_{i=1}^{N}\epsilon_i(l(y,f(x_i'))-l(y_i,f(x_i))\Big|\Big] \tag{2.35}$$

$$\leq \mathbb{E}_{S,S'}\Big[\sup_{f\in F}\Big|\frac{1}{N}\sum_{i=1}^{N}\epsilon_i l(y,f(x_i')\Big|+\Big|\frac{1}{N}\sum_{i=1}^{N}\epsilon_i l(y_i,f(x_i)\Big|\Big] \tag{2.36}$$

$$\leq 2\mathbb{E}_S\hat{\mathcal{R}}_n(l\circ\mathcal{F}|_S) \tag{2.37}$$

The first line uses the definition of L and $\hat{L}$; the second line uses the symmetrization technique by which $\mathbb{E}_{x,y}$ is replaced by an average of another sample $\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_{x'}l(y,f(x_i'))$; the third line uses $\mathbb{E}\ \sup \geq \sup\ \mathbb{E}$ and uses S' to denote the new sample $\{x'\}_1^N$; the fourth line adds the Rademacher random variable ϵ_i due to the symmetry of S and S'; the fifth line uses the fact $\sup|A+B| \leq \sup|A| + \sup|B|$; and the last line is the definition of Rademacher complexity. The following proposition provides a tighter upper bound on the estimation error of the interpretation loss.

Definition 2 *Mean squared error (MSE) is defined as*

$$L_{mse}(s) = \mathbb{E}_{x,y}[(y-s(x))^2] \tag{2.38}$$

The corresponding empirical mean squared error is defined as

$$\hat{L}_{mse}(s) = \frac{1}{N}\sum_{i=1}^{N}(y_i-s(x_i))^2 \tag{2.39}$$

Lemma 2.7.1 *Estimation error for interpretation equals to that of MSE.*

$$\mathbb{E}_S[L_{mse}(\hat{s}) - L_{mse}(s_F^*))] = \mathbb{E}_S[L_e(\hat{s}) - L_e(s_F^*))] \tag{2.40}$$

Proof of Lemma 2.7.1. Since y is sampled as a Bernoulli random variable with probability $s^*(x)$, $E[y|x] = s^*(x)$.

$$\mathbb{E}_{S,x,y}[(\hat{s}(x) - y)^2] \tag{2.41}$$

$$= \mathbb{E}_{S,x,y}((\hat{s}(x) - s^*(x) + s^*(x) - y)^2) \tag{2.42}$$

$$= \mathbb{E}_{S,x,y}[((\hat{s}(x) - s^*(x))^2 + 2(\hat{s}(x) - s^*(x))(s^*(x) - y) + (s^*(x) - y)^2)] \tag{2.43}$$

$$= \mathbb{E}_{S,x,y}[(\hat{s}(x) - s^*(x))^2] + \mathbb{E}_{x,y}[(s^*(x) - y)^2)] + 2\mathbb{E}_{S,x,y}[(\hat{s}(x) - s^*(x))(s^*(x) - y)] \tag{2.44}$$

$$= \mathbb{E}_{S,x,y}[(\hat{s}(x) - s^*(x))^2] + \mathbb{E}_{x,y}[(s^*(x) - y)^2)] + 2\mathbb{E}_x\left[\mathbb{E}_{S,y}[(\hat{s}(x) - s^*(x))(s^*(x) - y)|x]\right] \tag{2.45}$$

$$= \mathbb{E}_{S,x,y}[(\hat{s}(x) - s^*(x))^2] + \mathbb{E}_{x,y}[(s^*(x) - y)^2)] + 2\mathbb{E}_x\left[\mathbb{E}_S[(\hat{s}(x) - s^*(x))|x]\mathbb{E}_y[(s^*(x) - y)|x]\right] \tag{2.46}$$

$$= \mathbb{E}_{S,x,y}[(\hat{s}(x) - s^*(x))^2] + \mathbb{E}_{x,y}[(s^*(x) - y)^2)] \tag{2.47}$$

The fourth equality uses Law of Iterated Expectation; the fifth uses the conditional independence $S \perp y|x$; the lase one uses $E[y|x] = s^*(x)$. With very similar process, we could show

$$\mathbb{E}_{x,y}[(y - s_F^*(x))^2] \tag{2.48}$$

$$= \mathbb{E}_{x,y}[(y - s^*(x) + s^*(x) - s_F^*(x))^2] \tag{2.49}$$

$$= \mathbb{E}_{x,y}[(y - s^*(x))^2] + \mathbb{E}_{x,y}[(s^*(x) - s_F^*(x))^2] + 2\mathbb{E}_{x,y}[(y - s^*(x))(s^*(x) - s_F^*(x))] \tag{2.50}$$

$$= \mathbb{E}_{x,y}[(y - s^*(x))^2] + \mathbb{E}_{x,y}[(s^*(x) - s_F^*(x))^2] + 2\mathbb{E}_x\big[(s^*(x) - s_F^*(x))\mathbb{E}_y[y - s^*(x)|x]\big] \tag{2.51}$$

$$= \mathbb{E}_{x,y}[(y - s^*(x))^2] + \mathbb{E}_{x,y}[(s^*(x) - s_F^*(x))^2] \tag{2.52}$$

Combining the two equations above implies

$$\mathbb{E}_{x,y}[(s^*(x) - y)^2)] = \mathbb{E}_{S,x,y}[(\hat{s}(x) - y)^2] - \mathbb{E}_{S,x,y}[(\hat{s}(x) - s^*(x))^2] \tag{2.53}$$

$$= \mathbb{E}_{x,y}[(y - s_F^*(x))^2] - \mathbb{E}_{x,y}[(s^*(x) - s_F^*(x))^2] \tag{2.54}$$

By changing the notation, it implies

$$\mathbb{E}_S[L_{mse}(\hat{s}) - L_{mse}(s_F^*))] = \mathbb{E}_S[L_e(\hat{s}) - L_e(s_F^*))] \tag{2.55}$$

Proposition 3 *The estimation error of the choice probability functions can be upper bounded by*

$$\mathbb{E}_S[L_e(\hat{s}) - L_e(s_F^*)] \leq 4\mathbb{E}_S\hat{\mathcal{R}}_n(\mathcal{F}|_S) \tag{2.56}$$

in which $L_e(s)$ is defined as

$$L_e(s) = ||s^* - s||^2_{L^2(P_x)} = \int_x (s^*(x) - s(x))^2 dP(x) \tag{2.57}$$

Proof of Proposition 3. Lemma 2.7.1 shows that the estimation error on function estimation is the same as the one on MSE. Hence we will provide an upper bound on the MSE by using Proposition 2. Formally,

$$\mathbb{E}_S[L_{mse}(\hat{s}) - L_{mse}(s_F^*))] \leq 2\mathbb{E}_S[\hat{R}_n(l \circ \mathcal{F}\,|_S)] \tag{2.58}$$

$$\leq 4\mathbb{E}_S[\hat{R}_n(\mathcal{F}\,|_S)] \tag{2.59}$$

The first inequality uses Proposition 2; the second uses contraction inequality and the fact that squared loss here is bounded between $[0, 1]$ and that its Lipschitz constant is at most two. $\square$

Appendix III: Hyperparameter Space

Depth	$[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$
Width	$[25, 50, 100, 150, 200]$
L_1 penalty constants	$[0.1, 1E-2, 1E-3, 1E-5, 1E-10, 1E-20]$
L_2 penalty constants	$[0.1, 1E-2, 1E-3, 1E-5, 1E-10, 1E-20]$
Dropout rates	$[0.01, 1E-5]$

Table 2.4: Hyperparameter space

Chapter 3

Essay 2: Architectural Design with Alternative-Specific Utility Functions

3.1 Introduction

For decades, choice analysis has been an important research area across economics, transportation, and marketing [85, 13, 45]. Whereas discrete choice models were traditionally used to analyze this question, recently researchers have become increasingly interested in applying machine learning (ML) methods such as deep neural network (DNN) to analyze individual choices [65, 101, 140]. Whereas DNN has revealed its extraordinary predictive power in the tasks such as image recognition and natural language processing, its application to demand analysis is still hindered by at least three problems. First, as DNN gradually permeates into many domains, it is unclear how generic-purpose DNN classifiers can be reconciled with domain-specific knowledge [76, 78]. Whereas researchers in the ML community generally admire the effectiveness of automatic feature learning in DNN [76], heated debate continues with regard to the extent and manner in which domain knowledge can be used to improve ML models to solve domain-specific problems more efficiently [78]. Because DNN is a significantly complicated generic-purpose model, its interpretability is generally considered to be low [80, 70]. Whereas it is relatively straightforward to apply DNN to forecast demand, researchers have been able to obtain very limited policy and

behavioral insights from DNN until now. Morevoer, even prediction itself can be challenging, because DNN with high dimensionality could straightforwardly overfit data. To guarantee a high out-of-sample performance, it is critical to design effective regularization methods and DNN architectures. Whereas many recent progresses were achieved by creating novel DNN architectures, the procedure of designing deep architecture is still ad hoc without systematic guidance [148, 87]. These three challenges, including the tension between domain-specific and generic-purpose knowledge, lack of interpretability, and challenge of identifying effective regularization and architecture, are theoretically important and empirically crucial for applying DNN to any specific domain.

To address these problems, this study demonstrates the use of behavioral knowledge for designing a novel DNN architecture with alternative-specific utility functions (ASU-DNN), thereby improving both the predictive power and interpretability of DNN in choice analysis. We first elaborate on the implicit interpretation of random utility maximization (RUM) in DNN, framing the question of DNN architecture design as one of utility specification. This insight results in the design of the new architecture design of ASU-DNN. Herein, the utility of an alternative depends on only its own attributes, as opposed to a fully connected DNN (F-DNN) in which the utility of each alternative is the function of all the alternative-specific variables. We also demonstrate a broader framework of designing DNN by using utility connectivity graph (UCG), in which ASU-DNN is framed as the sparsest model while F-DNN is the richest one. By using statistical learning theory, we demonstrate that this ASU-DNN architecture can reduce the estimation error of F-DNN owing to its sparser connectivity and fewer parameters, although the approximation error of ASU-DNN could be higher. We further apply ASU-DNN, F-DNN, and other nine benchmark ML classifiers to predict travel mode choice by using two datasets, referred to as SGP and TRAIN in this study. The SGP dataset was collected in Singapore in 2017, and the TRAIN dataset is initially from the mlogit package in R. Both the datasets focus on the travel mode choice as the dependent variable. Our results demonstrate that ASU-DNN exhibits consistently higher prediction accuracy than F-DNN and

the other nine classifiers in predicting travel mode choice over the whole hyperparameter space. The alternative-specific connectivity design in ASU-DNN can also be considered as a domain-knowledge-based regularization, unlike generic-purpose regularization methods such as explicit and implicit regularization methods and other architectural hyperparameters. Our results reveal that the domain-knowledge-based regularization is more effective than the generic-purpose regularization methods in terms of improvements in the prediction performance. Finally, we interpret the substitution pattern of the travel mode alternatives in ASU-DNN by using sensitivity analysis and demonstrate that ASU-DNN is also more interpretable than F-DNN owing to its more regular and intuitive choice probability functions. Overall, the prior knowledge of alternative-specific utility function can be used to simultaneously address the three challenges of DNN applications by compromising generic-purpose DNN and domain-specific behavioral knowledge, improving the predictive power and interpretability of "black box" DNN, and functioning as an effective domain-knowledge-based regularization method.

Broadly, this study indicates a new research direction of injecting behavioral knowledge into DNN to adjust DNN architectures specifically for choice analysis. This direction advances domain-specific behavioral knowledge to DNN models, as opposed to the other direction of simply applying various DNNs to choice analysis adopted by most recent studies in the transportation domain. Our research direction is feasible because the behavioral knowledge used in traditional choice modeling has a counterpart in DNN architecture. Specifically, the substitution pattern between alternatives can be controlled by the connectivity of DNN architecture, and vice versa. From an ML perspective, behavioral knowledge can function as domain-knowledge-based regularization, which could better fit domain-specific tasks than generic-purpose regularizations do. Because the alternative-specific utility function is only a minute piece of behavioral knowledge among many, future studies could examine others to explore and create more noteworthy DNN architectures for choice analysis.

The paper is organized as the follows: The next section reviews relevant studies about DNN's applications, interpretability, and regularization methods. Section

3 focuses on three theoretical aspects: the relationship between RUM and DNN, architecture design of ASU-DNN, and estimation and approximation error tradeoff between ASU-DNN and F-DNN. Section 4 focuses on the experiments, discussing the prediction accuracy, effectiveness of domain-knowledge-based regularization, and interpretability of ASU-DNN. Section 5 presents the conclusions of our study.

3.2 Literature Review

Individual decision-making has been an important topic in many domains, including marketing [45], economics [85], transportation [13, 130], biology [116], and public policy [23]. In recent years as ML models permeated into these domains, researchers started to use various classifiers to analyze how individuals take decisions [101, 65]. In the transportation domain, Karlaftis and Vlahogianni (2011) [65] summarized the transportation fields in which DNN models are used, including (1) traffic operations (such as traffic forecasting and traffic pattern analysis); (2) infrastructure management and maintenance (such as pavement crack modeling and intrusion detection); (3) transportation planning (such as in travel mode choice and route choice modeling); (4) environment and transport (such as air pollution prediction); (5) safety and human behavior (such as accident analysis); and (6) air, transit, rail, and freight operations. Recently, many studies applied SVM, decision tree (DT), RF, and DNN to predict travel behavior, automobile ownership, traffic accidents, traffic flow, or even travelers' decision rules [106, 98, 113, 101, 26, 104, 82, 149, 132]. However, nearly all of these studies apply certain generic-purpose ML models to solve domain-specific transportation problems, but none of them explored how domain-specific knowledge could be used to improve generic-purpose ML models for specific tasks.

The balance between generic-purpose DNN classifiers and domain-specific knowledge is also a general challenge to the application of DNN to any specific domain. On the one hand, DNN is effective owing to its generic-purpose and automatic feature learning capacity [76, 14]. For example, the hyperparameters and architecture in feedforward neural network such as ReLU activation functions can be widely used

regardless of the differences between natural language processing (NLP), image recognition, and travel behavioral analysis [73, 125]. On the other hand, a few studies indicate that handcrafted features could still aid in constructing DNN models [78]. In fact, certain domain-specific knowledge is generally involved in DNN modeling. For example, the use of max pooling layer or data augmentation in CNN relies on our domain-specific understanding of images, such as their invariance properties [43].

Another challenge to DNN application is DNN's lack of interpretability, which is caused by its complex model assumptions [80, 34]. The interpretability of DNN is particularly important for reasons such as safety, transparency, trust, and construction of new knowledge [39, 22]. The majority of the ML studies applied to the transportation field focus exclusively on prediction, which is valid because ML models were initially designed for prediction [95, 107, 144, 98, 48]. Prediction-driven ML models differ significantly from the classical choice models, which are both predictive and interpretable [85]. However, to describe DNN as totally a "black-box" may be biased because many recent studies have demonstrated various methods of interpreting DNN. These methods could be categorized broadly into two: ex-ante interpretation [108], which improves interpretability before model building and post-hoc interpretation, which focuses on extracting information after model training [34]. For example, CNN can be interpreted in a post-hoc manner by visualizing the semantic contents in image recognition tasks [150]. In choice analysis, it appears feasible to post-hoc interpret DNN [15, 88, 140]. However, how to introduce prior knowledge into DNN in an ex-ante manner remains unclear.

Even only for prediction, it is significantly challenging to design effective regularization methods and DNN architectures. The regularization methods in DNN consist of explicit and implicit ones, and recent studies reveal that explicit regularizations such as l_1 and l_2 penalties may not effectively aid in the generalization of DNN [148]. New DNN architectures could also aid in improving DNN performance. Recent studies either manually design new architectures (such as AlexNet [73], GoogleNet [125], and ResNet [52]) or automatically search for novel architectural design by using Gaussian process, reinforcement learning, or other sequential modeling techniques

[122, 62, 152, 37]. However, most architecture designs are ad hoc explorations without systematic guidance, and the final DNN architecture identified through automatic searching is not interpretable.

3.3 Theory

3.3.1 Random Utility Maximization and Deep Neural Network

There are two types of inputs in choice modeling: alternative-specific variables x_{ik} and individual-specific variables z_i. Using travel mode choice as an example: x_{ik} could be the price of different travel modes, and z_i represents individual characteristics, such as income and education. $i \in \{1, 2, ...N\}$ is the individual index, and $k \in \{1, 2, ...K\}$ is the alternative index. Let $B = \{1, 2, ...K\}$ and $\tilde{x}_i = [x_{i1}^T, ..., x_{iK}^T]^T$. The output of choice modeling is individual i's choice, denoted as $y_i = [y_{i1}, y_{i2}, ...y_{iK}]$. Each $y_{ik} \in \{0, 1\}$ and $\sum_k y_{ik} = 1$. RUM assumes that the utility of each alternative is the sum of the deterministic utility V_{ik} and random utility ϵ_{ik}:

$$U_{ik} = V_{ik}(z_i, \tilde{x}_i) + \epsilon_{ik} \tag{3.1}$$

Individuals select the maximum utility out of K alternatives. The probability that individual i selects alternative k is

$$P_{ik} = Prob(V_{ik} + \epsilon_{ik} > V_{ij} + \epsilon_{ij}, \forall j \in B,\ j \neq k) \tag{3.2}$$

Assuming that ϵ_{ik} is independent and identically distributed across individuals and alternatives and that the cumulative distribution function of ϵ_{ik} is $F(\epsilon_{ik})$, the choice probability

$$P_{ik} = \int \prod_{j \neq k} F_{\epsilon_{ij}}(V_{ik} - V_{ij} + \epsilon_{ik}) dF(\epsilon_{ik}) \tag{3.3}$$

The following two propositions demonstrate how DNN and RUM are related. The

proof of the two propositions is available in Appendix I.

Proposition 4 *Suppose ϵ_{ik} follows the Gumbel distribution, with probability density function equals to $f(\epsilon_{ik}) = e^{-\epsilon_{ik}} e^{-e^{\epsilon_{ik}}}$ and cumulative distribution function equals to $F(\epsilon_{ik}) = e^{-e^{\epsilon_{ik}}}$. Then, the choice probability P_{ik} takes the form of the Softmax activation function*

$$P_{ik} = \frac{e^{V_{ik}}}{\sum_j e^{V_{ij}}} \tag{3.4}$$

The proof is available in many choice modeling textbooks [130, 13].

Proposition 5 *Suppose that Equation 3.3 holds and that choice probability P_{ik} takes the form of Softmax function as in Equation 3.4. If ϵ_{ik} is a distribution with the transition complete property, ϵ_{ik} follows the Gumbel distribution, with $F(\epsilon_{ik}) = e^{-\alpha e^{-\epsilon_{ik}}}$.*

The proof is available in lemma 2 of McFadden (1974) [85].

Propositions 4 and 5 illustrate the close relationship between RUM and DNN. When F-DNN is applied to the inputs $\tilde{x}_i$ and z_i, the implicit assumption is of RUM with a random utility term following the Gumbel distribution. The inputs into the Softmax function in the DNN could be interpreted as utilities of alternatives. The Softmax function itself could be considered as a soft method of comparing utility scores. The DNN transformation prior to the Softmax function could be considered as the process of specifying utilities.

Formally, V_{ik} in F-DNN follows:

$$V_{ik} = V(z_i, \tilde{x}_i) = w_k^T \Phi(z_i, \tilde{x}_i) = w_k^T (g_m ... \circ g_2 \circ g_1)(z_i, \tilde{x}_i) \tag{3.5}$$

m is the number of layers of DNN; $g_l(t) = ReLU(W_l^T t)$ and $ReLU(t) = max(0, t)$. It is important to note that $V_{ik} = V(z_i, \tilde{x}_i)$ implies that the utility of an alternative k is the function of the attributes of *all* the alternatives $\tilde{x}_i$ and the decision maker's socio-economic variables z_i. Equation 3.5 illustrates that V_{ik} becomes alternative-specific only in the final layer prior to the Softmax function when w_k is applied to $\Phi(z_i, \tilde{x}_i)$.

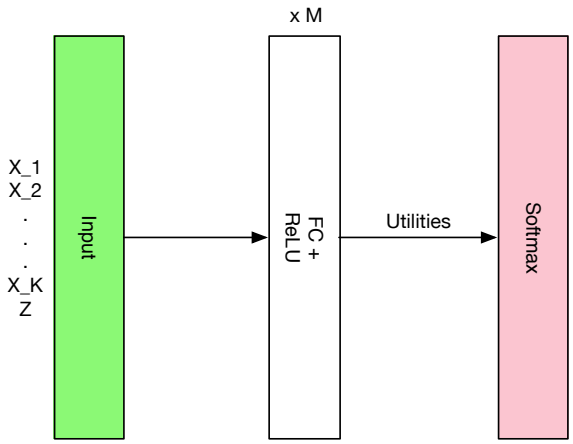


Figure 3-1: Fully Connected Feedforward DNN (F-DNN)

3.3.2 Architecture of ASU-DNN

This utility insight enables us to design a DNN architecture with alternative-specific utility function, which is commonly assumed in choice models. Figure 3-2 shows the architecture of ASU-DNN. Herein, each alternative-specific x_{ik} and individual-specific z_i undergo transformation first, and z_i enters the pathway of x_{ik} after M_1 layers. As a result, the utility of each alternative becomes only a function of its own attributes x_{ik} and of the decision maker's socio-demographic information z_i. This ASU-DNN dramatically reduces the complexity of F-DNN, while still capturing the heterogeneity of the utility function, which varies with the decision makers' socio-demographics. ASU-DNN could be considered as a stack of K subnetworks, interacting with socio-demographics z_i. In addition, this alternative-specific utility is equivalent to the constraint of independence of irrelevant alternative (IIA) in this DNN setting. This is because the ratio of the choice probabilities of two alternatives no longer depends on other irrelevant alternatives. Formally, the utility function in ASU-DNN becomes

$$V_{ik} = V(z_i, x_{ik}) = w_k^T \Phi(z_i, x_{ik}) = w_k^T (g_{M_2} ... \circ g_2 \circ g_1)((g_{M_1}^{x_k} ... \circ g_1^{x_k})(x_{ik}), (g_{M_1}^{z} ... \circ g_1^{z})(z_i)) \tag{3.6}$$

This ASU-DNN architecture can potentially address the three challenges mentioned at the beginning of this work. First, this architecture is a compromise between domain-specific knowledge and a generic-purpose DNN model. On the one hand,

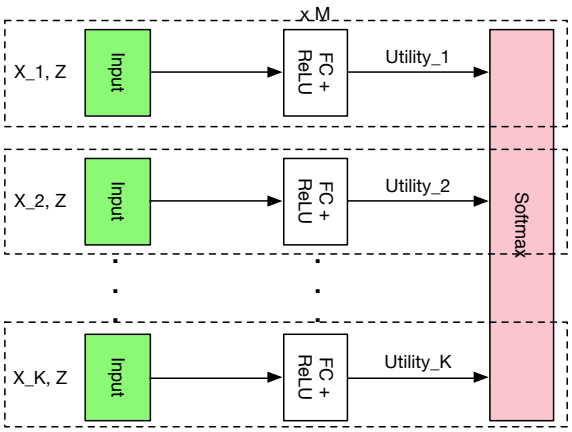
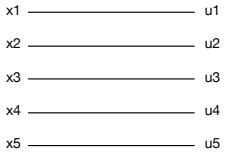


Figure 3-2: DNN with Alternative-Specific Utility Functions (ASU-DNN)

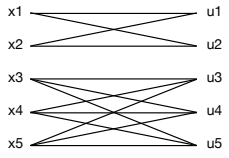
the design permits only alternative-specific connectivity based on the utility theory, whereby the meta-architecture is handcrafted. On the other hand, the fully connected layers in ASU-DNN exploit the automated feature learning capacity of DNN. Therefore, the sub-network in ASU-DNN still uses the power of DNN as a universal approximator [30, 59, 58]. Secondly, this alternative-specific connectivity design renders ASU-DNN more interpretable than F- DNN owing to the underlying utility theory. The two architectures in Figures 3-1 and 3-2 are associated with different behavioral mechanisms. F-DNN implies that the utility of each alternative depends on the other alternatives. A good example is the reference-dependent utilities: when people use the market average price as a reference point, the utility of an alternative depends directly on other alternatives [141, 33]. Meanwhile, the baseline utility theory indicates that the utility of an alternative depends on only the attributes of that alternative. Hence the comparison between the two architectures could be considered as a test between two behavioral mechanisms. Thirdly, F-DNN has substantially more parameters than ASU-DNN does. When both the DNN architectures have 10 layers and approximately 600 neurons in each layer, F-DNN has approximate three million parameters, whereas ASU-DNN has 0.5 million. Therefore, the alternative-specific connectivity design could be considered as a sparse architecture that regularizes DNN models.

3.3.3 DNN Design Guided by Utility Connectivity Graph

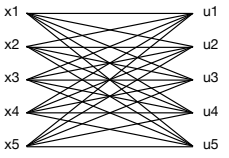
ASU-DNN and F-DNN can be framed under a unified framework about utility connectivity graph (UCG), which describes the global connection between alternative-specific variables and utilities. Figure 3-3 illustrates the idea of UCG by using five alternatives as an example. In Figure 3-3, Figure 3-3a describes the meta-architecture of ASU-DNN as in Figure 3-2, and Figure 3-3c describes the meta-architecture of F-DNN as in Figure 3-1. By taking a network perspective, the meta-architectures of ASU-DNN and F-DNN can be visualized by a fully disconnected network in Figure 3-3d and a fully connected network in Figure 3-3f. In both networks (Figure 3-3d and 3-3f), the existence of an edge betweeen node i and j implies the connection between the variables specific to i (j) and the utility of alternative j (i). With this graphical perspective, we can see some middle ground existing between the UCG of the ASU-DNN and F-DNN. For example, Figure 3-3b and 3-3e are a disconnected graph with two components, representing the meta-architecture of a DNN with full utility connection in each component (nest). In Figures 3-3b and 3-3e, the utilities of all the alternatives are specific to the nests, so they are called the DNN with nest-specific utility (NSU-DNN).



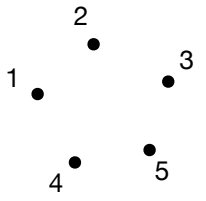
(a) Utility Connection of ASU-DNN



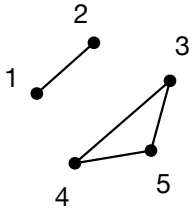
(b) Utility Connection of NSU-DNN



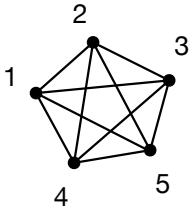
(c) Utility Connection of F-DNN



(d) Visualization of UCG (ASU-DNN)



(e) Visualization of UCG (NSU-DNN)



(f) Visualization of UCG (F-DNN)

Figure 3-3: Visualization of Utility Connectivity Graph for Three Architectures

The three examples of UCG in Figure 3-3 are associated with the following adjacency matrices. With the grpahical perspective, we can see that UCG is very generic because the graph can become weighted and directed, rather than unweighted and undirected. Correspondingly, the adacency matrices can take values between 0 and 1, and become asymmetric. To fully implement this UCG in DNN, we need to use Group-LASSO to control the connectivity between variables and utilities.

$$A_{ASU-DNN} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad A_{F-DNN} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad A_{NSU-DNN} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

UCG controls the substitution pattern between the alternatives; particularly, the substitution patterns of ASU-DNN, NSU-DNN, and F-DNN are very similar to the classical models of multinomial logit model, nested logit model, and mixed logit model with fully correlated random utility errors. Take as an example the models with five alternatives in Figure 3-3. In ASU-DNN, the ratio of the choice probabilities 1 and 3 is $P_1/P_3 = f(x_1, x_3; z)$, which is not a function of any irrelevant alternatives $(2, 4, 5)$, so ASU-DNN is similar to the IIA constraint in MNL. In NSU-DNN, the ratio of the choice probabilities 1 and 2 is $P_1/P_2 = f(x_1, x_2; z)$ and the ratio of the choice probabilities 1 and 3 is $P_1/P_3 = f(x_1, x_2, ..., x_5; z)$, so NSU-DNN is similar to the classical NL model. By the same reasoning, the F-DNN has a more flexible substitution pattern, similar to the mixed logit model with fully correlated random utility errors. However, it is beyond the scope of this study to demonstrate exactly why they are similar. Furthermore, we will focus on only ASU-DNN and F-DNN as the running examples in the following discussions.

3.3.4 Estimation and Approximation Error Tradeoff Between ASU-DNN and F-DNN

It is not true that ASU-DNN could always outperform F-DNN. This is because any constraint applied to F-DNN can potentially cause misspecification errors. Let $\mathcal{F}_1$ and $\mathcal{F}_2$ denote the model family of ASU-DNN and F-DNN; use $\hat{f}_1$ and $\hat{f}_2$ to denote the estimated decision rules from ASU-DNN and F-DNN, and f^* to denote the true data generating process (DGP). The *Excess error* is:

$$\mathbb{E}_S[L(\hat{f}) - L(f^*)] = \mathbb{E}_S[L(\hat{f}) - L(f_F^*)] + \mathbb{E}_S[L(f_F^*) - L(f^*)], \quad \mathcal{F} \in \{\mathcal{F}_1, \mathcal{F}_2\}; \hat{f} \in \{\hat{f}_1, \hat{f}_2\} \tag{3.7}$$

where $L = \mathbb{E}_{x,y} l(y, f(x))$ is the expected loss function and S represents the sample $\{x_i, y_i\}_1^N$. $f_F^* = \underset{f \in \mathcal{F}}{\operatorname{argmin}}\, L(f)$, the best function in function class $\mathcal{F}$ to approximate f^*. The excess error measures the average out-of-sample performance difference between the estimated function $\hat{f}$ and the true model f^*. The excess error can be decomposed as an *estimation error*

$$\mathbb{E}_S[L(\hat{f}) - L(f_F^*)] \tag{3.8}$$

And an *approximation error*

$$\mathbb{E}_S[L(f_F^*) - L(f^*)] \tag{3.9}$$

Formally, the statistical learning theory could demonstrate that ASU-DNN outperforms F-DNN owing to the smaller estimation error of ASU-DNN. However, F-DNN could possibly outperform ASU-DNN owing to the smaller approximation error of F-DNN. When ASU-DNN and F-DNN have equal width and depth, the approximation error of ASU-DNN ($\mathcal{F}_1$) is larger:

$$\mathbb{E}_S[L(f_{\mathcal{F}_1}^*) - L(f^*)] \geq \mathbb{E}_S[L(f_{\mathcal{F}_2}^*) - L(f^*)], \quad \mathcal{F}_1 \subset \mathcal{F}_2 \tag{3.10}$$

This is intuitive because $f^*_{\mathcal{F}_1}$ also belongs to model family $\mathcal{F}_2$ and thus $f^*_{\mathcal{F}_2}$ could outperform $f^*_{\mathcal{F}_1}$ in terms of approximating the true model f^*. A more challenging question is regarding the estimation errors, the proof of which relies on the empirical process theory that uses Rademacher complexity as an upper bound.

Definition 3 *Empirical Rademacher complexity of function class $\mathcal{F}$ is defined as:*

$$\hat{\mathcal{R}}_n(\mathcal{F}|_S) = \mathbb{E}_\epsilon \sup_{f\in\mathcal{F}} \frac{1}{N}\sum_{i=1}^{N} \epsilon_i f(x_i) \tag{3.11}$$

ϵ_i is the Rademacher random variable, taking values $\{-1, +1\}$ with equal probabilities.

Proposition 6 *The estimation error of an estimator $\hat{f}$ can be bounded by the Rademacher complexity of $\mathcal{F}$.*

$$\mathbb{E}_S[L(\hat{f}) - L(f^*_F)] \leq 2\mathbb{E}_S\hat{\mathcal{R}}_n(\mathcal{F}|_S) \tag{3.12}$$

Definition 3 provides a measurement for the complexity of the function class $\mathcal{F}$. Proposition 6 implies that the estimation error is controlled by the complexity of $\mathcal{F}$. This is consistent with traditional wisdom that the estimation error increases when the number of parameters in a model is larger. Details of Definition 3 and Proposition 6 are available in recent studies about the statistical learning theory [136, 139, 10].

Proposition 7 *Let H_d be the class of neural network with depth D over the domain $\mathcal{X}$, where each parameter matrix W_j has the Frobenius norm at most $M_F(j)$, and with ReLU activation functions. Then*

$$\hat{\mathcal{R}}_n(\mathcal{F}|_S) \leq \frac{(\sqrt{2\log(D)}+1)\sqrt{\frac{1}{N}\sum_{i=1}^{N}||x_i||^2}}{\sqrt{N}} \times \prod_{j=1}^{D} M_F(j) \tag{3.13}$$

Remarks on Proposition 7:

1. As this result is from Golowich et al. (2017) [42], so its proof is omitted in this study. Other relevant proofs are available in [10, 92, 3].

2. Proposition 7 indicates that the estimation error of DNN is a function of the depth D, Frobenius norm of each layer $M_F(j)$, diameter of x, and sample size N.

3. Unlike traditional results based on VC-dimension [134, 8], this upper bound relies on the norm of coefficients in each layer, which can be controlled by l_1 or l_2 regularizations, rather than the number of parameters.

4. Suppose the width of DNN is T and each entry in W_j is at most c. The upper bound of F-DNN ($\mathcal{F}_2$) in Proposition 7 can be re-expressed as:

$$\hat{\mathcal{R}}_n(\mathcal{F}_2|_S) \leq \frac{(\sqrt{2\log(D)}+1)\sqrt{\frac{1}{N}\sum_{i=1}^{N}||x_i||^2}}{\sqrt{N}} \times c^D T^D \tag{3.14}$$

Proposition 8 *Suppose ASU-DNN has a total depth D over the domain $\mathcal{X}$, wherein each entry in the matrix W_j is at most c and the width $T = KT_x$. K is the number of alternatives in each choice scenario and T_x is the width of each sub-network* [1]*. With ReLU activation functions*

$$\hat{\mathcal{R}}_n(\mathcal{F}_1|_S) \leq \frac{(\sqrt{2\log(D)}+1)\sqrt{\frac{1}{N}\sum_{i=1}^{N}||x_i||^2}}{\sqrt{N}} \times \frac{c^D T^D}{K^{D/2}} \tag{3.15}$$

Remarks on Proposition 8:

1. Proposition 8 can be derived from Proposition 7 by plugging in the coefficient matrix of each layer in ASU-DNN.

2. The estimation error of ASU-DNN ($\mathcal{F}_1$) shrinks by a factor of $O(K^{D/2})$ compared to F-DNN ($\mathcal{F}_2$), implying that ASU-DNN performs better than F-DNN as K or D increases.

Equations 3.7-3.15 constitute the formal method for illustrating the tradeoff between ASU-DNN and F-DNN. Owing to its sparse connectivity, ASU-DNN has smaller

[1] This assumption simplies the ASU-DNN by omitting the socioeconomic inputs, because adding socioeconomic inputs into this proposition does not change our main conclusion.

estimation error as its main advantage, particularly when K is large, as shown in Equation 3.15. Meanwhile, the larger approximation error could be the main disadvantage of ASU-DNN. When the alternative-specific utility constraint is not true in reality, this constraint could be excessively restrictive, resulting in a low model performance. This problem is also commonly acknowledged in the field of choice modeling, although framed in a different way. Because the alternative-specific utility function in this DNN setting is similar to the IIA constraint, the large approximation error of ASU-DNN could be equivalently framed as a problem of IIA being too restrictive. This drawback appears unavoidable in the approach wherein DNN's interpretability is improved ex-ante. This is because any prior knowledge may be too restrictive in reality. However, compared to classical choice modeling methods that rely exclusively on handcrafted feature learning, misspecification in ASU-DNN is less problematic because it is robust to utility specification *conditioning on* the alternative-specific utility constraint. In addition, Equations 3.14 and 3.15 indicate that the estimation error gap between ASU-DNN and F-DNN could reduce as the sample size increases. Overall, the trade-off between ASU-DNN and F-DNN involves complex dynamics between true models, sample size, number of alternatives, and regularization strength. To compare their performance, we need to apply them to real choice datasets.

3.4 Setup of Experiments

3.4.1 Datasets

Our experiments are based on two datasets, an online survey data collected in Singapore with the aid of a professional survey company and a public dataset in R mlogit package. They are referred to as SGP and TRAIN, respectively, in this study. The SGP survey consisted of a section of choice preference and a section for eliciting socioeconomic variables. At the beginning, all the respondents reported their home and working locations and present travel mode. After obtaining the geographical information, our algorithm computed the walking time, waiting time, in-vehicle travel time,

and travel cost of each travel mode based on the origin and destination provided by the participants and the price information collected from official data sources in Singapore. The SGP and TRAIN datasets include $8,418$ and $2,929$ observations. In the SGP dataset, the output y_i represents the travel mode choice among walking, public transit, driving, ride sharing, and autonomous vehicles (AV); alternative-specific inputs x_{ik} are the attributes of each travel mode, such as price and time cost; and individual-specific inputs z_i are the attributes of decision-makers, such as their income and education backgrounds. In the TRAIN dataset, y_i represents the binary travel mode choice between two different types of trains; the alternative-specific input x_{ik} represents the price, time cost, and level of comfort; and no z_i exists for the TRAIN dataset. Both of the datasets are divided into training, validation, and testing sets in the ratio $4:1:1$. Five-fold cross-validation is used for the model selection, and the model evaluation is based on both the validation and testing sets.

3.4.2 Hyperparameter Space and Searching

A challenge in the comparison between the two DNN architectures is the large number of hyperparameters, on which the performance of DNN largely depends. To address this challenge, we specified the hyperparameter space and searched randomly within this space to identify the DNN configurations with a high prediction accuracy [16]. The empirical risk minimization (ERM) is

$$\min_{w} E(w, w_h) = \min_{w} \frac{1}{N} \sum_{i}^{N} l(y_i, P_{ik}; w, w_h) + \gamma ||w||_p \tag{3.16}$$

in which w represents parameters; w_h represents hyperparameters; $l()$ is the cross-entropy loss function, and $\gamma||w||_p$ represents l_p penalty. Suppose w^* minimizes $E(w, w_h)$ conditioning on one specific w_h. By randomly sampling $w_h^{(s)}$, we could identify the best hyperparameter w_h^*

$$w_h^* = \underset{w_h \in \{w_h^{(1)}, w_h^{(2)}, \ldots, w_h^{(S)}\}}{\operatorname{argmin}} E(w^*, w_h) \tag{3.17}$$

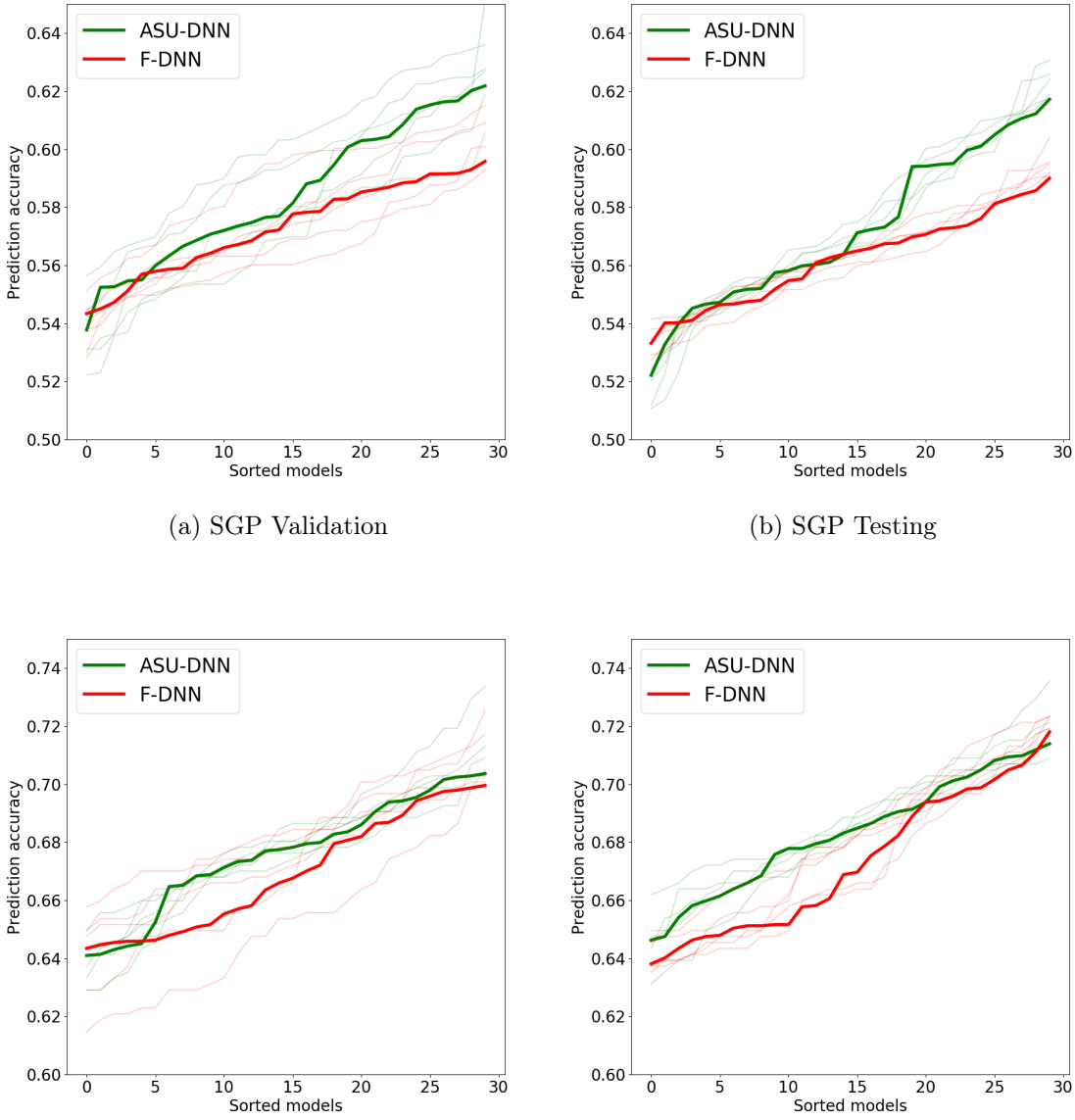
Appendix II summarizes the details of the hyperparameter space and the value ranges of all the hyperparameters. The hyperparameters consist of invariant ones, varying ones specific to F-DNN or ASU-DNN, and varying ones shared by F-DNN and ASU-DNN. The difference between F-DNN and ASU-DNN is referred to as alternative-specific connectivity hyperparameter, which plays a similar role as the other hyperparameters do because it changes the architecture of DNN, controls the number of parameters, and performs regularization. In total, 100 DNN models were trained, 50 each for the two DNN architectures.

3.5 Experiment Results

The result section consists of three parts. The first part compares the prediction accuracy of ASU-DNN, F-DNN, and the other nine ML classifiers. The second part evaluates how effective the alternative-specific connectivity is as a regularization method, as opposed to other generic-purpose regularization methods. The final part compares ASU-DNN and F-DNN in terms of their interpretability by visualizing their choice probability functions. The first part uses both SGP and TRAIN datasets, and the second and third parts focus only on the SGP dataset for simplicity.

3.5.1 Prediction Accuracy

Figure 3-4 summarizes the prediction accuracy of the top 30 models in the validation and the testing sets in the SGP and TRAIN datasets. All the four figures illustrate that ASU-DNN performs better than F-DNN does, although there are marginal differences between the SGP and TRAIN datasets. In the SGP dataset, the prediction accuracy of ASU-DNN in the first 15 out of the visualized 30 models is approximately 0.5% higher than that of F-DNN. Moreover, the difference in prediction accuracy increases as the models' prediction accuracy increases. The top 10 ASU-DNNs outperform the top 10 F-DNNs by approximately $2-3\%$ prediction accuracy in both validation and testing sets. The best ASU-DNN outperforms the best F-DNN by approximately 3%. In the TRAIN dataset, whereas the ASU-DNN still consistently



(a) SGP Validation

(b) SGP Testing

(c) TRAIN Validation

(d) TRAIN Testing

Figure 3-4: Hyperparameter Searching Results

outperforms F-DNN, the gap is smaller in its top 10 models. The first 15 out of the visualized 30 ASU-DNN models outperform the F-DNN models by $2-3\%$ of prediction accuracy, whereas the top 10 ASU-DNNs outperform F-DNN by only 0.5%. An outlier case is the top 1 model in the testing set of TRAIN; herein, the prediction accuracy of F-DNN is marginally higher than that of ASU-DNN. Nonetheless, it is evident that in nearly all the cases, ASU-DNN consistently performs higher than F-

DNN does in the whole hyperparameter space. Table 3.1 also illustrates that both F-DNN and ASU-DNN perform better than the other nine ML classifiers, implying that DNN models fit choice analysis tasks very effectively. Because the prediction accuracy gap between ASU-DNN and F-DNN is identified by using random sampling from the hyperparameter space, we could attribute this gain in prediction accuracy to only the alternative-specific connectivity design and not to any other regularization method. In addition, from the perspective of the behavioral test, the better performance of ASU-DNN than F-DNN indicates that the utility of an alternative was computed based on its own attributes rather than the attributes of all the alternatives.

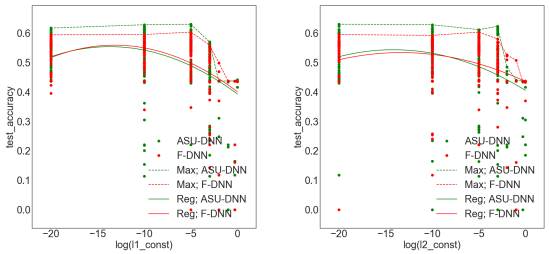
	ASU-DNN (Top 1)	F-DNN (Top 1)	ASU-DNN (Top 10)	F-DNN (Top 10)	MNL (l1_reg)	MNL (l2_reg)	SVM (Linear)	SVM (RBF)	Naive Bayesian	KNN_3	DT	AdaBoost	QDA
Validation (SGP)	62.3%	59.2%	61.3%	58.8%	54.5%	54.7%	54.3%	45.6%	44.7%	58.5%	51.9%	54.6%	47.2%
Test (SGP)	61.0%	58.7%	60.4%	57.6%	52.1%	52.1%	51.8%	44.3%	41.6%	57.9%	50.2%	52.1%	44.9%
Validation (TRAIN)	70.5%	70.1%	69.8%	69.4%	69.5%	69.5%	68.8%	60.9%	57.3%	60.0%	65.0%	67.5%	60.2%
Test (TRAIN)	71.4%	72.1%	71.2%	70.7%	67.8%	67.9%	68.3%	58.7%	56.4%	57.7%	65.0%	69.8%	60.5%

Table 3.1: Prediction accuracy of all classifiers

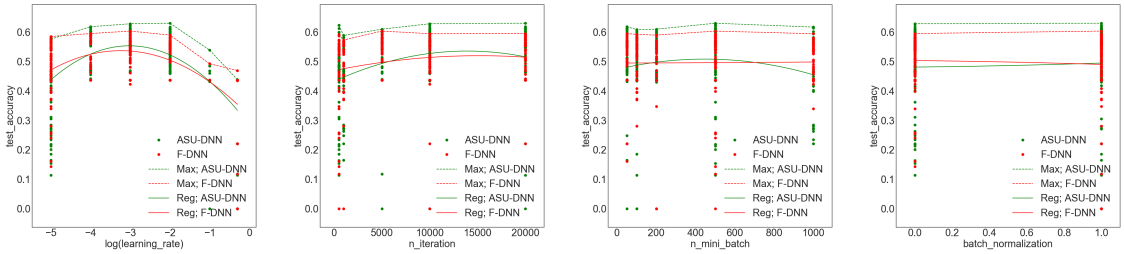
3.5.2 Alternative-Specific Connectivity Design and Other Regularizations

We further examine whether the alternative-specific connectivity hyperparameter is more effective than the other hyperparameters, including explicit regularizations, implicit regularizations, and architectural hyperparameters. Figure 3-5 shows the results, with each of the subfigures depicting the comparison of a hyperparameter with the alternative-specific connectivity hyperparameter.

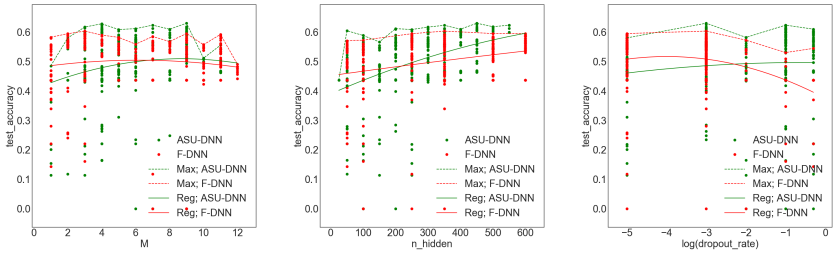
Explicit regularizations. Figures 3-5a and 3-5b show how the prediction accuracy varies with the alternative-specific connectivity hyperparameter and two hyperparameters of explicit regularizations: l_1 and l_2 penalties. The $2-3\%$ prediction accuracy gain by ASU-DNN is retained across the different values of the l_1 and l_2 reg-



(a) l_1 Regularization (b) l_2 Regularization



(c) Learning Rates (d) Number of Iteration (e) Size of Mini Batch (f) Batch Normalization



(g) Depth of DNN (h) Width of DNN (i) Dropout Rates

Figure 3-5: Comparing alternative-specific connectivity to explicit regularizations, implicit regularizations, and architectural hyperparameters

ularizations. When the l_1 penalty is smaller than 10^{-5} and l_2 penalty is smaller than 10^{-3}, ASU-DNN exhibits consistently higher prediction accuracy than F-DNN does. The l_1 and l_2 regularizations fail to aid in achieving a higher prediction accuracy by either ASU-DNN and F-DNN, as illustrated by the nearly flat maximum prediction accuracy curve when l_1 and l_2 values are small and a large decrease in the prediction accuracy as l_1 and l_2 increase, in both Figures 3-5a and 3-5b. In other words, the most commonly used l_1 and l_2 regularizations cannot aid model prediction, or at least they are less effective than the alternative-specific connectivity hyperparameter.

Implicit regularizations. Figures 3-5c, 3-5d, 3-5e, and 3-5f show the relationship between the alternative-specific connectivity hyperparameter and four implicit regularizations: learning rates, number of total iterations, size of mini batch, and batch normalization. These regularization methods are implicit because they are not explicitly used in the empirical risk minimization in Equation 3.16, although they have impacts on model training through the computational process. Again, the prediction accuracy gain owing to the alternative-specific connectivity is highly robust regardless of the values of the other four hyperparameters: in all four figures, the dashed green curves are always placed higher than the dashed red curves are. In Figure 3-5c, both green and red curves assume a marginally concave quadratic form. The learning rates associated with the highest prediction accuracies are between 10^{-3} and 10^{-2}, which are the default values in Tensorflow. This concave quadratic shape is intuitive because highly marginal learning rates are generally inadequate for achieving the optimum values and very large learning rates generally overshoot. In Figures 3-5d, 3-5e, and 3-5f, the dashed and solid curves of both F-DNN and ASU-DNN are nearly horizontal. This indicates that the number of iterations, size of mini batches, and batch normalization are immaterial for improving DNN's prediction accuracy in choice modeling tasks.

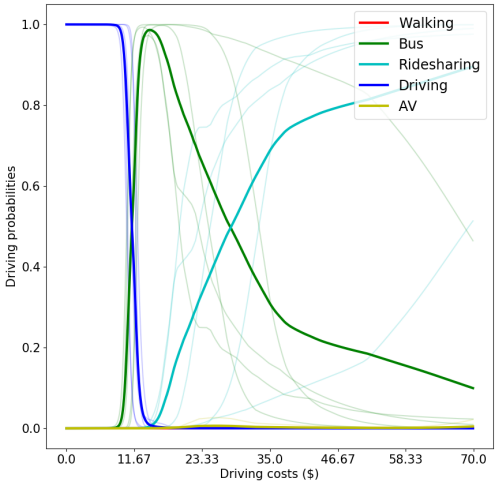
Architectural hyperparameters. Figures 3-5g, 3-5h, and 3-5i compare the alternative-specific connectivity hyperparameter to three architectural hyperparameters: depth and width of DNN, and dropout rates. Similarly, the $2-3\%$ prediction accuracy gain remains over approximately the whole range of the architectural hyperparameters. In Figure 3-5g, the green dashed line is consistently higher than the red dashed line for the majority of the M values (from three to ten). However, this result is not exactly true when the depth of DNN is very small or very large. It is worthnoting that the model performance increases dramatically from one-layer to three-layer ASU-DNN. This indicates that the IIA constraint is less restrictive than the linear specification of each alternative's utility conditioning on the IIA constraint. The maximum and average prediction accuracy of F-DNN form almost horizontal lines everywhere. Finally, in Figure 3-5i, whereas the prediction accuracy difference

remains approximately $2-3\%$ for most of the values of the dropout rate, this difference becomes approximately 10% when the dropout rate is larger than 0.1. The prediction accuracy of ASU-DNN increases marginally as the dropout rates increase, whereas that of F-DNN decreases. These results imply that the alternative-specific connectivity exerts an interaction effect of activating architectural hyperparameters, in addition to its first order effects of $2-3\%$ prediction gain.

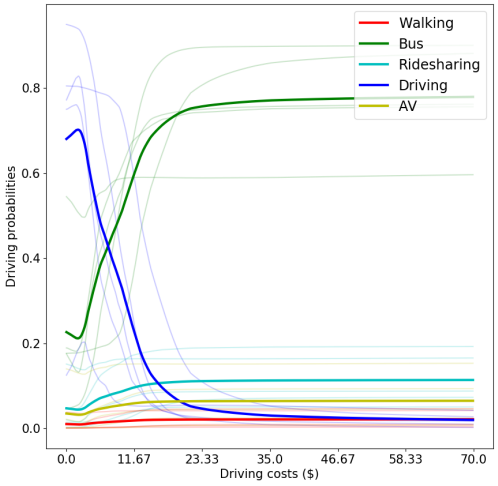
3.5.3 Interpretation of ASU-DNN

Whereas DNN is generally criticized as lacking interpretability, a method for interpreting DNN models is to visualize the choice probability function with respect to inputs. This method has been used in many studies [15, 88, 5, 110, 140]. Figure 3-6 shows how, following this method, the probabilities of selecting five travel modes vary with increasing driving costs in the Top 1 and 10 models, while holding all other variables constant at their empirical mean values.

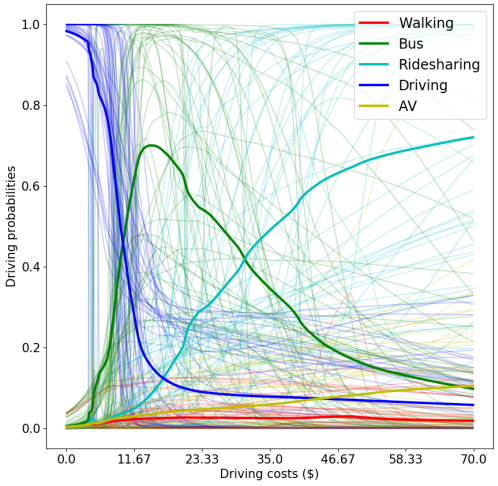
The choice probability functions of ASU-DNN appear more intuitive than those of F-DNN for at least two reasons. The first difference between ASU-DNN and F-DNN is in the probability of selecting driving as the driving costs approach zero. ASU-DNN predicts that individuals exhibit 70% probability of selecting driving when driving costs become zero, whereas F-DNN predicts this probability being close to 100%. The latter value appears unreasonable because all the other variables including driving time is fixed as the mean value of the sample, resulting in the likelihood of the selection of alternative travel modes. The second difference is with regard to the substitution pattern between the five travel modes; specifically, F-DNN predicts that the probability of catching buses will decrease dramatically as the driving cost increases beyond \$15, whereas ASU-DNN predicts that this probability will increase marginally. The substitute effect between driving and catching buses predicted by ASU-DNN appears to be more reasonable because the reason for the dramatic decrease in the probability of selecting buses as the price of driving increases is unclear. Note that the substitution pattern of travel modes in ASU-DNN describes that individuals could switch from driving to the other modes in a proportional manner,



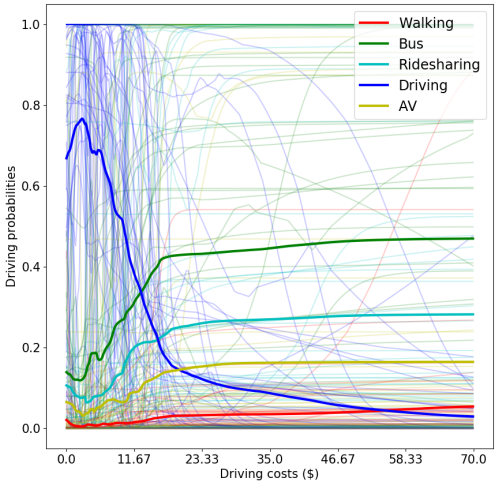
(a) F-DNN (Top 1 Model)



(b) ASU-DNN (Top 1 Model)



(c) F-DNN (Top 10 Models)



(d) ASU-DNN (Top 10 Models)

Figure 3-6: Choice probability functions of ASU-DNN and F-DNN in the SGP testing set

which is similar to a standard multinomial logit model. This regularity in ASU-DNN is caused by the built-in alternative-specific connectivity design.

3.6 Conclusion

This study is motivated by the challenges in the application of DNN to choice analysis, including the tension between domain-specific knowledge and generic-purpose models, and the lack of interpretability and effective regularization methods. In contrast to most of the recent studies in the transportation domain that straightforwardly apply various DNN models to choice analysis, we demonstrate that the benefit could flow in the other direction: from domain knowledge to DNN models. Specifically, it is feasible to inject behavioral insights into DNN architecture owing to the implicit RUM interpretation in DNN. By using the alternative-specific utility constraint, we design a new DNN architecture ASU-DNN, which achieves a certain compromise between domain-specific knowledge and generic-purpose DNN, and between the handcrafted feature learning and automatic feature learning paradigms. This compromise between is significantly effective, as demonstrated by our empirical results that ASU-DNN model is both more predictive and interpretable than F-DNN. ASU-DNN could outperform F-DNN by approximately $2-3\%$ in both validation and testing datasets regardless of the values of DNN's other hyperparameters. The behavioral insights from ASU-DNN are also more reasonable than those from F-DNN, as shown in the choice probability functions of the five travel modes. Theoretically, this alternative-specific utility constraint can be considered as a regularization method. This causes the DNN architecture to be sparser, resulting in a lower estimation error. This insight is supported by our empirical result, because the alternative-specific utility constraint as a domain-knowledge-based regularization is more effective than other explicit and implicit regularization methods, and architectural hyperparameters. In addition, the comparison between ASU-DNN and F-DNN could function as a behavioral test, and our results indicate that individuals are more likely to compute the utility based on an alternative's own attributes rather than the attributes of all the alternatives. This finding is consistent with the long-standing practice in choice modeling.

A caveat is the potentially large approximation error of ASU-DNN, because this constraint could be incorrect in reality. However, it is important to note that this

problem exists in any modeling practice because any prior knowledge could be incorrect. The method of using prior knowledge in ASU- DNN is fundamentally different from that in traditional choice models. ASU-DNN starts with a universal approximator F-DNN as a baseline and "builds downward" F-DNN by using only a piece of prior knowledge (alternative-specific utility in this study) to reduce the complexity of F-DNN. In contrast, traditional choice modeling starts from scratch as a baseline and "builds upwards" a choice model by using all types of prior knowledge (e.g. linearity and additivity of utilities). The former is a significantly more conservative method of using prior knowledge. As a result, the downward-built models are more robust to the function misspecification problem.

Regardless of certain caveats, our results are promising because they present a solution to many challenges in DNN applications. More importantly, it indicates a new research direction of using utility theory to design DNN architectures for choice models, which could become more predictive owing to lower estimation errors and be more interpretable owing to the knowledge introduced into DNN as regularization. We consider that this research direction has immense potential because both utility theory and DNN architectures are exceptionally rich and active research fields. The alternative-specific utility connectivity is only a tiny piece among a vast number of insights in utility theory. Therefore, the immediate next steps could be to use more flexible utility functions (such as those in nested and mixed logit models) to design novel DNN architectures. Future studies could follow this thread of ideas to create more noteworthy and practical DNN architectures for choice analysis.

Appendix I. Proof of Propositions 4 and 5

Proof of Proposition 4. This proof can be found in all choice modeling textbooks [130, 13]. With Gumbel distributional assumption, Equation 3.3 could be solved in an analytical way:

$$\begin{aligned} P_{ik} &= \int_{-\infty}^{+\infty} \prod_{j \neq k} e^{-e^{-(V_{ik}-V_{ij}+\epsilon_{ik})}} f(\epsilon_{ik}) d\epsilon_{ik} \\ &= \int \prod_{j} e^{-e^{-(V_{ik}-V_{ij}+\epsilon_{ik})}} e^{-\epsilon_{ik}} d\epsilon_{ik} \\ &= \int exp(e^{-\epsilon_{ik}} \sum_{j} -e^{-(V_{ik}-V_{ij})}) e^{-\epsilon_{ik}} d\epsilon_{ik} \\ &= \int_{\infty}^{0} exp(-t \sum_{j} e^{-(V_{ik}-V_{ij})}) dt \\ &= \frac{e^{V_{ik}}}{\sum_{j} e^{V_{ij}}} \end{aligned} \tag{3.18}$$

in which the fourth equation uses $t = e^{-\epsilon_{ik}}$. Note this formula in Equation 3.18 is the Softmax function in DNN. V_{ik} is both the deterministic utility in RUM and the inputs into the Softmax function in DNN.

Proof of Proposition 5. This proof can be found in lemma 2 of Mcfadden (1974) [85]. Here is a brief summary of the proof. Suppose that one individual i firstly chooses between alternative k and T alternatives j. Then according to Equations 3.3 and 3.18,

$$\begin{aligned} P_{ik} &= \frac{e^{V_{ik}}}{e^{V_{ik}} + T e^{V_{ij}}} \\ &= \int F(\epsilon_{ik} + V_{ik} - V_{ij})^{T} dF(\epsilon_{ik}) \end{aligned} \tag{3.19}$$

Suppose that the individual i chooses between alternatives k and alternative l in another choice scenario, and alternative l is constructed such that $T e^{V_{ij}} = e^{V_{il}}$. Then

$$
\begin{aligned}
P_{ik} &= \frac{e^{V_{ik}}}{e^{V_{ik}} + e^{V_{il}}} \\
&= \int F(\epsilon_{ik} + V_{ik} - V_{il}) dF(\epsilon_{ik}) \\
&= \int F(\epsilon_{ik} + V_{ik} - V_{ij} - logT) dF(\epsilon_{ik})
\end{aligned}
\tag{3.20}
$$

By construction, Equations 3.19 and 3.20 are equivalent

$$
\int F(\epsilon_{ik} + V_{ik} - V_{ij} - logT) - F(\epsilon_{ik} + V_{ik} - V_{ij})^T dF(\epsilon_{ik}) = 0
$$

Since $F(\epsilon)$ is transition complete, meaning that $\forall a$, $Eh(\epsilon+a) = 0$ implies $h(\epsilon) = 0, \forall \epsilon$, it implies

$$
F(V_{ik} - log\ T) = F(V_{ik})^T, \forall V_{ik}, T
$$

Taking $V_{ik} = 0$ implies $F(-log\ T) = e^{-\alpha T}$. Taking $V_{ik} = log\ T - log\ L$ implies $F(-log\ L) = F(log\ T/L)^T$. Hence $F(log\ T/L) = F(-log\ L)^{1/T} = e^{-\alpha L/T}$. Therefore, $F(\epsilon) = e^{-\alpha e^{-\epsilon}}$. This is the function of Gumbel distribution when $\alpha = 1$.

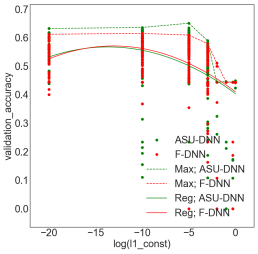
Appendix II. Hyperparameter Space

Hyperparameters	**Values**
Panel 1. Invariant Hyperparameters	
Activation functions	ReLU and Softmax
Loss	Cross-entropy
Initialization	He initialization
Panel 2. Varying Hyperparameters of F-DNN	
M	$[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]$
Width n	$[60, 120, 240, 360, 480, 600]$
Panel 3. Varying Hyperparameters of ASU-DNN	
M_1	$[0, 1, 2, 3, 4, 5, 6]$
M_2	$[0, 1, 2, 3, 4, 5, 6]$
Width n_1	$[10, 20, 40, 60, 80]$
Width n_2	$[10, 20, 40, 60, 80, 100]$
Panel 4. Varying Hyperparameters of F-DNN and ASU-DNN	
γ_1 (l_1 penalty)	$[1.0, 0.5, 0.1, 0.01, 10^{-3}, 10^{-5}, 10^{-10}]$
γ_2 (l_2 penalty)	$[1.0, 0.5, 0.1, 0.01, 10^{-3}, 10^{-5}, 10^{-10}]$
Dropout rate	$[0.5, 0.1, 0.01, 10^{-3}, 10^{-5}]$
Batch normalization	$[True, False]$
Learning rate	$[0.5, 0.1, 0.01, 10^{-3}, 10^{-5}]$
Num of iteration	$[500, 1000, 5000, 10000, 20000]$
Mini-batch size	$[50, 100, 200, 500, 1000]$

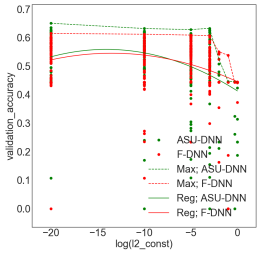
Table 3.2: Hyperparameter space of F-DNN and ASU-DNN

Appendix III. Alternative-Specific Connectivity Design and Other Reglarizations in SGP Validation Set

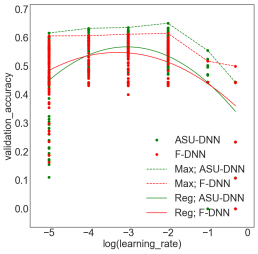
Figure 3-7 compares the alternative-specific connectivity regularization to other regularization methods in the validation set of SGP. The results are very similar to Figure 3-5.



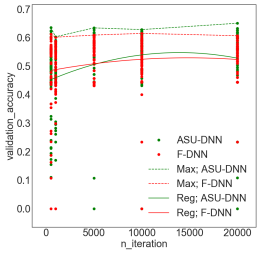
(a) l_1 Regularization



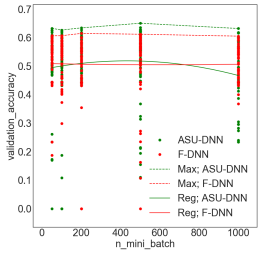
(b) l_2 Regularization



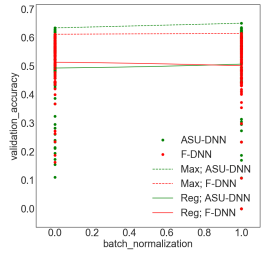
(c) Learning Rates



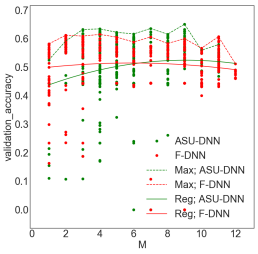
(d) Number of Iteration



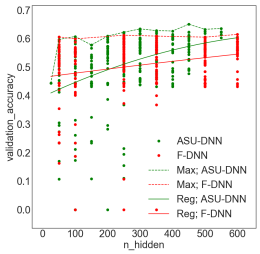
(e) Size of Mini Batch



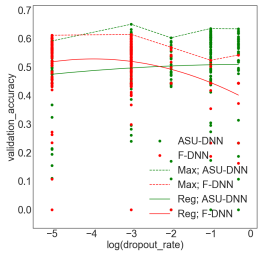
(f) Batch Normalization



(g) Depth of DNN



(h) Width of DNN



(i) Dropout Rates

Figure 3-7: Comparing Alternative-Specific Connectivity to Explicit Regularizations, Implicit Regularizations, and Architectural Hyperparameters in SGP Validation Set

Chapter 4

Essay 3: Theory-Based Deep Residual Neural Networks

4.1 Introduction

To understand individual decision-making, researchers can adopt two disparate modeling paradigms, either theory- or data-driven methods. On the one side, economists have been developing and using decision-making theories to understand individual decision-making for decades [91, 64, 131, 33, 112, 97]. On the other side, many decision-making questions can be simply treated as a classification task, to which any machine learning (ML) classifier can apply. ML classifiers and particularly deep neural networks (DNNs) have been ubiquitously used in various fields like health care, computer vision, language recognition, and many studies in economics, showing extraordinary prediction power and flexibility [72, 65, 73, 76, 40].

Theory- and data-driven models have their own pros and cons, particularly in terms of the tradeoff between prediction power and interpretability. One important weakness of the DNNs is its lack of interpretability, which even machine learning researchers generally admit [72, 80, 34]. In the field of individual decision-making, interpretability is particularly crucial for various important reasons, such as safety in the use of autonomous vehicles, knowledge distillation for academic researchers, and transparency in local governance [80, 34, 39]. As opposed to the "black box" DNNs,

parsimonious decision-making theories are much more interpretable, although they can be misspecified, leading to their relatively low prediction accuracy. The high prediction power and low interpretability of the DNNs and the low prediction power and high interpretability of decision-making theories are complementary. It appears to be a natural question whether researchers can synthesize these two modeling approaches to retain both the high prediction accuracy in DNNs and the high interpretability in decision-making theories. However, since DNNs and decision-making theories emerged from two different research communities (computer science and economics), it is unclear whether this synthesis is even possible, let alone the question of jointly improving prediction accuracy and interpretability in the context of decision-making analysis.

To fill these research gaps, this study designs a theory-based residual neural network (TB-ResNet), demonstrating that the synthesis of DNNs and decision-making theories is not only feasible but also desirable owing to the joint improvement of model prediction and interpretation. First, as a premise of introducing TB-ResNets, we recount the results in McFadden (1974) [85], showing that DNN has an implicit utility maximization interpretation. Second, by using an analogy to ResNet, which consists of an identity and a DNN feature mapping, we create the framework of TB-ResNets, which linearly combines the handcrafted utility specification from decision-making theories and the automatically learnt utility specification from DNNs. The two parts are trained in a sequential way, first the decision-making theory and then the DNNs, so that the decision-making theory can retain a large amount of information to stablize the local pattern of the TB-ResNets. Third, three instances of TB-ResNet are designed based on the choice modeling reasoning (CM-ResNet) for common choice analysis, the prospect theory (PT-ResNet) for risk preference, and the hyperbolic discounting (HD-ResNet) for time preference. The relationship between TB-ResNets, CM-ResNet, PT-ResNet, and HD-ResNet are visualized in Figure 4-1. To empirically test the effectiveness of the TB-ResNets, we applied the CM-ResNet, PT-ResNet, and HD-ResNet to three datasets. Our results show a considerable improvement of all three TB-ResNets than the CM, PT and HD models and the DNNs in terms of

the out-of-sample model prediction, the interpretation of utility functions, and the robustness to adversarial attacks.

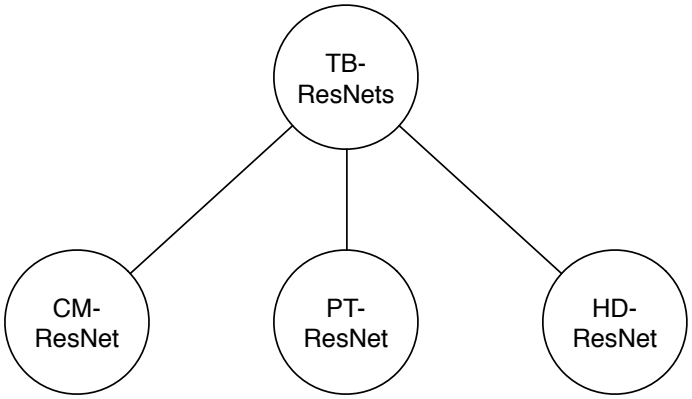


Figure 4-1: Relationship Between TB-ResNets, CM-ResNet, PT-ResNet, and HD-ResNet

The study makes the following contributions. First, while relatively straightforward, it is the first study that demonstrates the feasibility of using DNNs to tackle generic decision-making problems. Compared to other ML classifiers, DNNs are closer to the classical choice modeling due to the framework of random utility maximization (RUM). Second, TB-ResNets represent one way of combining theory- and data-driven methods, achieving higher prediction power and interpretability than either pure theory- or data-driven methods. TB-ResNets are very generic, since many other decision-making problems can be framed in the same way, as long as they are under the utility maximization framework.

The next section reviews related studies. Section 3 connects DNNs to RUM and introduces the TB-ResNets and its three instances. Section 4 discusses the implementation of our experiments. Section 5 presents the results, and Section 6 concludes our findings.

4.2 Literature Review

For decades, individual decision-making has been a classical research question in economics, transportation, and marketing. There are three types of prevalent decision-

making models: discrete choice models that analyze decision-making between several alternatives, prospect theory models that analyze decision-making under risk and uncertainty, and hyperbolic discounting models that analyze temporal decision-making. In terms of choice modeling (CM), McFadden (1974) developped the seminal multinomial logit model based on random utility maximization and applied the model to travel behavioral analysis [85]. After McFadden (1974), several generations of researchers refined the multinomial logit model by incorporating more statistical concerns about heterogeneity, endogeneity, and more complicated substitution patterns [129, 130, 13]. In terms of risk preference, Neumann and Morgenstein [91] created expected utility (EU) model to analyze how individuals make decisions with risky inputs. Kahneman and Tversky [64, 131] created prospect theory (PT), which addresses the abnormality that cannot be explained by the initial EU model [91, 105, 4, 124]. In the recent two decades, researchers gradually improved these models by specifying the formulation of reference points or adding more interactions between attributes and probabilities [127, 33, 71]. In terms of time preference, models have also been developed for decades, including exponential discounting (ED) [112], hyperbolic discounting (HD) [83], quasi-hyperbolic discounting (QHD) [96] and many others. Given the prevalence of individual decision-making across a massive number of fields, the three types of theories have been widely applied to analyze technology adoption, fuel economy, travel behavior, policy decisions, insurance premium, procrastination, and self-control [25, 94, 81, 97, 66].

These decision-making questions can be also treated as a classification task from the perspective of machine learning (ML). Some researchers have started to explore how ML relates to the classical econometrics methods and economics questions. For example, researchers argued that ML classifiers can be used to establish an upper bound of prediction accuracy in decision-making tasks [24]. Sometimes, ML methods are simplified to L_1 or L_2 regularizations [102, 135]. Many studies also compare the prediction accuracy of the traditional decision-making models and the ML classifiers, and typically researchers conclude that ML classifiers such as DNN and random forest can outperform the traditional models. This type of research widely exists in the

transportation domain, analyzing the choice of travel modes or car ownership [95, 54, 144, 26, 27, 65, 128, 98], and also exists in economics, analyzing the demand of general consumers [6]. However, the investigation into how decision-making questions relate to DNNs is mainly limited to the comparison of prediction accuracy. Given the importance of decision-making as one foundation of economics questions and DNNs as one extraordinary powerful tool in the large family of ML classifiers, it is an imperative to demonstrate how to adopt the DNN perspective to address decision-making questions beyond prediction.

Despite DNNs' high prediction accuracy, DNNs are often perceived as lacking interpretability. This problem is partially unsurprising because DNNs were initially designed to maximize the prediction power. DNNs and decision-making theories focus on the difference between prediction and interpretation, or equivalently, on predicting $\hat{y}$ and estimating $\hat{\beta}$ [90]. Model interpretation is important for companies to explain their predictions to users to build up trusts; for governments to make decisions transparent to citizens; and even for any model user to gain new knowledge and generate new insights [80, 39, 34]. While recently many methods have been created to improve the interpretability of DNNs [55, 108, 36, 5, 126], researchers still agree that interpretability is generally missing [72, 80].

DNNs also lack robustness, implying that many attacks are available to generate adversarial instances to fool the DNNs that predict accurately. Szegedy et al. (2014) [126] firstly demonstrate that DNNs do not have local generalization: adversarial examples are perceived as the same as initial pictures by human but are labeled wrongly by DNNs with high confidence. To attack a DNN, several methods have been created, including fast gradient sign method (FGSM); 2) one-step target gradient sign method (TGSM); 3) basic iterative method; and 4) iterative least-likely class method, as illustrated in [44, 99, 100, 74, 75]. To make the DNNs more robust, researchers also designed adversarial training methods to counter the attacks. The adversarial training methods include defensive distillation [100], adversarial training by using both clean and adversarial examples [74], minimax formulation of robust optimization [84], and the input gradient regularizations in training [110]. While all these methods are

valid and effective robust training methods, we seek to improve the robustness and interpretability of the DNNs by using domain-specific decision-making theories.

4.3 Theory

4.3.1 Theory-Based Residual Neural Networks (TB-ResNets)

The V_{ik} could be specified by DNN ($V_{DNN,ik}$) or decision-making theories ($V_{T,ik}$). The DNN utility specification is:

$$V_{DNN,ik}(z_i, \tilde{x}_i) = w_{m,k}^T \Phi(z_i, \tilde{x}_i) = w_{m,k}^T (g_{m-1} ... \circ g_2 \circ g_1)(z_i, \tilde{x}_i) \tag{4.1}$$

m is the number of layers of DNN; $g_l(t) = ReLU(W_l^T t)$; and $ReLU(t) = \max(0, t)$. The utility specification $V_{DNN,ik}$ in DNNs enable a process of automatic learning, relying on the superior approximation power of DNNs [59, 58, 30]. On the contrary, $V_{T,ik}$ represents the process of handcrafting utility specification based on decision-making theories. Therefore, DNNs and decision-making theories are similar in that both have implicit or explicit utility maximization assumption, but they are different since decision-making theories rely on handcrafted features while DNN automatically learns utility specification.

To combine the two types of utility specification, we create theory-based residual neural networks (TB-ResNets), which are composed of one theory-based feature mapping $V_{T,ik}(z_i, \tilde{x}_i)$ and one feedforward DNN $V_{DNN,ik}(z_i, \tilde{x}_i)$. Formally,

$$V_{TB-ResNet,ik} = (V_{T,ik} + \delta V_{DNN,ik})(z_i, \tilde{x}_i) \tag{4.2}$$

where $V_{T,ik}$ represents the utility function of decision-making theories and $V_{DNN,ik}$ represents that from DNN. In Equation 4.2, δ adjusts the weight between the theory-based and DNN utility functions. Therefore, TB-ResNets can be seen as a linear combination of two model families with a flexible weighting controlled by δ.

This TB-ResNet model needs to be trained in a two-stage manner: train $V_{T,ik}$ on

the first stage and $V_{DNN,ik}$ on the second stage. This two-stage procedure allows the TB-ResNet to find the best theory-based utility function on the first stage and to fit the unexplained utility residuals on the second. This two-stage procedure appears more reasonable than a simultaneous training procedure. Since a DNN is a universal approximator [59, 58, 30], $\hat{V}_{DNN,ik}$ can potentially capture all the valuable information that could have been explained by $\hat{V}_{T,ik}$, when $V_{T,ik}$ and $V_{DNN,ik}$ are jointly trained. The simultaneous training would damage the interpretability of the theory-based utility function $V_{T,ik}$ designed in TB-ResNet, and also the capacity of $V_{T,ik}$ stablizing the whole input space.

Our model is named as theory-based residual neural network because it is similar to the deep residual neural network (ResNet). The ResNet consists of one identity feature mapping and one feedforward DNN feature mapping

$$V_{ResNet,ik} = (V_{I,ik} + V_{DNN,ik})(z_i, \tilde{x}_i) \tag{4.3}$$

where $V_{I,ik}$ represents an identity feature mapping $V_{I,ik}(x) = x$. Intuitively, as the true model is close to a linear specification, this ResNet specification might approximate the true model better than a feedforward DNN. This reasoning is similar to our reasoning of using theory-based utility function as the first stage training, because the classical utility theory may have captured the valuable information in the true model. This ResNet also implicitly involves a "two-stage" training procedure because $V_{I,ik}$ is fixed in the training process. Hence the training of ResNet is the same as the second-stage training in the TB-ResNet. The ResNet has been empirically shown and theoretically proved as a more efficient DNN architecture than feedforward DNN [52, 79].

By its design, it is expected that this TB-ResNet can improve the prediction accuracy and the interpretability of both DNNs and decision-making theories. Compared to a pure theory-driven model, the insights from theory is maintained in the $V_{T,ik}$ of the TB-ResNet, and the second-stage training of TB-ResNet can improve the prediction accuracy since the DNN part of the TB-ResNet can address the misspeci-

fication error that commonly exists in any theory-based model. The second-stage of TB-ResNet can also improve the interpretability by augmenting the $\delta V_{DNN,ik}$ utility function to $V_{T,ik}$, rendering the final utility function richer than the simple $V_{T,ik}$. Compared to a DNN model that does not have regular local information, TB-ResNet is more interpretable owing to the theory-based utility function, which stablizes the local information for the whole input domain. Therefore, TB-ResNets should be more robust than DNNs regarding the adversarial examples, since it would be more difficult to create adversarial examples in the local region of each instance for the TB-ResNets when the local information is stablized by the theory-based utility function in TB-ResNets. TB-ResNets are also likely to improve the prediction accuracy of DNNs, when the decision-making theory is informative in capturing the true behavior. In short, as TB-ResNets combine decision-making theories and DNNs, we expect TB-ResNet to incorporate the strength from both sides and address the weaknesses for each other.

4.3.2 Three Instances of TB-ResNets

Choice Modeling Based Resisual Neural Networks (CM-ResNets)

TB-ResNets can take different forms in different contexts. In the choice modeling setting, we use the typical linear utility specification in choice modeling practice as the theory part of TB-ResNets. Replacing the $V_{T,ik}$ in Equation 4.2 by $V_{CM,ik}$ and removing index i from all functions for simplicity, the CM-ResNet is formulated as

$$V_{CM-ResNet,k} = (V_{CM,k} + \delta V_{DNN,k})(z, \tilde{x}) \tag{4.4}$$

$$V_{CM,k} = w_{x_k} x_k + w_z z \tag{4.5}$$

where w_{x_k} represents the parameters for the alternative-specific variables and w_z represents the paarameter for the individual-specific variables. Despite its simplicity, this linear specification of $V_{CM,k}$ is widely used in the choice modeling practice.

Prospect Theory Based Residual Neural Networks (PT-ResNets)

In the risk preference setting, we use prospect theory as the theory part of TB-ResNets. Replacing $V_{T,ik}$ in Equation 4.2 by $V_{PT,ik}$, the PT-ResNet is formulated as

$$V_k = V_{PT-ResNet,k} = (V_{PT,k} + \delta V_{DNN,k})(z, \tilde{x}) \tag{4.6}$$

$$V_{PT,k} = \sum_j v(x_{kj})\pi(p_{kj}) \tag{4.7}$$

$$v(x_{kj}) = \begin{cases} x_{kj}^r & x_{kj} \geq 0 \\ -\lambda(-x_{kj})^r & x_{kj} \leq 0 \end{cases} \tag{4.8}$$

$$\pi(p_{kj}) = e^{-(-ln\ p_{kj})^\alpha} \tag{4.9}$$

$$\alpha = \alpha(z) = \alpha_0 + z'w_{\alpha z} \tag{4.10}$$

$$r = r(z) = r_0 + z'w_{rz} \tag{4.11}$$

$$\lambda = \lambda(z) = \lambda_0 + z'w_{\lambda z} \tag{4.12}$$

where j is the index of uncertain monetary payoffs; x_{kj} is the monetary payoff for alternative k at the value indexed by j; p_{kj} is the probability of x_{kj}; α represents the probability weighting factor; r represents the concavity of value functions; λ represents the loss aversion factor; α, r, and λ are individual specific and can be partially explained by socioeconomic variables z. The specification in Equations from 4.6 to 4.12 is basically the same as Tanaka et al. (2010) [127]. [1]

[1]There are two slight differences between our PT model and Tanaka et al. (2010): the initial paper used a non-parametric method to estimate individuals' risk preference parameters and sequentially estimated the coefficients $w_{\alpha z}$, w_{rz}, and $w_{\lambda z}$, while our PT model follows a parametric method and simultaneously estimates all the coefficients.

Hyperbolic Discounting Based Residual Neural Networks (HD-ResNets)

Similarly, another instance of TB-ResNets is the hyperbolic discounting residual network (HD-ResNet) for time preference, formulated as following.

$$V_k = V_{HD-ResNet,k} = (V_{HD,k} + \delta V_{DNN,k})(z, \tilde{x}) \tag{4.13}$$

$$V_{HD,k} = \sum_j x_{kj} \beta e^{-r t_{kj}} \tag{4.14}$$

$$\beta = \beta(z) = \beta_0 + z' w_{\beta z} \tag{4.15}$$

$$r = r(z) = r_0 + z' w_{rz} \tag{4.16}$$

where x_{kj} is the monetary payoff and t_{kj} is the associated time; r is the conventional time discounting factor; β is the present-bias factor; both r and β can be partially explained by socioeconomic variables z. Again the specifications from Equation 4.13 to 4.16 are the same as Tanaka et al. (2010) [127]. While this $V_{HD,k}$ is actually more generic than the hyperbolic discounting model, we will call it hyperbolic discounting (HD) for simplicity throughout our study.

4.4 Experiment Setup

4.4.1 Datasets

This study uses the three datasets in the experiments. The first dataset was collected in Singapore in 2017, focused on the adoption of five travel modes consisting of walking, buses, ridesharing, driving, and autonomous vehicles. The second and third datasets come from Tanaka et al. (2010) [127], focused on the risk and time preference with two alternatives in the choice sets. All three cases collected the data by stated preference surveys. The sample size of the three datasets are respectively $8,418$, $6,335$ and $5,340$. All three datasets are split into the training and testing sets with a ratio of $4:1$.

4.4.2 Training

CM-ResNets, PT-ResNets, and HD-ResNets with varying δ factors are trained and compared to the baseline models CM, PT, HD, and DNN. To implement the δ factor, we used the l_2 norm penalty with the penalty constant equals to λ in the second stage training of TB-ResNets. Hence δ is approximately inversely proportional to λ, although the exact mapping of their magnitudes is unclear. The λ penalty constant varies from the weakest $(1e-10)$ to the strongest (about 1.0) to demonstrate the transition of TB-ResNets from DNN to decision-making theories. To focus on the investigation into the δ (or similarly λ) factor, we fix the DNN architecture in all our experiments, by using these hyperparameters: (1) depth = 3; (2) width = 100; (3) number of iterations = 5000; and (4) size of mini-batch: 100. These hyperparameters are chosen by manual adjustment before our formal experiments.

To train a TB-ResNet, we follow the two-stage training procedure: first stage trained on decision-making theories and the second stage fitting utility residuals. On the first stage, empirical risk minimization (ERM) is formulated as following:

$$\min_{V_{T,ik}\in\mathcal{F}} L(\tilde{x}_i, z_i; w) = \min_{V_{T,ik}\in\mathcal{F}} -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} y_{ik}\log\ \frac{e^{V_{T,ik}(\tilde{x}_i,z_i)}}{\sum_{j=1}^{K} e^{V_{T,ij}(\tilde{x}_i,z_i)}} \tag{4.17}$$

which is the same as maximum likelihood estimation (MLE). $\mathcal{F}$ represents the class of decision-making theories. The second stage is conditioning on $\hat{V}_{T,ik}$ trained on the first stage. With $\mathcal{G}$ representing the class of functions in DNN, second-stage training is:

$$\min_{V_{DNN,ik}\in\mathcal{G}} L(\hat{V}_{T,ik}, \tilde{x}_i, z_i; w) = \min_{V_{DNN,ik}\in\mathcal{G}} -\frac{1}{N}\sum_{i=1}^{N}\sum_{i=1}^{K} y_{ik}\log\ \frac{e^{(\hat{V}_{T,ik}+\delta V_{DNN,ik})(\tilde{x}_i,z_i)}}{\sum_{j=1}^{K} e^{(\hat{V}_{T,ij}+\delta V_{DNN,ij})(\tilde{x}_i,z_i)}} \tag{4.18}$$

4.5 Experiment Results

4.5.1 Comparing Model Performance

Table 4.1 summarizes the model performance of the three groups of models, including choice modeling (CM), prospect theory (PT), and hyperbolic discounting (HD) respectively in three panels. Each panel includes the decision-making models, the DNN models, and the TB-ResNets with varying regularization constants. The regularization constants are chosen to show the compromise between DNNs and decision-making theories. Since the optimum regularization constant is data-specific, the best TB-ResNet for CM, PT, and HD are associated with different regularization constants. The two columns in Table 4.1 report the prediction accuracy and cross-entropy losses, both evaluated in the testing sets.

	Prediction Accuracy (Testing)	Cross-entropy Loss (Testing)
Panel 1. Performance of CM Models		
DNN	55.2%	2.759
CM ResNet (λ = 1e-10)	56.4%	2.820
CM ResNet (λ = 0.005)	57.3%	1.457
CM ResNet (λ = 0.01)	56.8%	1.343
CM	44.7%	1.384
Panel 2. Performance of PT Models		
DNN	87.5%	0.405
PT ResNet (λ = 1e-05)	89.3%	0.305
PT ResNet (λ = 0.0001)	89.0%	0.321
PT ResNet (λ = 0.01)	75.8%	0.545
PT	69.9%	0.584
Panel 3. Performance of HD Models		
DNN	72.8%	0.853
HD ResNet (λ = 1e-05)	75.9%	0.449
HD ResNet (λ = 0.001)	78.5%	0.491
HD ResNet (λ = 0.01)	57.3%	0.685
HD	55.5%	0.687

Table 4.1: Performance of CM, PT, and HD Models

First, before analyzing the performance of the TB-ResNets, Table 4.1 demonstrates that the DNN models dramatically outperform decision-making theories in terms of prediction accuracy, although it is not necessarily the case when evaluated by cross-entropy losses. In the three panels in Table 4.1, the prediction accuracy of DNNs is 55.2%, 87.5%, and 72.8%, which are respectively 10.5%, 17.6%, and 17.3% higher than the corresponding models based on decision-making theory (CM, PT, and HD). This higher prediction accuracy of DNNs is consistent with the previous studies, which found the better performance of DNNs in a variety of scenarios [95, 144, 26, 65, 98]. However, DNNs do not necessarily outperform decision-making theories in terms of cross-entropy losses. In fact, the cross-entropy loss of DNNs is larger than the CM and HD models, and smaller than only the PT model. The inconsistency between prediction accuracy and cross-entropy loss can be caused by some DNN predictions that are correct but have relatively low confidence, since the cross-entropy loss takes into account the probability value of the correct label.

Second, the CM-ResNets, PT-ResNets, and HD-ResNets can achieve a flexible compromise between decision-making theories and DNN models in terms of prediction accuracy and cross-entropy losses. On the one hand, the CM-ResNets, PT-ResNets, and HD-ResNets become more similar to DNNs as the regularization constant (λ) decreases. For example, when $\lambda = 1e-10$, the prediction accuracy and cross-entropy loss of the CM-ResNets is significantly close to that of DNN. On the other hand, the CM-ResNets, PT-ResNets, and HD-ResNets become more similar to the CM, PT, and HD models as the regularization constant (λ) increases. For example, when $\lambda = 0.01$, the prediction accuracy and cross-entropy loss of the PT-ResNets are 75.8% and 0.545, closer to the values of the PT model (69.9% and 0.584) than that of the DNNs (87.5% and 0.405), as shown in panel 2. In addition, the transition of the TB-ResNets from the decision-making theories to the DNNs appears not linear. For example, there is a significant decrease of prediction accuracy from the CM-ResNet ($\lambda = 0.01$) to the CM model, while the variation of the DNN models and all the CM-ResNets is very small. Ths transition is also specific to the evaluation metrics. In panel 1, the prediction accuracy significantly decreases from the CM-ResNet ($\lambda = 0.01$) to the CM model,

while the cross-entropy loss significantly decreases from the CM-ResNet ($\lambda = 1e-10$) to the CM-ResNet ($\lambda = 0.005$).

Third, the CM-ResNets, PT-ResNets, and HD-ResNets are better than the linear combination of DNN models and decision-making theories, as the three ResNets can outperform both DNNs and decision-making theories in terms of prediction accuracy and cross-entropy losses at certain values of λ. Specifically, the CM-ResNet ($\lambda = 0.005$) outperforms the DNN and the CM in terms of prediction accuracy, and its cross-entropy loss is significantly lower than the DNN, although slightly higher than the CM. The PT-ResNet ($\lambda = 0.0001$) and the HD-ResNet ($\lambda = 0.001$) outperform the DNN and their corresponding PT and HD models in terms of both prediction accuracy and cross-entropy loss. Therefore, some optimum λ exists for each one panel, although the optimum values are different across the three scenarios.

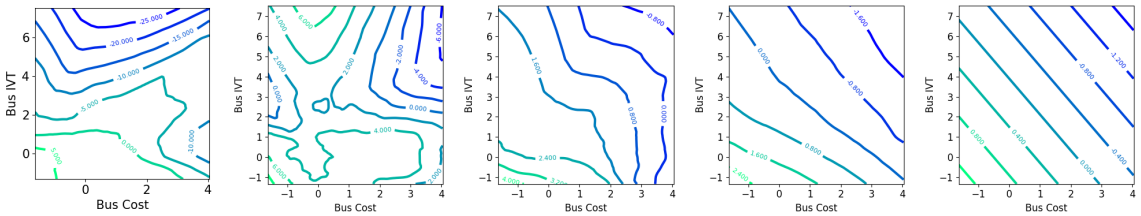
TB-ResNets outperform DNNs because of the localization and norm regularization in TB-ResNets, and TB-ResNets outperform decision-making theories because the second stage training of TB-ResNets addresses the misspecification problem in the CM, PT, and HD models. First, the first stage training in TB-ResNets aids in localizing the training model, and this localization can help the second stage training by reducing the estimation error. One extreme example is that the first stage training has no misspecification error. In this case, the first stage training of TB-ResNets has almost recovered the correct model, and thus TB-ResNets can outperform DNNs owing to the smaller estimation error of decision-making theories. The second-stage training of TB-ResNets is only a *local* searching centered around the decision-making model estimated on the first stage, while the training of a full DNN is a *global* searching in the whole DNN model family. In fact, the decision-making theory is used in the TB-ResNets in a way similar to the Bayesian prior. When the prior knowledge is informative, it helps to improve the model generalizability. Second, the λ term is implemented as the norm regularization for the parameters of the TB-ResNets. With norm regularization, the model complexity of TB-ResNets is better controlled. As to the better performance of TB-ResNets over the decision-making theories, the reason is more straightforward. While the CM, PT, and HD models are all classical models,

they must have misspecification problems since they are specified based on expert knowledge, which is not complete compared to the true data generating process. For example, the PT relies on only three parameters to capture the risk preference and the HD relies on only two parameters to capture the intertemporal decision-making, while real human decision-making can be much more complicated than these parsimonious theories. Since DNNs have been proved to be a universal approximator [59, 58, 30], it can be used to fit the utility residuals to address the misspecification problem in decision-making theories.

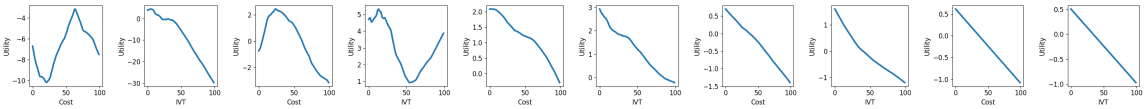
4.5.2 Interpretation of Utility Functions

Due to the implicit utility interpretation in DNNs, TB-ResNets can be seen as a process of automatically augmenting utility functions to handcrafted utility theory (CM, PT and HD). While DNNs have been widely criticized as "black box" in terms of its statistical and optimization properties, it is relatively straightforward to show how inputs and outputs are related by using visualization and sensitivity analysis [88, 140]. Figures 4-2, 4-3, and 4-4 show how utility values relate to the different values of inputs. In Figure 4-2, the five graphs on the upper row visualize how the utility of taking buses varies with the monetary cost (x-axis) and the in-vehicle travel time (y-axis), and the ten graphs on the lower row visualize how the utility varies with the monetary cost (x0) and the in-vehicle travel time (x1) respectively, holding all the other variables constant. The formats of Figures 4-3 and 4-4 are similar to Figure 4-2 except for the meaning of the variables. In Figure 4-3, the monetary payoff and winning probability are the x- and y-axes on the upper row and the x0 and x1 on the lower row. In Figure 4-4, the monetary payoff and time are the x- and y-axes on the upper row and the x0 and x1 on the lower row.

As shown in Figures 4-2a-4-2e, the CM-ResNets as the compromise of the CM and DNN models can address the problem of irregularity in the DNN models and the misspecification in the CM model. On the one side, the utility function of the CM model is very regular, as shown by the decreasing utility values of taking buses with increasing values of bus costs and in-vehicle travel time, as shown by Figures



(a) DNN (55.2%) (b) CM ResNet($\lambda = 1e-10$; 56.4%) (c) CM ResNet($\lambda = 0.005$; 57.3%) (d) CM ResNet ($\lambda = 0.01$; 56.8%) (e) CM (44.7%)



(f) x0 (g) x1 (h) x0 (i) x1 (j) x0 (k) x1 (l) x0 (m) x1 (n) x0 (o) x1

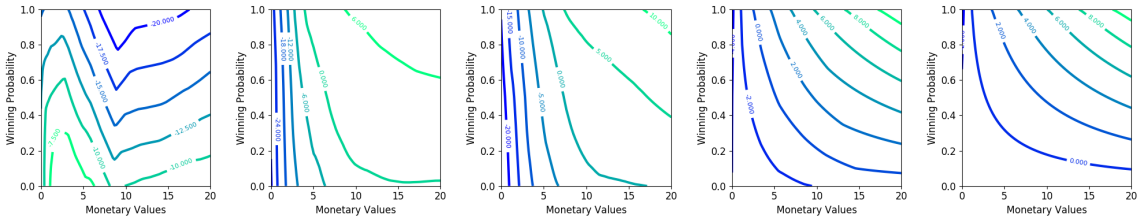
Figure 4-2: Utility Functions of CM-ResNets

4-2e, 4-2n, and 4-2o. While this utility function of CM has regular countours, it is highly likely the true utility function is more complex than the linear specification, leading to the large misspecification error and low prediction accuracy of the CM model. On the other side, the utility function of a DNN model has very irregular countours. Specifically, as shown in Figure 4-2f, the DNN predicts that the utility of using buses firstly decreases and then increases as the cost increases, which is not reasonable. While the DNN model can provide higher prediction accuracy than the CM model, the irregular utility function generates the difficulty of interpreting the DNN model in a reasonable way. However, as opposed to both DNNs and CM models, the CM-ResNets can achieve a desirable compromise between the two models. As the regularization constant λ increases, the utility function of CM-ResNets becomes more regular and similar to the CM model, and as λ decreases, the utility function becomes more irregular and thus similar to DNNs. For example, the CM-ResNet ($\lambda = 0.005$) model is similar to the CM model, since it has a relatively regular utility countour and the utility values decrease as the cost and in-vehicle travel time increase. But unlike the CM model, the CM-ResNet ($\lambda = 0.005$) has richer patterns, which can capture the real decision-making mechanism better than the CM model. The CM-ResNet ($\lambda = 0.005$) model retains the monotonicity in the utility function, which is more

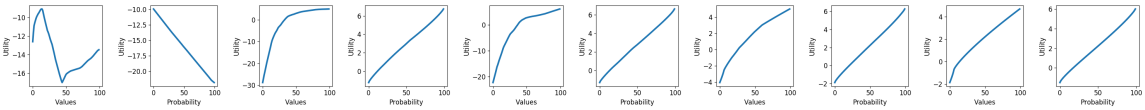
reasonable than the DNN model. Therefore, the CM-ResNet model becomes more interpretable than the DNN owing to the regular local pattern of the utility function.

Both the PT-ResNets and HD-ResNets also present an effective compromise between the decision-making theories and the DNN models, similar to the findings in the CM-ResNets. As shown in Figure 4-3, the utility function of the PT model (Figure 4-3e) is the same as its original theory: the utility function with respect to the monetary value has a concave shape (Figure 4-3n) and the utility function maintains the S-shaped relationship with respect to the winning chances (Figure 4-3o). The bottomline of the PT utility function is its monotonic increasing property, since larger winning rates and monetary payoff leads to a higher probability of choosing the alternative. However, this monotonic increasing property is violated in the DNN model, as shown in Figures 4-3a, 4-3f, and 4-3g, rendering the DNN model impossible to interpret. Unlike both the PT and the DNN models, the PT-ResNet ($\lambda = 0.0001$) retains the monotonicity of the PT model and improves it by allowing a richer pattern augmented by the second-stage training; the PT-ResNet is more interpretable than the DNN model, as its utility function is monotonically increasing with respect to the monetary value and the winning chances (Figure 4-3j and 4-3k). As a reminder, this PT-ResNet ($\lambda = 0.0001$) also achieves higher prediction accuracy than both the DNN and the PT model. In terms of the HD models, the HD-ResNet reveals a similarly successful pattern, showing richer utility patterns than the HD model and more regular and interpretable pattern than the DNN model. The utility function is supposed to increase with higher monetary value and decrease with longer waiting time (more temporal discounting), as shown in Figures 4-4e, 4-4n, and 4-4o. The best HD-ResNet ($\lambda = 0.001$) retains these patterns with richer details revealed, and is more reasonable than the DNN model (Figures 4-4a, 4-4f, and 4-4g), in which the utility of one future payoff increases as the waiting time increases.

There are two reasons why the CM-ResNets, PT-ResNets, and HD-ResNets are more interpretable than the DNN models. The first reason comes from the weakness of DNNs in function estimation, in that DNNs can achieve the same high prediction accuracy with totally different model parameters [19]. As a result, the local utility

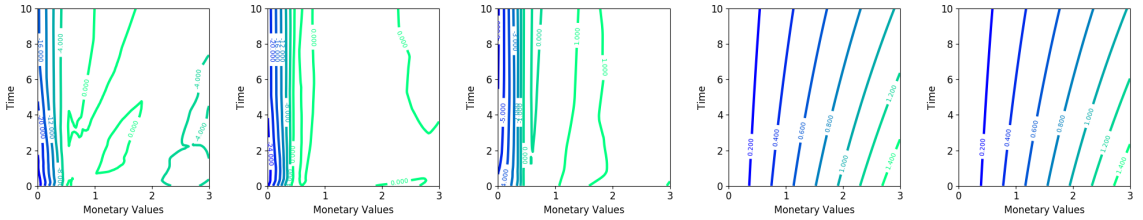


(a) DNN (87.5%) (b) PT ResNet ($\lambda = 1e-5$; 89.3%) (c) PT ResNet ($\lambda = 0.0001$; 89.0%) (d) PT ResNet ($\lambda = 0.01$; 75.8%) (e) PT (69.9%)

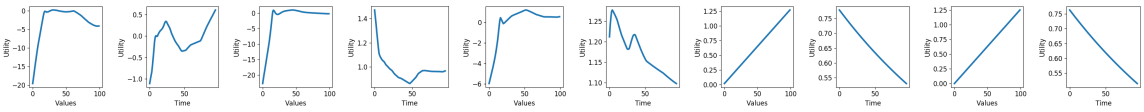


(f) x0 (g) x1 (h) x0 (i) x1 (j) x0 (k) x1 (l) x0 (m) x1 (n) x0 (o) x1

Figure 4-3: Utility Functions of PT-ResNets, PT, and DNNs



(a) DNN (72.8%) (b) HD Resnet ($\lambda = 1e-5$; 75.9%) (c) HD Resnet ($\lambda = 0.001$; 78.5%) (d) HD Resnet ($\lambda = 0.01$; 57.3%) (e) HD (55.5%)



(f) x0 (g) x1 (h) x0 (i) x1 (j) x0 (k) x1 (l) x0 (m) x1 (n) x0 (o) x1

Figure 4-4: Utility Functions of HD-ResNets, HD, and DNNs

pattern of the DNN model can be very irregular even when it achieves the highest prediction accuracy. The second comes from the strength of the TB-ResNets. The utility specification of TB-ResNet equals to $V_T + \delta V_{DNN}$, which is a compromise between decision-making theories and DNNs. The δ factor in TB-ResNets retains the flexibility between decision-making theory and DNN. With only a small amount of utility augmented to V_T, TB-ResNets provide reliable and interpretable utility specification, similar to but still different from the decision-making theories. Intuitively, the decision-making theory is used to stablize the local utility patterns, particularly

when the δ factor is small.

4.5.3 Robustness

Similar to the interpretability discussion, the CM-, PT-, and HD-ResNets become more robust than DNN models. To test the robustness of these models, we used the fast gradient sign method (FGSM) and the target gradient sign method (TGSM) to generate the adversarial examples [44, 74]:

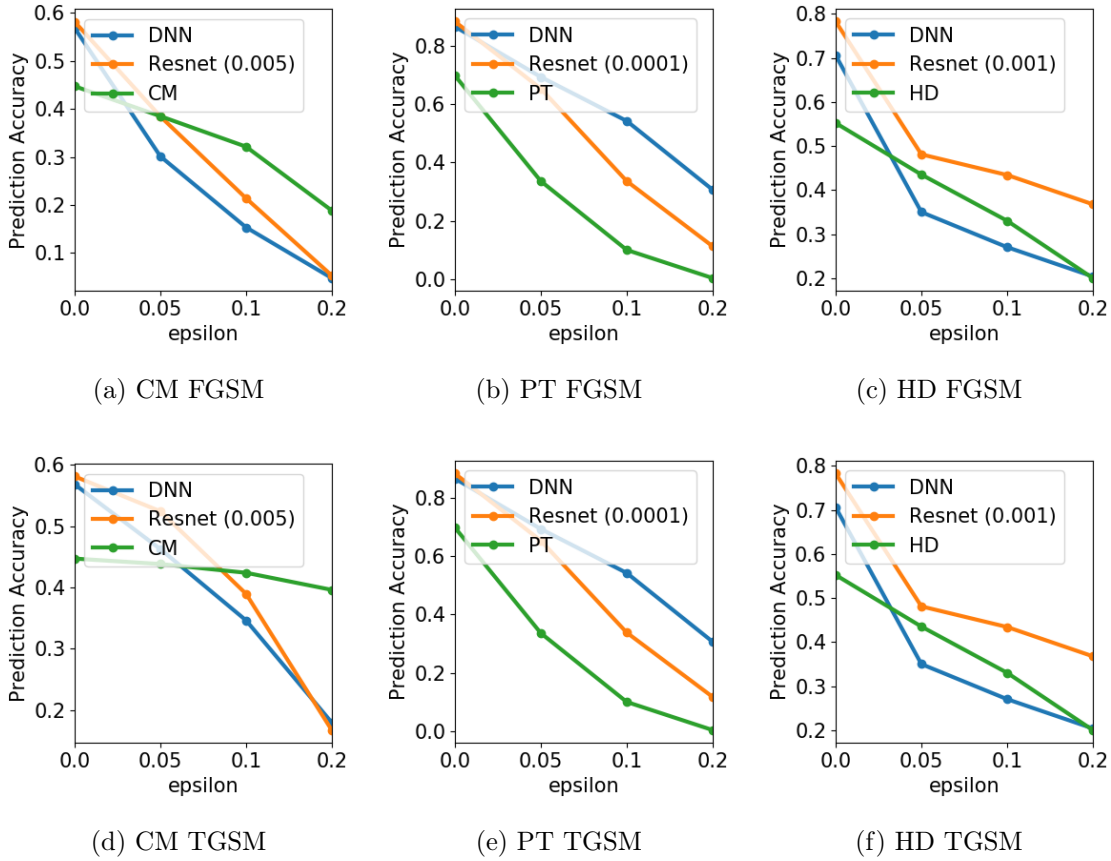
$$FGSM : X_{adv} = X + \epsilon \times sign(\nabla_x L(y, \hat{y})) \tag{4.19}$$

$$TGSM : X_{adv} = X - \epsilon \times sign(\nabla_x L(y_{target}, \hat{y})) \tag{4.20}$$

with varying $\epsilon \in [0.05, 0.1, 0.2]$. Figure 4-5 shows the prediction accuracy of the models, with upper row showing the results of FGSM and the lower row the results of TGSM.

In the two cases of CM and HD models, both CM-ResNets and HD-ResNets are more robust than the DNN models, as shown in Figures 4-5a, 4-5d, 4-5c, and 4-5f. In all these figures, the DNN models are much less robust than both the CM and the HD models, as the prediction accuracy curve of the DNNs decreases much quickly than that of the CM and the HD models. For example, whereas the prediction accuracy of the CM model is much lower than that of the DNN model when $\epsilon = 0$, that of the CM model is much higher than that of the DNN when $\epsilon > 0.1$. As the CM-ResNet and the HD-ResNet models are the combination of decision-making theories and the DNN models, the two ResNets are more robust than DNNs, as shown by the orange curves lying above the blue curves in Figures 4-5a, 4-5d, 4-5c, and 4-5f. This is quite intuitive, since the gradients of the CM-ResNets and the HD-ResNets are much better restricted, as shown in the previous subsection. With smaller gradient values, it is less likely to identify the adversarial examples for the CM-ResNets and the HD-ResNets around a small ϵ region.

The PT results are different from the CM and HD models, and it is because the gradients of PT are categorically different from those of CM and HD. While CM and



(a) CM FGSM (b) PT FGSM (c) HD FGSM

(d) CM TGSM (e) PT TGSM (f) HD TGSM

Figure 4-5: Prediction Accuracy in Adversarial Examples (FGSM and TGSM)

HD theories are designed to have smooth input gradients, PT has very large local gradients. Specifically, the probability weighting function in the PT model can have the gradient value close to infinity as the probability value approaches zero. As a result, the PT model itself is designed to be sensitive to the adversarial examples that are created by using the local gradient information. This can be seen in Figures 4-5b and 4-5e: the PT models not only have smaller prediction accuracy than the DNNs as $\epsilon = 0$, but also decrease much more quickly than the DNNs. As a result, the prediction accuracy of the PT-ResNets decrease faster than the DNNs. While usually this quick decrease of prediction accuracy implies that the system is less robust, here this is caused by the specialty of the PT. In general, the gradients of decision-making theories should be smoother than the DNN models, and thus the robustness of the TB-ResNets should be improved, compared to the DNN models.

4.6 Conclusion

This study tackles the decision-making problem by synthesizing the theory- and data-driven methods. We firstly point out that RUM and DNNs are related, since DNNs have an implicit utility maximization interpretation. This observation enables us to design a TB-ResNet that synthesizes the decision-making theories and DNN models by using a linear combination and a two-stage training procedure. Three instances of TB-ResNets, including CM-ResNets, PT-ResNets and HD-ResNets, are created and tested on three empirical datasets.

The empirical results demonstrates that TB-ResNets are better than both decision-making theories and DNN models in terms of prediction accuracy, interpretability, and robustness, as summarized in Table 4.2. Compared to decision-making theories, TB-ResNets are more predictive since they incorporate the approximation power of DNNs and are more interpretable by augmenting utility functions to enrich the regular but overly simplified utility function in decision-making theories. Compared to DNNs, TB-ResNets are more predictive owing to their localization and regularization offered by the first-stage training and more interpretable and robust since they use the regular utility functions from the decision-making theories to stablize the local information. These findings are consistent across the CM, PT, and HD scenarios.

	CM, PT and HD	**DNN**
Prediction	Significant Improvement (by addressing function misspecification)	Marginal Improvement (by localization and regularization)
Interpretability	Significant Improvement (by augmenting and enriching utility functions)	Significant Improvement (by stablizing local information)
Robustness	NA	Significant Improvement (by stablizing local information)

Table 4.2: Improvement of TB-ResNets Compared to Decision-Making Theories and DNN

TB-ResNets reconcile the handcrafted decision-making theories and the automatically learnt utility specification, which resonates with many recent discussions. The

dominant performance of DNNs is owing to its capacity of automatically learning utility specification, named as "end-to-end" systems that can "learn from scratch", and this power of automation is treated as its main strength over traditional methods that rely on domain knowledge to handcraft features [90, 76]. However on the other side, recently researchers argued that automatic feature learning with zero prior knowledge seems not a viable approach. Liao and Poggio [78] contended that "being lazy is good, but being too lazy is not". In fact, the brittle DNN models indeed need other types of inputs to make it more interpretable and robust. Our results of TB-ResNets provide a tangible evidence for the viability of compromising handcrafted and automatically learnt systems in the context of decision-making analysis.

While these results are promising, readers can improve upon these results in different ways. First, to obtain a more reliable utility specification by TB-ResNets, it involves careful adjustment of regularization to guarantee the robustness of DNN models used in the second-stage training of TB-ResNets, and the ideal way to impose the regularization can be more complicated than the simple L_2 penalty used in this study. Second, the robustness problem in DNNs can be addressed by using robust DNN training methods [93, 44, 74, 110], it is unclear whether our approach that uses domain-specific knowledge is better than the generic robust DNN training methods. Third, while the augmented utility specification by TB-ResNets can effectively adjust the decision-making theory, these results still cannot tell scholars how to revise and improve theories, which are important for the interpretation purpose. TB-ResNets can be treated as a combination of an interpreter (decision-making theories) and a predictor (DNNs). The sequential training approach can be seen as the first step of a game between them. Researchers can further explore how to interact the two parts in a more interactive and automatic manner. Overall, this study points out a direction in which theory- and data-driven methods mutually benefit from each other, by using the inductive data-driven model to enrich utility theory and using prespecified utility theory to regularize complex data-driven models. Given the richness of DNN architectures, utility theories, and currently abundant computational power, we believe these questions will yield more fruitful and interesting research results in the future.

Appendix I: PT and HD Validation

We validated the individual risk and time preference parameters estimated from PT and HD models. Table 4.3 shows the summary statistics of the five parameters. Overall, our results are relatively similar to Tanaka et al. (2010) [127]. The average values of $(\lambda_i, \alpha_i, r_i)$ in PT are $(1.654, 0.918, 0.623)$, as opposed to $(2.63, 0.74, 0.61)$ in Tanaka et al. (2010). The average values of (b_i, r_i) in HD are $(0.586, 0.006)$, as opposed to $(0.82, 0.078)$ in Tanaka et al. (2010). The slight difference could be caused by the difference of the estimation procedures. For example, our training for PT is a parametric estimation applied to the whole population, while Tanaka et al. used a non-parametric method applied to individuals. The distributions of the five individual preference parameters are relatively concentrated around the mean values, and each individual could have different parameters depending on their socio-economic variables. For instance, while the average loss aversion parameter is about 1.65, the max could be 3.1.

	λ_i (PT)	α_i (PT)	r_i (PT)	b_i (HD)	r_i (HD)
count	5068	5068	5068	4272	4272
mean	1.654	0.918	0.623	0.586	0.006
std	0.488	0.080	0.054	0.114	0.006
min	0.748	0.562	0.514	0.319	0.001
25%	1.249	0.893	0.584	0.503	0.002
50%	1.578	0.935	0.62	0.593	0.005
75%	2.000	0.973	0.655	0.667	0.009
max	3.178	1.000	0.815	0.913	0.056

Table 4.3: Summary of Five Parameters in PT and HD Models

Chapter 5

Conclusion and Future Studies

This dissertation examines how to use DNNs for choice analysis. The first essay connects DNNs to classical MNL models with utility interpretation, demonstrating that it is feasible to derive from DNNs the economic information as complete as DCMs. The second essay targets the architecture design perspective of DNNs, demonstrating how to use utility theory to control the global connectivity of the utilities. The third essay designs a TB-ResNet, which synthesizes any utility maximization theory and DNNs. The TB-ResNet improves the prediction accuracy, interpretability, and robustness in comparison to both decision-making theories and DNN models. Detailed contributions have been discussed in each essay, so I only summarize the major contributions of this dissertation at a very high level.

First, this dissertation bridges the perspectives between DNNs and utility theories. While the two types of theories appear to arise from very different backgrounds, this dissertation demonstrates their similarity owing to the implicit utility interpretation in DNNs. As a result, researchers can approach the same choice modeling issue from two different perspectives. For example, the utility specification is equivalent to the architectural design of DNNs, so any new DNN architecture implicitly involves a different utility specification and any new utility specification naturally relates to a DNN architecture. Researchers can use utility theory to guide the architectural design of DNNs, as demonstrated in essay 2, or reversely, design new DNN architectures with high prediction accuracy to provide more insights into utility theory. Hence

the theory building of PT can be treated as a architecture design problem from DNN perspectives, and so are the other behavioral theories. The meta-architecture designed through utility theories is more interpretable than many DNN architectures that are built with focuses on only higher prediction accuracy. Associated with the similarity between utility specification and DNN architectures, many other components are also similar. The soft labels of DNNs are the same as choice probability functions, and elasticities are just the same as the input gradients of DNNs.

Second, this dissertation lays out a new foundation for using DNNs to analyze individual decision-making, by contrasting the *subtractive* perspective in DNNs to the *additive* perspective in classical choice modeling. Traditionally, researchers gradually enrich a baseline model by adding more components and making more assumptions. This additive approach is understandable in the traditional setting because the baseline models are typically too simple to capture the reality. However, for DNN-based choice models, the way to ask questions is opposite. We should ask how to remove rather than enrich the components in order to improve DNNs. This *subtractive* reasoning is exactly opposite to the classical DCMs. Instead of going from simplicity to complexity, researchers need to ask how to reduce rather than improve the heterogeneity of individuals' preference, how to regularize rather than enrich local behavioral patterns, how to impose rather than release monotonicity constraints. The difference between the additive vs. subtractive reasoning can be formally revealed by using statistical learning theory. Classical choice models mainly have large approximation errors, which can only be reduced by enriching models, while DNN-based models mainly have large estimation errors, which can only be reduced by simplifying models. To reduce the large estimation errors, researchers can explore any regularization tool in the DNN framework, including the explicit regularizations such as L_p penalties, the implicit regularizations such as training time and number of iterations, and the architectural design such as incorporating more skip connections. The sources of imposing these constraints typically come from domain knowledge because domain knowledge is typically associated with certain function classes. For locally regular information, researchers can use more robust training methods. It is beyond the

scope of this dissertation to demonstrate all these methods, and future studies should examine whether these methods can work well in the context of choice analysis.

Third, DNN-based choice models follow the paradigm of *prediction-driven interpretation*, rather than *interpretation-driven prediction*, which is the paradigm in classical choice models. The prediction-driven interpretation approach starts with model building that seeks to maximize the prediction accuracy, and then invites interpretation in a post-hoc manner. This approach does not rely on the completeness of experts' knowledge, but mainly the approximation power of prespecified models, such as DNNs. Therefore, it can avoid the limitation of domain experts' knowledge as well as the misspecification errors that plague any model built through domain expertise. However, the process of prediction-driven interpretation does not imply that no domain knowledge is used at all. As discussed in essays 2 and 3, the process still involves certain pieces of domain expertise. For example, essay 2 uses the knowledge about utility connectivity and essay 3 uses the behavioral model as the first stage training. But in both cases, the domain-specific knowledge is used in a much more conservative way than the traditional modeling methods. In both cases, certain automatic learning capacity from DNNs is at least augmented to improve the prediction power. Therefore, prediction-driven interpretation does not preclude the ex-ante use of domain-specific knowledge, but at least partially rely on the extraordinary approximation power of DNN models.

Lastly, each one of the three essays provides very concrete contributions. The first essay provides an empirical recipe about how to derive all the economic information from DNNs. Future studies can use the idea in any choice analysis setting. Essay 1 also functions as a gateway study that points out future directions that researchers can explore to improve DNNs' interpretability for economic information. One possible future research direction based on essay 1 is other interpretable information from DNNs, rather than limiting to only the economic information. The second essay uses the connection between utility specification and DNN architecture. Since ASU-DNN follows the long-standing practice of choice modeling, the behavioral contents follow the IIA constraint and still retain the flexibility owing to the DNNs' approximation

power. Theoretically, the perspective of utility connectivity graph points out the relationship between DNN architectural design and the substitution patterns of alternatives. Future studies can explore how to provide a new theoretical framework to reframe the classical multinomial logit, nested logit, and mixed logit models, through the meta-architectural design perspective. The TB-ResNet designed in essay 3 can be used to combine any decision-making theory and DNN models for a wide range of applications. Note that this TB-ResNet can be treated as one step dynamic between theory- and data-driven methods, and future studies can explore further dynamics between the two parts to simultaneously facilitate the improvement of theory and DNN models.

This dissertation is restricted to (1) the simplest feedforward DNN out of a large family of DNN models, (2) prospect theory and hyperbolic discounting models out of a large family of decision-making theories, (3) one DCM model out of a large family of econometrics models, and (4) one travel mode choice out of a vast number of individual decision-making applications. Varying each component can lead to a different research agenda. For example, future studies can investigate the combination of recurrent neural networks (RNNs), temporal decision-making theory, time-series econometrics models, and sequential consumption choice, answering the question about how to synthesize temporal decision-making rules and RNNs. Future studies can also explore the combination of CNN, panel regression, and paneled locational choice, answering the question how to use CNN to reframe panel regression in the case of housing locational choice. In addition, future studies can also explore using new information formats in choice analysis by including 2D matrix/image, 3D matrix/space/buildings, 1D ordered vector/text, or graphs. Using new information formats appears an appealing research direction because people naturally read texts or images before making decisions. The unstructured data such as images and texts are also the key to unleash the full predictive power of DNNs. It is impossible to enumerate all possible future studies, and I hope future researchers can improve upon this dissertation and identify more interesting perspectives, models, and applications for choice analysis with DNNs.

Bibliography

[1] Agnar Aamodt and Enric Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI communications*, 7(1):39–59, 1994.

[2] Anuradha M Annaswamy, Yue Guan, H Eric Tseng, Hao Zhou, Thao Phan, and Diana Yanakiev. Transactive control in smart cities. *Proceedings of the IEEE*, 106(4):518–537, 2018.

[3] Martin Anthony and Peter L Bartlett. *Neural network learning: Theoretical foundations*. cambridge university press, 2009.

[4] Kenneth Joseph Arrow. *Aspects of the theory of risk-bearing*. Yrjo Jahnssonin Saatio, 1965.

[5] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert MÃžller. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11(Jun):1803–1831, 2010.

[6] Patrick Bajari, Denis Nekipelov, Stephen P Ryan, and Miaoyu Yang. Machine learning methods for demand estimation. *American Economic Review*, 105(5):481–85, 2015.

[7] Bowen Baker, Otkrist Gupta, Nikhil Naik, and Ramesh Raskar. Designing neural network architectures using reinforcement learning. *arXiv preprint arXiv:1611.02167*, 2016.

[8] Peter L Bartlett, Nick Harvey, Chris Liaw, and Abbas Mehrabian. Nearly-tight vc-dimension and pseudodimension bounds for piecewise linear neural networks. *arXiv preprint arXiv:1703.02930*, 2017.

[9] Peter L Bartlett, Michael I Jordan, and Jon D McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

[10] Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

[11] Moshe Ben-Akiva, Michel Bierlaire, Daniel McFadden, and Joan Walker. *Discrete Choice Analysis*. 2014.

[12] Moshe Ben-Akiva, John L Bowman, and Dinesh Gopinath. Travel demand model system for the information era. *Transportation*, 23(3):241–266, 1996.

[13] Moshe E Ben-Akiva and Steven R Lerman. *Discrete choice analysis: theory and application to travel demand*, volume 9. MIT press, 1985.

[14] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[15] Yves Bentz and Dwight Merunka. Neural networks and the multinomial logit for brand choice modelling: a hybrid approach. *Journal of Forecasting*, 19(3):177–200, 2000.

[16] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305, 2012.

[17] James S Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In *Advances in neural information processing systems*, pages 2546–2554, 2011.

[18] Chris M Bishop. Training with noise is equivalent to tikhonov regularization. *Neural computation*, 7(1):108–116, 1995.

[19] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[20] Olivier Bousquet, Stéphane Boucheron, and Gábor Lugosi. Introduction to statistical learning theory. In *Advanced lectures on machine learning*, pages 169–207. Springer, 2004.

[21] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[22] Robert Brauneis and Ellen P Goodman. Algorithmic transparency for the smart city. 2017.

[23] Axel Börsch-Supan and John Pitkin. On discrete choice models of housing demand. *Journal of Urban Economics*, 24(2):153–172, 1988.

[24] Colin F Camerer. Artificial intelligence and behavioral economics. In *Economics of Artificial Intelligence*. University of Chicago Press, 2017.

[25] Colin F Camerer and Howard Kunreuther. Decision processes for low probability events: Policy implications. *Journal of Policy Analysis and Management*, 8(4):565–592, 1989.

[26] Giulio Erberto Cantarella and Stefano de Luca. Multilayer feedforward networks for transportation mode choice analysis: An analysis and a comparison with random utility models. *Transportation Research Part C: Emerging Technologies*, 13(2):121–155, 2005.

[27] Hilmi Berk Celikoglu. Application of radial basis function and generalized regression neural networks in non-linear utility function specification for travel mode choice modelling. *Mathematical and Computer Modelling*, 44(7):640–658, 2006.

[28] Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks. In *Artificial Intelligence and Statistics*, pages 192–204, 2015.

[29] Jonathan D Cohen, Keith Marzilli Ericson, David Laibson, and John Myles White. Measuring time preferences. Technical report, National Bureau of Economic Research, 2016.

[30] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

[31] Yann N Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Advances in neural information processing systems*, pages 2933–2941, 2014.

[32] Juan De Dios Ortuzar and Luis G Willumsen. *Modelling transport*. John Wiley and Sons, 2011.

[33] Sanjit Dhami. *The Foundations of Behavioral Economic Analysis*. Oxford University Press, 2016.

[34] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. 2017.

[35] Yanjie Duan, Yisheng Lv, Yu-Liang Liu, and Fei-Yue Wang. An efficient realization of deep learning for traffic data imputation. *Transportation research part C: emerging technologies*, 72:168–181, 2016.

[36] Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. *University of Montreal*, 1341(3):1, 2009.

[37] Stefan Falkner, Aaron Klein, and Frank Hutter. Bohb: Robust and efficient hyperparameter optimization at scale. *arXiv preprint arXiv:1807.01774*, 2018.

[38] Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim. Do we need hundreds of classifiers to solve real world classification problems. *Journal of Machine Learning Research*, 15(1):3133–3181, 2014.

[39] Alex A Freitas. Comprehensible classification models: a position paper. *ACM SIGKDD explorations newsletter*, 15(1):1–10, 2014.

[40] Edward L Glaeser, Scott Duke Kominers, Michael Luca, and Nikhil Naik. Big data and big cities: The promises and limitations of improved measures of urban life. *Economic Inquiry*, 56(1):114–137, 2018.

[41] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[42] Noah Golowich, Alexander Rakhlin, and Ohad Shamir. Size-independent sample complexity of neural networks. *arXiv preprint arXiv:1712.06541*, 2017.

[43] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[44] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2015.

[45] Peter M Guadagni and John DC Little. A logit model of brand choice calibrated on scanner data. *Marketing science*, 2(3):203–238, 1983.

[46] Maya Gupta, Andrew Cotter, Jan Pfeifer, Konstantin Voevodski, Kevin Canini, Alexander Mangylov, Wojciech Moczydlowski, and Alexander Van Esbroeck. Monotonic calibrated interpolated look-up tables. *The Journal of Machine Learning Research*, 17(1):3790–3836, 2016.

[47] Aurélien Géron. *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems.* " O'Reilly Media, Inc.", 2017.

[48] Julian Hagenauer and Marco Helbich. A comparative study of machine learning classifiers for modeling travel mode choice. *Expert Systems with Applications*, 78:273–282, 2017.

[49] Jerry A Hausman, Jason Abrevaya, and Fiona M Scott-Morton. Misclassification of the dependent variable in a discrete-response setting. *Journal of Econometrics*, 87(2):239–269, 1998.

[50] David Haussler and Philip M Long. A generalization of sauer's lemma. *Journal of Combinatorial Theory, Series A*, 71(2):219–240, 1995.

[51] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[52] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[53] John Paul Helveston, Yimin Liu, Elea McDonnell Feit, Erica Fuchs, Erica Klampfl, and Jeremy J Michalek. Will subsidies drive electric vehicle adoption? measuring consumer preferences in the us and china. *Transportation Research Part A: Policy and Practice*, 73:96–112, 2015.

[54] David Hensher, Jordan Louviere, and Joffre Swait. Combining sources of preference data. *Journal of Econometrics*, 89(1-2):197–221, 1998.

[55] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[56] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[57] Chinh Q Ho, Corinne Mulley, Yoram Shiftan, and David A Hensher. Vehicle value of travel time savings: Evidence from a group-based modelling approach. *Transportation Research Part A: Policy and Practice*, 88:134–150, 2016.

[58] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.

[59] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[60] Xiuling Huang, Jie Sun, and Jian Sun. A car-following model considering asymmetric driving behavior based on long short-term memory neural networks. *Transportation Research Part C: Emerging Technologies*, 95:346–362, 2018.

[61] Neal Jean, Marshall Burke, Michael Xie, W Matthew Davis, David B Lobell, and Stefano Ermon. Combining satellite imagery and machine learning to predict poverty. *Science*, 353(6301):790–794, 2016.

[62] Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. An empirical exploration of recurrent network architectures. In *International Conference on Machine Learning*, pages 2342–2350, 2015.

[63] Patiphan Kaewwichian, Ladda Tanwanichkul, and Jumrus Pitaksringkarn. Car ownership demand modeling using machine learning: Decision trees and neural networks. *International Journal of Geomate*, 17(62):219–230, 2019.

[64] Daniel Kahneman and Amos Tversky. Prospect theory: An analysis of decision under risk. *Econometrica: Journal of the Econometric Society*, pages 263–291, 1979.

[65] Matthew G Karlaftis and Eleni I Vlahogianni. Statistical methods versus neural networks in transportation research: Differences, similarities and some insights. *Transportation Research Part C: Emerging Technologies*, 19(3):387–399, 2011.

[66] Supreet Kaur, Michael Kremer, and Sendhil Mullainathan. Self-control at work. *Journal of Political Economy*, 123(6):1227–1277, 2015.

[67] Been Kim and Finale Doshi-Velez. Interpretable machine learning (icml tutorials). In *International Conference of Machine Learning*, Sydney, 2017.

[68] Been Kim, Cynthia Rudin, and Julie A Shah. The bayesian case model: A generative approach for case-based reasoning and prototype classification. In *Advances in Neural Information Processing Systems*, pages 1952–1960, 2014.

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

[70] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. *arXiv preprint arXiv:1703.04730*, 2017.

[71] Botond Koszegi and Matthew Rabin. A model of reference dependent preferences. *The Quarterly Journal of Economics*, pages 1133–1165, 2006.

[72] Sotiris B Kotsiantis, I Zaharakis, and P Pintelas. Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160:3–24, 2007.

[73] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[74] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

[75] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2017.

[76] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[77] Michel Ledoux and Michel Talagrand. *Probability in Banach Spaces: isoperimetry and processes*. Springer Science Business Media, 2013.

[78] Qianli Liao and Tomaso Poggio. When is handcrafting not a curse? Technical report, 2018.

[79] Hongzhou Lin and Stefanie Jegelka. Resnet with one-neuron hidden layers is a universal approximator. *arXiv preprint arXiv:1806.10909*, 2018.

[80] Zachary C Lipton. The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*, 2016.

[81] Elaine M Liu. Time to change what to sow: Risk preferences and technology adoption decisions of cotton farmers in china. *Review of Economics and Statistics*, 95(4):1386–1403, 2013.

[82] Lijuan Liu and Rung-Ching Chen. A novel passenger flow prediction model using deep learning methods. *Transportation Research Part C: Emerging Technologies*, 84:74–91, 2017.

[83] George Loewenstein and Drazen Prelec. Anomalies in intertemporal choice: Evidence and an interpretation. *The Quarterly Journal of Economics*, 107(2):573–597, 1992.

[84] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[85] Daniel McFadden. Conditional logit analysis of qualitative choice behavior. 1974.

[86] Grégoire Mesnil, Yann Dauphin, Xavier Glorot, Salah Rifai, Yoshua Bengio, Ian Goodfellow, Erick Lavoie, Xavier Muller, Guillaume Desjardins, and David Warde-Farley. Unsupervised and transfer learning challenge: a deep learning approach. In *Proceedings of the 2011 International Conference on Unsupervised and Transfer Learning workshop-Volume 27*, pages 97–111. JMLR. org, 2011.

[87] Hrushikesh Mhaskar, Qianli Liao, and Tomaso Poggio. Learning functions: when is deep better than shallow. *arXiv preprint arXiv:1603.00988*, 2016.

[88] Gregoire Montavon, Wojciech Samek, and Klaus-Robert Muller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15, 2018.

[89] Mikhail Mozolin, J-C Thill, and E Lynn Usery. Trip distribution forecasting with multilayer perceptron neural networks: A critical evaluation. *Transportation Research Part B: Methodological*, 34(1):53–73, 2000.

[90] Sendhil Mullainathan and Jann Spiess. Machine learning: an applied econometric approach. *Journal of Economic Perspectives*, 31(2):87–106, 2017.

[91] J von Neumann and Oskar Morgenstern. Theory of games and economic behavior, 1944.

[92] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In *Conference on Learning Theory*, pages 1376–1401, 2015.

[93] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 427–436, 2015.

[94] Walter Nicholson and Christopher Snyder. Uncertainty and strategy. In *Microeconomics Theory: Basic Principles and Extensions*. 2012.

[95] Peter Nijkamp, Aura Reggiani, and Tommaso Tritapepe. Modelling inter-urban transport flows in italy: A comparison between neural network analysis and logit analysis. *Transportation Research Part C: Emerging Technologies*, 4(6):323–338, 1996.

[96] Ted O'Donoghue and Matthew Rabin. Doing it now or later. *American Economic Review*, pages 103–124, 1999.

[97] Ted O'Donoghue and Matthew Rabin. Choice and procrastination. *The Quarterly Journal of Economics*, 116(1):121–160, 2001.

[98] Hichem Omrani. Predicting travel mode of individuals by machine learning. *Transportation Research Procedia*, 10:840–849, 2015.

[99] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016.

[100] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 582–597. IEEE, 2016.

[101] Miguel Paredes, Erik Hemberg, Una-May O'Reilly, and Chris Zegras. Machine learning or discrete choice models for car ownership demand estimation and prediction? In *Models and Technologies for Intelligent Transportation Systems (MT-ITS), 2017 5th IEEE International Conference on*, pages 780–785. IEEE, 2017.

[102] Alexander Peysakhovich and Jeffrey Naecker. Using methods from machine learning to evaluate behavioral models of choice under risk and ambiguity. *Journal of Economic Behavior and Organization*, 133:373–384, 2017.

[103] Tomaso Poggio, Hrushikesh Mhaskar, Lorenzo Rosasco, Brando Miranda, and Qianli Liao. Why and when can deep-but not shallow-networks avoid the curse of dimensionality: a review. *International Journal of Automation and Computing*, 14(5):503–519, 2017.

[104] Nicholas G Polson and Vadim O Sokolov. Deep learning for short-term traffic flow prediction. *Transportation Research Part C: Emerging Technologies*, 79:1–17, 2017.

[105] John W Pratt. Risk aversion in the small and in the large. *Econometrica: Journal of the Econometric Society*, pages 122–136, 1964.

[106] Sarada Pulugurta, Ashutosh Arun, and Madhu Errampalli. Use of artificial intelligence for mode choice analysis and comparison with traditional multinomial logit model. *Procedia-Social and Behavioral Sciences*, 104:583–592, 2013.

[107] PV Subba Rao, PK Sikdar, KV Krishna Rao, and SL Dhingra. Another insight into artificial neural networks through behavioural analysis of access mode choice. *Computers, environment and urban systems*, 22(5):485–496, 1998.

[108] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM, 2016.

[109] David Rolnick and Max Tegmark. The power of deeper networks for expressing natural functions. *arXiv preprint arXiv:1705.05502*, 2017.

[110] Andrew Slavin Ross and Finale Doshi-Velez. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

[111] Andrew Slavin Ross, Michael C Hughes, and Finale Doshi-Velez. Right for the right reasons: Training differentiable models by constraining their explanations. *arXiv preprint arXiv:1703.03717*, 2017.

[112] Paul A Samuelson. A note on measurement of utility. *The Review of Economic Studies*, 4(2):155–161, 1937.

[113] Ch Ravi Sekhar and E Madhu. Mode choice analysis using random forrest decision trees. *Transportation Research Procedia*, 17:644–652, 2016.

[114] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 618–626, 2017.

[115] Toru Seo, Takahiko Kusakabe, Hiroto Gotoh, and Yasuo Asakura. Interactive online machine learning approach for activity-travel survey. *Transportation Research Part B: Methodological*, 2017.

[116] PC Sham and D Curtis. An extended transmission/disequilibrium test (tdt) for multi-allele marker loci. *Annals of human genetics*, 59(3):323–336, 1995.

[117] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[118] Kenneth Small and Clifford Winston. The demand for transportation: models and applications. In *Essays in Transportation Economics and Policy*. 1998.

[119] Kenneth A Small, Erik T Verhoef, and Robin Lindsey. Travel demand. In *The economics of urban transportation*, volume 2. Routledge, 2007.

[120] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

[121] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012.

[122] Jasper Snoek, Oren Rippel, Kevin Swersky, Ryan Kiros, Nadathur Satish, Narayanan Sundaram, Mostofa Patwary, Mr Prabhat, and Ryan Adams. Scalable bayesian optimization using deep neural networks. In *International Conference on Machine Learning*, pages 2171–2180, 2015.

[123] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3319–3328. JMLR. org, 2017.

[124] Justin Sydnor. (over) insuring modest risks. *American Economic Journal: Applied Economics*, 2(4):177–199, 2010.

[125] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. Cvpr, 2015.

[126] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2014.

[127] Tomomi Tanaka, Colin F Camerer, and Quang Nguyen. Risk and time preferences: linking experimental and household survey data from vietnam. *American Economic Review*, 100(1):557–71, 2010.

[128] Liang Tang, Chenfeng Xiong, and Lei Zhang. Decision tree method for modeling travel mode switching in a dynamic behavioral process. *Transportation Planning and Technology*, 38(8):833–850, 2015.

[129] Kenneth Train. A structured logit model of auto ownership and mode choice. *The Review of Economic Studies*, 47(2):357–370, 1980.

[130] Kenneth E Train. *Discrete choice methods with simulation*. Cambridge university press, 2009.

[131] Amos Tversky and Daniel Kahneman. Advances in prospect theory: Cumulative representation of uncertainty. *Journal of Risk and uncertainty*, 5(4):297–323, 1992.

[132] Sander van Cranenburgh and Ahmad Alwosheel. An artificial neural network based approach to investigate travellers' decision rules. *Transportation Research Part C: Emerging Technologies*, 98:152–166, 2019.

[133] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science and business media, 2013.

[134] Vladimir Naumovich Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999, 1999.

[135] Hal R Varian. Big data: New tricks for econometrics. *The Journal of Economic Perspectives*, 28(2):3–27, 2014.

[136] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge University Press, 2018.

[137] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008.

[138] Ulrike Von Luxburg and Bernhard Schölkopf. Statistical learning theory: Models, concepts, and results. In *Handbook of the History of Logic*, volume 10, pages 651–706. Elsevier, 2011.

[139] Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

[140] Shenhao Wang and Jinhua Zhao. Using deep neural network to analyze travel mode choice with interpretable economic information: An empirical example. *arXiv preprint arXiv:1812.04528*, 2018.

[141] Ray Weaver and Shane Frederick. Transaction disutility and the endowment effect. *NA-Advances in Consumer Research Volume 36*, 2009.

[142] Yuankai Wu, Huachun Tan, Lingqiao Qin, Bin Ran, and Zhuxi Jiang. A hybrid deep learning based traffic flow prediction method and its understanding. *Transportation Research Part C: Emerging Technologies*, 90:166–180, 2018.

[143] Guangnian Xiao, Zhicai Juan, and Chunqin Zhang. Detecting trip purposes from smartphone-based travel surveys with artificial neural networks and particle swarm optimization. *Transportation Research Part C: Emerging Technologies*, 71:447–463, 2016.

[144] Chi Xie, Jinyang Lu, and Emily Parkany. Work travel mode choice modeling with data mining: decision trees and neural networks. *Transportation Research Record: Journal of the Transportation Research Board*, (1854):50–61, 2003.

[145] Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *Transactions of the Association for Computational Linguistics*, 4:259–272, 2016.

[146] Luca Zamparini and Aura Reggiani. The value of travel time in passenger and freight transport: an overview. In *Policy analysis of transport networks*, pages 161–178. Routledge, 2016.

[147] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[148] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

[149] Zhenhua Zhang, Qing He, Jing Gao, and Ming Ni. A deep learning approach for detecting traffic accidents from social media data. *Transportation research part C: emerging technologies*, 86:580–596, 2018.

[150] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. *arXiv preprint arXiv:1412.6856*, 2014.

[151] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pages 2921–2929. IEEE, 2016.

[152] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.

[153] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. *arXiv preprint arXiv:1707.07012*, 2(6), 2017.

[154] Jacek M Zurada, Aleksander Malinowski, and Ian Cloete. Sensitivity analysis for minimization of input data dimension for feedforward neural network. In *Proceedings of IEEE International Symposium on Circuits and Systems-ISCAS'94*, volume 6, pages 447–450. IEEE, 1994.