Deep Neural Networks for Choice Analysis: Architecture Design with Alternative-Specific Utility Functions

Shenhao Wang
Baichuan Mo
Jinhua Zhao

Massachusetts Institute of Technology

Abstract

Whereas deep neural network (DNN) is increasingly applied to choice analysis, it is challenging to reconcile domain-specific behavioral knowledge with generic-purpose DNN, to improve DNN's interpretability and predictive power, and to identify effective regularization methods for specific tasks. To address these challenges, this study demonstrates the use of behavioral knowledge for designing a particular DNN architecture with alternative-specific utility functions (ASU-DNN) and thereby improving both the predictive power and interpretability. Unlike a fully connected DNN (F-DNN), which computes the utility value of an alternative k by using the attributes of *all* the alternatives, ASU-DNN computes it by using only k's *own* attributes. Theoretically, ASU-DNN can substantially reduce the estimation error of F-DNN because of its lighter architecture and sparser connectivity, although the constraint of alternative-specific utility can cause ASU-DNN to exhibit a larger approximation error. Empirically, ASU-DNN has 2-3% higher prediction accuracy than F-DNN over the whole hyperparameter space in a private dataset collected in Singapore and a public dataset available in the R mlogit package. The alternative-specific connectivity is associated with the independence of irrelevant alternative (IIA) constraint, which as a domain-knowledge-based regularization method is more effective than the most popular generic-purpose explicit and implicit regularization methods and architectural hyperparameters. ASU-DNN provides a more regular substitution pattern of travel mode choices than F-DNN does, rendering ASU-DNN more interpretable. The comparison between ASU-DNN and F-DNN also aids in testing behavioral knowledge. Our results reveal that individuals are more likely to compute utility by using an alternative's own attributes, supporting the long-standing practice in choice modeling. Overall, this study demonstrates that behavioral knowledge can guide the architecture design of DNN, function as an effective domain-knowledge-based regularization method, and improve both the interpretability and predictive power of DNN in choice analysis. Future studies can explore the generalizability of ASU-DNN and other possibilities of using utility theory to design DNN architectures.

Keywords: Deep Neural Network; Alternative-Specic Utility; Choice Analysis

1. Introduction

Choice analysis is an important research area across economics, transportation, and marketing [37, 5, 22]. 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 [30, 43, 59]. While DNN has demonstrated 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 [33, 34]. Whereas the ML community generally admires the effectiveness of automatic feature learning in DNN [33], heated debates continue with regard to the extent and manner in which domain knowledge can be used to improve ML models and solve domain-specific problems more efficiently [34]. Second, because DNN is a significantly more complicated generic-purpose model, its interpretability is generally considered to be low [35, 31]. Even though it is relatively straightforward to apply DNN to forecast demand, researchers have obtained limited policy and behavioral insights from DNN until now. Third, even the prediction itself can be challenging because of the high dimensionality and data overfitting of DNN. Effective regularization methods and DNN architectures are important to improve the out-of-sample performance. Whereas many recent progresses were achieved by creating novel DNN architectures, the procedure of designing deep architecture is still largely ad hoc without systematic guidance [63, 38]. 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 critical for applying DNN to any specific domains.

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 ASU-DNN architecture, in which the utility of an alternative depends only on 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. Using statistical learning theory, we demonstrate that this ASU-DNN architecture can reduce the estimation error of F-DNN thanks to its much sparser connectivity and fewer parameters, although the approximation error of ASU-DNN could be higher. We then apply ASU-DNN, F-DNN, multinomial logit (MNL), nested logit (NL), and 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 was from the mlogit package in R. Our results demonstrate that ASU-DNN exhibits consistently higher prediction accuracy than F-DNN and the other eleven classifiers in predicting travel mode choice over the whole hyperparameter space. The alternative-specific connectivity design in ASU-DNN leads to an IIA-constraint substitution pattern across the alternatives, which can be considered as a domain-knowledge-based regularization, in contrast to the generic-purpose regularization methods such as explicit and implicit regularizations and other architectural hyperparameters. Our results show that the domain-knowledge-based regularization is more effective than the generic-purpose regularization in improving the prediction performance. Finally, we interpret the substitution pattern across travel mode alternatives in ASU-DNN by using sensitivity analysis and demonstrate that ASU-DNN reveals more reasonable behavioral patterns than F-DNN owing to its more regular and intuitive choice probability functions. Overall, the behavioral knowledge of alternative-specific utility function can be used to partially address all three challenges of DNN applications by integrating 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.

Broadly speaking, this study points to a new research direction of injecting behavioral knowledge into DNN and enhancing DNN architectures specifically for choice analysis. We aim to advance domain-specific behavioral knowledge using DNN, as opposed to simply applying DNNs for prediction adopted by most recent studies in the transportation domain. This research direction is feasible because the behavioral knowledge used in the classic choice models has a counterpart in the DNN architecture. Specifically, the substitution pattern between alternatives can be controlled by the connectivity of the DNN architecture, and vice versa. From an ML perspective, behavioral knowledge can function as domain-knowledge-based regularization, which better fits domain-specific tasks than generic-purpose regularizations. The alternative-specific utility is only one small piece in the rich set of behavioral insights accumulated over decades of transportation scholarship, and future studies can explore and create more noteworthy DNN architectures for choice analysis based on this behavioral perspective.

The paper is organized as follows: The next section reviews studies on DNN's applications, interpretability, and regularization methods. Section 3 examines three theoretical aspects of DNN: 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 presents the experiments, and discusses the prediction accuracy, effectiveness of domain-knowledge-based regularization, and interpretability of ASU-DNN. Section 5 concludes.

2. Literature Review

Individual decision-making has been an important topic in many domains, including marketing [22], economics [37], transportation [5, 55], biology [51], and public policy [9]. In recent years as ML models permeated into these domains, researchers started to use various classifiers to analyze how individuals take decisions [43, 30]. In the transportation domain, Karlaftis and Vlahogianni (2011) [30] 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 [46, 42, 50, 43, 11, 45, 36, 64, 12]. 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 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 [33, 6]. 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 [32, 54]. On the other hand, a few studies indicate that handcrafted features could still aid in constructing DNN models [34]. 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 [21].

Another challenge to DNN application is DNN's lack of interpretability, which is caused by its complex model assumptions [35, 15]. The interpretability of DNN is particularly important for reasons such as safety, transparency, trust, and construction of new knowledge [17, 10]. 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 [41, 47, 62, 42, 23]. Prediction-driven ML models differ significantly from the classical choice models, which are both predictive and interpretable [37]. 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 [48] (which improves interpretability before model building) and post-hoc interpretation (which focuses on extracting information after model training) [15]. For example, CNN can be interpreted in a post-hoc manner by visualizing the semantic contents in image recognition tasks [65]. In choice analysis, it appears feasible to post-hoc interpret DNN and derive the economic information from DNNs [59, 47, 7]. Some other studies used the computational graphs to represent the travel demand structures [61, 53]. However, these studies that use the visualization of computational graphs did not examine the connection between the utility theory that the choice modeling relies on and the compositional structure of the hidden layers that is the hallmark of DNNs [44], failing to take advantage of either the function approximation capacity of DNNs or the rigorous behavioral insights captured in utility theories.

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 [63]. New DNN architectures could also aid in improving DNN performance. Recent studies either manually design new architectures (such as AlexNet [32], GoogLeNet [54], and ResNet [24]) or automatically search for novel architectural design by using Gaussian process, reinforcement learning, or other sequential modeling techniques [52, 29, 66, 16]. However, most architecture designs are ad hoc explorations without systematic guidance, and the final DNN architecture identified through automatic searching is not interpretable.

3. Theory

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 $\sum\limits_k y_{ik} = 1$. RUM assumes that the utility of each alternative is the sum of the deterministic utility V_{ik} and random utility ϵ_{ik}:

$$U_{ik} = V_{ik}(z_i, \tilde{x}_i) + \epsilon_{ik} \tag{1}$$

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

$$P_{ik} = Prob(V_{ik} + \epsilon_{ik} > V_{ij} + \epsilon_{ij}, \forall j \in B,\ j \neq k) \tag{2}$$

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

$$P_{ik} = \int \prod_{j \neq k} F_{\epsilon_{ij}}(V_{ik} - V_{ij} + \epsilon_{ik}) dF(\epsilon_{ik}) \tag{3}$$

The following two propositions demonstrate how DNN and RUM are related. The proof of the two propositions is available in Appendix I.

Proposition 1. *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 $P_{ik} = \frac{e^{V_{ik}}}{\sum\limits_j e^{V_{ij}}}$.*

The proof is available in many choice modeling textbooks [55, 5].

Proposition 2. *Suppose that Equation 3 holds and that choice probability P_{ik} takes the form of Softmax function as in Equation 17. 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) [37].

Propositions 1 and 2 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.

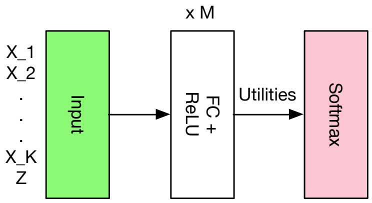


Fig. 1. Fully Connected Feedforward DNN (F-DNN); it is a standard feedforward DNN. The inputs incorporate both alternative-specific and individual-specific variables. The inputs into the Softmax activation function can be interpreted as utilities.

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

$$V_{ik} = V(z_i, \tilde{x}_i) = w_k^T \Phi(z_i, \tilde{x}_i) = w_k^T (g_m ... \circ g_2 \circ g_1)(z_i, \tilde{x}_i) \tag{4}$$

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 4 illustrates that V_{ik} becomes alternative-specific only in the final layer prior to the Softmax function when w_k is applied to $\Phi(z_i, \tilde{x}_i)$.

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 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

$$V_{ik} = V(z_i, x_{ik}) = w_k^T \Phi(z_i, x_{ik}) = w_k^T (g_{M_2} ... \circ g_2 \circ g_1)((g_{M_1}^{x_k} ... \circ g_1^{x_k})(x_{ik}), (g_{M_1}^{z} ... \circ g_1^{z})(z_i)) \qquad (5)$$

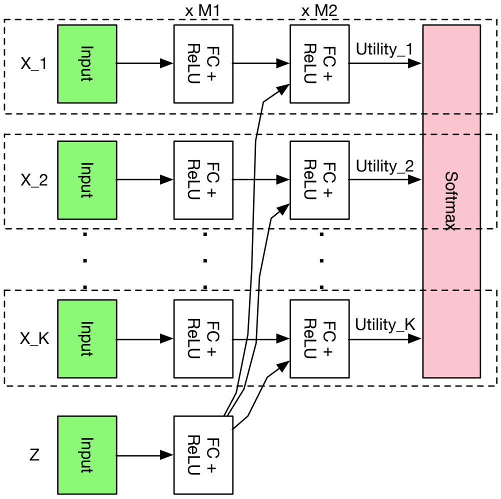


Fig. 2. ASU-DNN; Deep neural network architecture based on utility theory. It could be considered as a stack of fully connected subnetworks, with each computing a utility score for each alternative. Individual-specific variables interact with alternative-specific variables after M_1 layers.

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, 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 [13, 28, 27]. Secondly, this alternative-specific connectivity design could provide more regular information than F-DNN owing to the underlying utility theory. The two architectures in Figures 1 and 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 [60, 14]. 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. However, to formally evaluate the effectiveness of this regularization, the statistical learning theory is required to discuss the tradeoff between the approximation and estimation errors, as shown in the next section.

3.3. Estimation and Approximation Error Tradeoff Between ASU-DNN and F-DNN

It is not true that ASU-DNN can always outperform F-DNN. This is because any constraint applied to DNN could 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:

$$\mathbb{E}_S[L(\hat{f}) - L(f^*)] = \mathbb{E}_S[L(\hat{f}) - L(f^*_F)] + \mathbb{E}_S[L(f^*_F) - L(f^*)], \quad \mathcal{F} \in \{\mathcal{F}_1, \mathcal{F}_2\}; \hat{f} \in \{\hat{f}_1, \hat{f}_2\} \tag{6}$$

where $L = \mathbb{E}_{x,y}[l(y, f(x)]$ is the expected loss function and S represents the sample $\{x_i, y_i\}_1^N$. $f^*_F = \underset{f \in \mathcal{F}}{\text{argmin}}\ L(f)$, 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*

$$\mathbb{E}_S[L(\hat{f}) - L(f^*_F)] \tag{7}$$

And an *approximation error*

$$\mathbb{E}_S[L(f^*_F) - L(f^*)] \tag{8}$$

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:

$$\mathbb{E}_S[L(f^*_{\mathcal{F}_1}) - L(f^*)] \geq \mathbb{E}_S[L(f^*_{\mathcal{F}_2}) - L(f^*)], \quad \mathcal{F}_1 \subset \mathcal{F}_2 \tag{9}$$

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

$$\hat{\mathcal{R}}_n(\mathcal{F}|_S) = \mathbb{E}_\epsilon \sup_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^N \epsilon_i f(x_i) \tag{10}$$

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

Proposition 3. *The estimation error of an estimator $\hat{f}$ can be bounded by the Rademacher complexity of $\mathcal{F}$.*

$$\mathbb{E}_S[L(\hat{f}) - L(f^*_F)] \leq 2\mathbb{E}_S\hat{\mathcal{R}}_n(\mathcal{F}|_S) \tag{11}$$

Definition 1 provides a measurement for the complexity of the function class $\mathcal{F}$. Proposition 3 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 1 and Proposition 3 are available in recent studies about the statistical learning theory [57, 58, 3].

Proposition 4. *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*

$$\hat{\mathcal{R}}_n(\mathcal{F}|_S) \leq \frac{(\sqrt{2\log(D)}+1)\sqrt{\frac{1}{N}\sum_{i=1}^{N}||x_i||^2}}{\sqrt{N}} \times \prod_{j=1}^{D} M_F(j) \tag{12}$$

Remarks on Proposition 4:

1. As this result is from Golowich et al. (2017) [20], so its proof is omitted in this study. Other relevant proofs are available in [3, 40, 1].
2. Proposition 4 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 [56, 4], 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 4 can be re-expressed as:

$$\hat{\mathcal{R}}_n(\mathcal{F}_2|_S) \leq \frac{(\sqrt{2\log(D)}+1)\sqrt{\frac{1}{N}\sum_{i=1}^{N}||x_i||^2}}{\sqrt{N}} \times c^D T^D \tag{13}$$

Proposition 5. *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*

$$\hat{\mathcal{R}}_n(\mathcal{F}_1|_S) \leq \frac{(\sqrt{2\log(D)}+1)\sqrt{\frac{1}{N}\sum_{i=1}^{N}||x_i||^2}}{\sqrt{N}} \times \frac{c^D T^D}{K^{D/2}} \tag{14}$$

Remarks on Proposition 5:

[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.

1. Proposition 5 can be derived from Proposition 4 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 6-14 constitute the formal method for illustrating the tradeoff between ASU-DNN and F-DNN. Owing to its sparse connectivity, ASU-DNN has smaller estimation error as its main advantage, particularly when K is large, as shown in Equation 14. 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 indicates 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 13 and 14 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.

4. Setup of Experiments

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. Detailed summary statistics of TRAIN and SGP are attached in Appendix II.

4.2. Hyperparameter Space

A challenge in the comparison between the two DNN architectures is the large number of hyperparameters, on which the performance of DNN largely depends. Table 1 summarizes a list of hyperparameters and the range of their values. 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.

Table 1: Hyperparameter space of F-DNN and ASU-DNN; *Panel 1.* Hyperparameters that don't change in the hyperparameter searching; *Panel 2.* Hyperparameters that change in only F-DNN; *Panel 3.* Hyperparameters that change in only ASU-DNN; M_1 and n_1 are the depth and width before the interaction between x_{ik} and z_i. *Panel 4.* Hyperparameters that change in both F-DNN and ASU-DNN.

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)	$[1.0,0.5,0.1,0.01,10^{-3},10^{-5},10^{-10},10^{-20}]$
γ_2 (l_2 penalty)	$[1.0,0.5,0.1,0.01,10^{-3},10^{-5},10^{-10},10^{-20}]$
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	$[50,100,200,500,1000]$

A brief introduction for some hyperparameters is as following. **Activation Functions.** Rectified linear unit (ReLU) is used in the middle layers and Softmax is used in the last layer. Other activation functions are also possible, although recent studies have shown that non-saturated activation functions (e.g. ReLU) perform better than the saturated activation functions (e.g. Tanh) [32]. **Initialization.** It refers to the process of initializing the parameters in DNN. DNN initialization does not have formal theory yet, although Glorot and He initializations are commonly used in practice [18, 19, 25]. **Depth and Width.** They refer to the number of layers and the number of neurons in each layer of DNN. Depth and width control the model complexity: DNN models have smaller approximation errors and larger estimation errors, when they become wider and deeper. **Penalties.** Both l_1 and l_2 penalties are explicit regularization added to the standard cross-entropy loss function. The l_1 penalty encourages model sparsity; the l_2 penalty shrinks the magnitude of coefficients. **Dropout.** It refers to the process of randomly dropping certain proportion of the neurons in training [26], and since this procedure leads to sparser architecture, it can also be treated as a regularization method. **Batch Normalization.** It is the normalization of each batch in the stochastic gradient descent (SGD). **Number of Iterations.** It refers to the number of iterations in the training. Too few training iterations could lead to an underfitted model and too many iterations could lead to an overfitted model. As a result, a relatively small number of iterations (e.g. early stopping) can be considered as a regularization method.

4.3. Hyperparameter Searching

It is a benchmark method to randomly search in the hyperparameter space to identify the DNN configuration with a high prediction accuracy [8]. In our study, 100 DNN models were trained, 50 each for the two DNN architectures. Formally, the empirical risk minimization (ERM) is

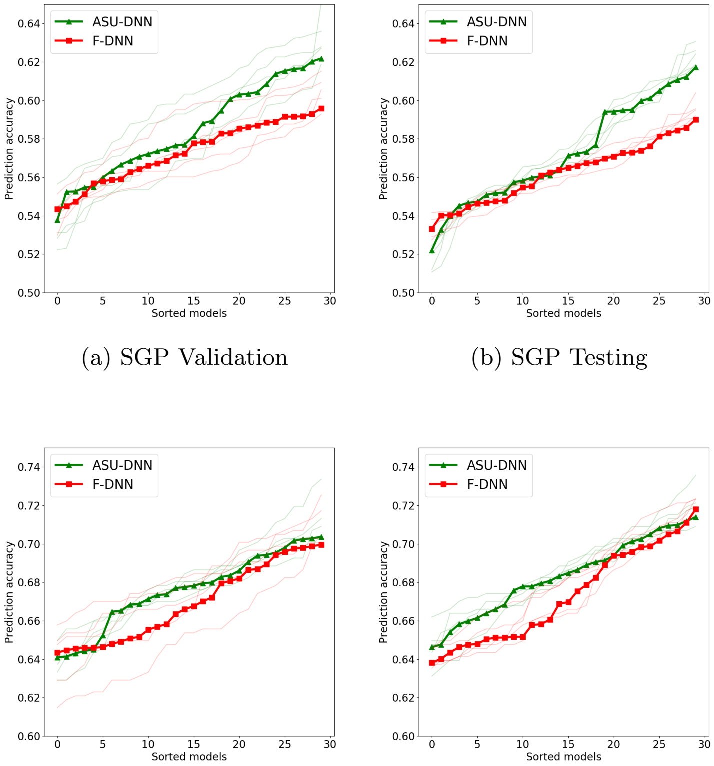
$$\min_w E(w, w_h) = \min_w \frac{1}{N}\sum_i^N l(y_i, P_{ik}; w, w_h) + \gamma ||w||_p \tag{15}$$

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^*

$$w_h^* = \underset{w_h \in \{w_h^{(1)}, w_h^{(2)}, \dots, w_h^{(S)}\}}{\operatorname{argmin}} E(w^*, w_h) \tag{16}$$

5. Experiment Results

The result section consists of three parts. The first part compares the prediction accuracy of ASU-DNN, F-DNN, MNL, NL, and 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, F-DNN, MNL, and NL in terms of their interpretability by visualizing their choice probability functions and computing their elasticity



(a) SGP Validation

(b) SGP Testing

(c) TRAIN Validation

(d) TRAIN Testing

Fig. 3. Hyperparameter Searching Results; in all four subfigures, models are sorted according to prediction accuracy. Green curves represent ASU-DNN performance, and red ones represent F-DNN. Dark curves are the average of five-fold cross-validation, and light ones are the individual trainings. Overall, ASU-DNN consistently outperforms F-DNN. The information of top DNN architectures is attached in Appendix III.

coefficients. The first part uses both SGP and TRAIN datasets, and the second and third parts focus on only the SGP dataset for simplicity.

5.1. Prediction Accuracy

Figure 3 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 [2]. 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

[2] Here we focus on only the top models since researchers only choose the top ML models for analysis. For example, researchers compare the top 1 model or the top 5 models in two different model families, so we don't discuss the mean or the variance of the models' performance.

sets. The best ASU-DNN outperforms the best F-DNN by approximately 3%. In the TRAIN dataset, whereas the ASU-DNN still consistently 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.

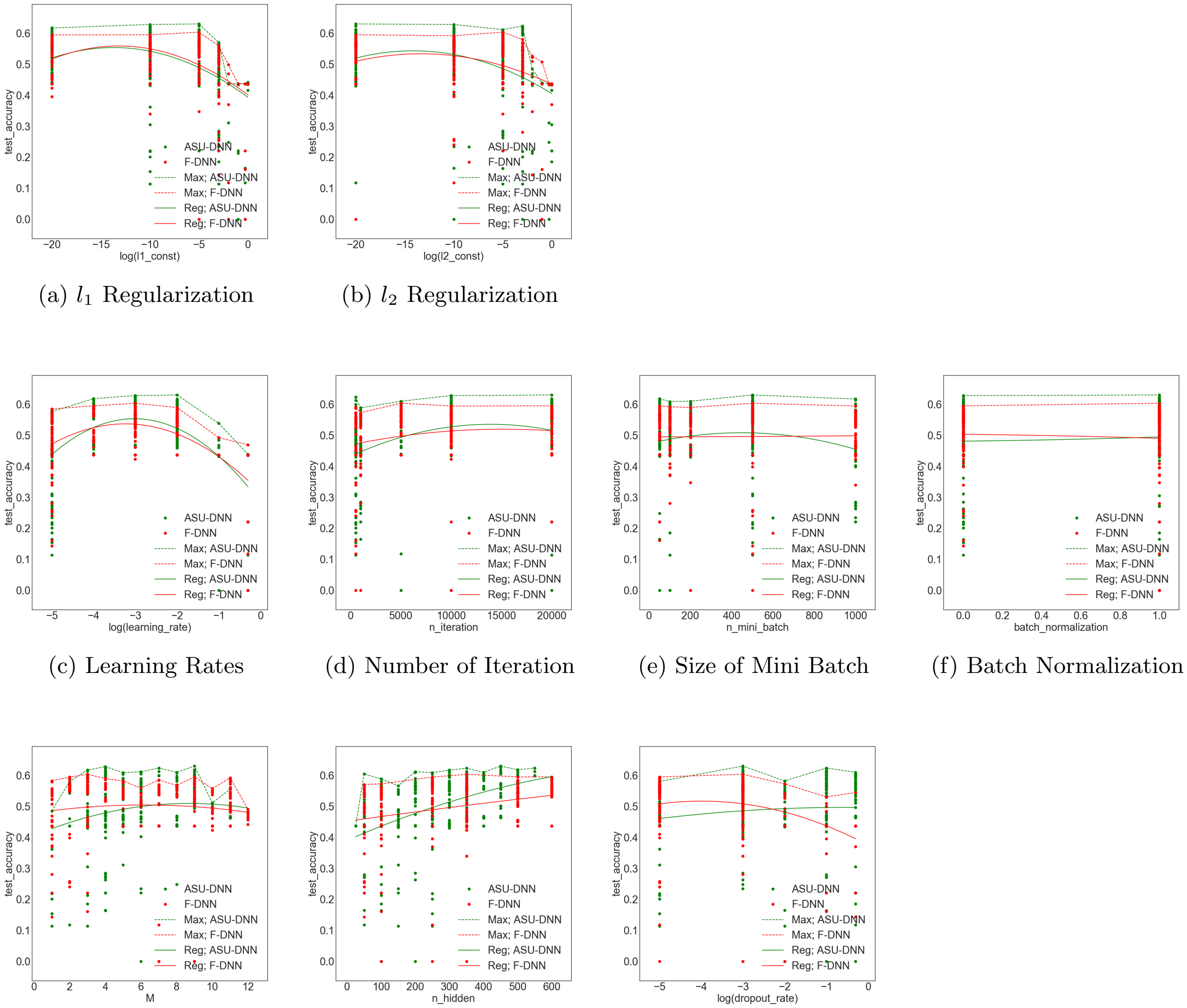
Table 2 also illustrates that both F-DNN and ASU-DNN perform better than the other eleven classifiers, implying that DNN models fit choice analysis tasks very effectively. Specifically, F-DNN and ASU-DNN outperform the baseline MNL and NL by about 8% prediction accuracy, implying that the compositional function structure of DNN is effective. 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.

Table 2: Prediction accuracy of all classifiers; MNL represents the multinomial logit model and NL represents the nested logit model. Nest 1: walking + bus (the corresponding scale parameter μ_1 is fixed to 1); Nest 2: AV + ridesharing + driving. NL is not applicable (N.A.) to the TRAIN data set because it has only two alternatives. LR (l1_reg/l2_reg) represents a logistic regression model with mild l1 or l2 regularization; SVM (Linear/RBF) represents for support vector machine with linear or RBF kernels; KNN_3 represents three-nearest neighbor classifier; decision tree is abbreviated as DT; quadratic discriminant analysis is as QDA. The DNN models outperform all the other classifiers.

	ASU-DNN (Top 1)	F-DNN (Top 1)	ASU-DNN (Top 10)	F-DNN (Top 10)	MNL	NL	LR (l1_reg)	LR (l2_reg)	SVM (Linear)	SVM (RBF)	Naive Bayesian	KNN_3	DT	AdaBoost	QDA
Validation (SGP)	62.3%	59.2%	61.3%	58.8%	53.0%	54.1%	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%	51.2%	52.1%	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%	69.4%	N.A.	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%	67.9%	N.A.	67.8%	67.9%	68.3%	58.7%	56.4%	57.7%	65.0%	69.8%	60.5%

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 4 shows the results, with each of the subfigures depicting the comparison of a hyperparameter with the alternative-specific connectivity hyperparameter.



(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

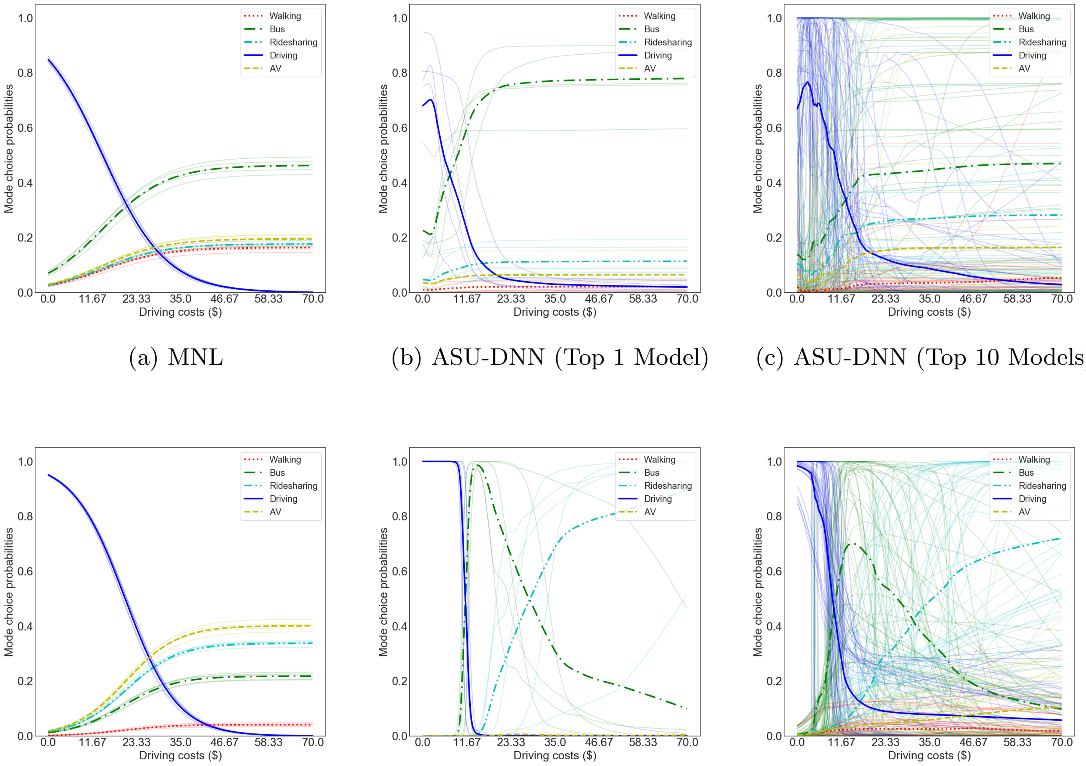
(i) Dropout Rates

Fig. 4. Comparing alternative-specific connectivity to explicit regularizations, implicit regularizations, and architectural hyperparameters in the SGP testing dataset; *First row*: Explicit regularizations; *Second row*: Implicit regularizations; *Third row*: Architectural hyperparameters. In all the subfigures, the x-axis represents the hyperparameter and the y-axis represents the prediction accuracy. The dashed lines connect the models with the highest prediction accuracy for each single value of the hyperparameter on the x-axis. The solid curves are the quadratic regression curves of prediction accuracy on the hyperparameter on the x-axis. The maximum prediction accuracy (dashed curves) is more important than the average accuracy (solid curves) because we target only top models rather than average models. The results for the validation set are available in Appendix IV. Overall, ASU-DNN could outperform F-DNN regardless of the values of the other hyperparameters.

Explicit regularizations. Figures 4a and 4b 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 regularizations. 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 4a and 4b. 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 4c, 4d, 4e, and 4f 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 15, 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 4c, 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 4d, 4e, and 4f, 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 4g, 4h, and 4i 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 4g, 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. In Figure 4h, the maximum prediction accuracy of F-DNN form almost horizontal lines everywhere. Finally, in Figure 4i, 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



(a) MNL

(b) ASU-DNN (Top 1 Model)

(c) ASU-DNN (Top 10 Models)

(d) NL

(e) F-DNN (Top 1 Model)

(f) F-DNN (Top 10 Models)

Fig. 5. Choice probability functions of MNL, NL, ASU-DNN, and F-DNN in the SGP testing set. *Upper row:* MNL and ASU-DNN models. *Lower row:* NL and F-DNN models. Each light curve represents a training result; the dark curves represent the average of all the training results. ASU-DNN compromises MNL and F-DNN, since it retains the global IIA-constraint substitution pattern of MNL and the local richness of F-DNN.

gain.

5.3. Interpretation of ASU-DNN: Combining IIA and DNN

Whereas DNN is generally criticized as lacking interpretability, we can visualize the choice probability functions and compute the elasticity coefficients in DNN models by using numerical simulation [7, 39, 2, 49, 59]. Figure 5 shows how, following this method, the probabilities of selecting five travel modes vary with increasing driving costs in the ASU-DNN, F-DNN, MNL, and NL models, while holding all other variables constant at their empirical mean values.

The choice probability functions of ASU-DNN mix the behavioral patterns of MNL and F-DNN, since ASU-DNN retains the global IIA-constraint substitution pattern from MNL and the local richness from F-DNN. Comparing ASU-DNN and F-DNN, the choice probability functions of ASU-DNN appear more intuitive than those of F-DNN for at least two reasons. The first 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, consistent with the common notion that the alternatives are often substitute goods. 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, which is similar to the MNL model in Figure 5a. The second 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. Overall, ASU-DNN presents more regularity than F-DNN, which is caused by the built-in alternative-specific connectivity design.

Tables 3-6 summarize the elasticity coefficients for the MNL, NL, top 10 ASU-DNN, and top 10 F-DNN models, with negative values being bolded to highlight the structure in each table. These elasticity coefficients are computed by simulation, with each input variable varied by 1% holding all the other variables constant at the sample mean values. As shown in Table 3, the MNL model clearly reveals its IIA substitution pattern in two ways. First, all the self-elasticity coefficients are negative as highlighted on the main diagonal, while the cross-elasticity coefficients are all positive. Second, the four cross-elasticity coefficients regarding one specific attribute have the same magnitude. For example, regarding the walking time, the cross-elasticity coefficients of taking buses, ride-sharing, driving, and using AVs are all 0.134, which is consistent with the elasticity formula of MNL models [3]. Table 4 demonstrates how a NL model has a more flexible substitution pattern than MNL. The elasticity coefficients take a clear block-wise shape and the values within a nest are different from those cross nests.

Table 3: Elasticity coefficients of MNL

	Walk	Bus	Ridesharing	Drive	AV
Walk: walk time	**-1.890**	0.134	0.134	0.134	0.134
Bus: cost	0.137	**-0.546**	0.137	0.137	0.137
Bus: in-vehicle time	0.128	**-0.475**	0.128	0.128	0.128
Ridesharing: cost	0.029	0.029	**-0.240**	0.029	0.029
Ridesharing: in-vehicle time	0.083	0.083	**-0.740**	0.083	0.083
Drive: cost	0.288	0.288	0.288	**-0.793**	0.288
Drive: in-vehicle time	0.280	0.280	0.280	**-0.440**	0.280
AV: cost	0.048	0.048	0.048	0.048	**-0.449**
AV: in-vehicle time	0.060	0.060	0.060	0.060	**-0.560**

The elasticity coefficients in Table 5 shows that the substitution pattern of ASU-DNN is very similar to MNL in Table 3. The similarity can be seen from the positive self-elasticity coefficients on the main diagonal, the negative cross-elasticity coefficients on the off-diagonal, and the same

[3] Please refer to Chapter 3 in Train's textbook [55]

Table 4: Elasticity coefficients of NL. Nest 1: walk and bus. Nest 2: ridesharing, drive, and AV

	Walk	Bus	Ridesharing	Drive	AV
Walk: walk time	**-1.481**	**-0.067**	0.163	0.163	0.163
Bus: cost	**-0.074**	**-0.550**	0.170	0.170	0.170
Bus: in-vehicle time	**-0.050**	**-0.356**	0.116	0.116	0.116
Ridesharing: cost	0.039	0.039	**-0.488**	0.080	0.080
Ridesharing: in-vehicle time	0.073	0.073	**-0.988**	0.146	0.146
Drive: cost	0.229	0.229	0.424	**-0.905**	0.424
Drive: in-vehicle time	0.269	0.269	0.482	**-0.593**	0.482
AV: cost	0.048	0.048	0.093	0.093	**-0.687**
AV: in-vehicle time	0.057	0.057	0.109	0.109	**-0.800**

cross-elasticity coefficients regarding one specific attribute. This similarity should not be a surprise, since the ASU-DNN in the family of DNN models corresponds to the MNL in the family of discrete choice models, owing to the alternative-specific utility functions in ASU-DNN. As a comparison, the elasticity coefficients of F-DNN in Table 6 are much more irregular than those of ASU-DNN and even NL: many cross-elasticity coefficients are negative and the elasticity coefficients don't have the block-wise pattern as in NL. Note that this "irregularity" in F-DNN does not necessarily have a negative connotation. It can be the case that the elasticity pattern in F-DNN captures the real data generating process that is out of the model families of MNL, NL, or even ASU-DNN. Therefore, F-DNN might enable researchers to capture the highly correlated utility errors, as in mixed logit (MXL) models. However, it is hard to make a definitive judgment by using only our empirical results. We leave these two questions, whether F-DNN describes the highly correlated utility error terms (as in MXL) and whether the behavioral patterns revealed in ASU-DNN and F-DNN are realistic, open to future studies.

Table 5: Average elasticity coefficients of top 10 ASU-DNN Models

	Walk	Bus	Ridesharing	Drive	AV
Walk: walk time	**-10.016**	1.029	1.028	1.029	1.030
Bus: cost	0.381	**-1.983**	0.395	0.396	0.391
Bus: in-vehicle time	0.440	**-3.198**	0.438	0.435	0.436
Ridesharing: cost	0.219	0.221	**-2.638**	0.221	0.223
Ridesharing: in-vehicle time	0.420	0.421	**-4.878**	0.420	0.420
Drive: cost	1.709	1.735	1.726	**-2.249**	1.731
Drive: in-vehicle time	2.138	2.172	2.178	**-1.952**	2.171
AV: cost	0.383	0.379	0.380	0.380	**-4.681**
AV: in-vehicle time	0.364	0.362	0.363	0.362	**-3.485**

6. Conclusion and Discussion

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

Table 6: Average elasticity coefficients of top 10 F-DNN Models

	Walk	Bus	Ridesharing	Drive	AV
Walk: walk time	**-4.228**	0.580	0.447	0.172	0.109
Bus: cost	**-0.696**	**-2.052**	**-0.093**	0.623	0.342
Bus: in-vehicle time	**-0.053**	**-1.803**	**-0.339**	0.502	0.588
Ridesharing: cost	0.055	0.292	**-1.858**	0.142	1.457
Ridesharing: in-vehicle time	**-0.139**	**-0.115**	**-3.436**	0.434	0.268
Drive: cost	0.897	1.404	2.079	**-1.711**	1.474
Drive: in-vehicle time	1.266	1.690	2.164	**-1.748**	1.937
AV: cost	**-0.516**	0.036	0.356	0.443	**-3.781**
AV: in-vehicle time	**-0.769**	0.457	0.074	0.360	**-3.288**

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 is significantly effective, as demonstrated by our empirical results that ASU-DNN model is more predictive and provides more regular behavioral information than F-DNN. ASU-DNN could outperform F-DNN by approximately $2 - 3\%$ in both validation and testing data sets 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 specification leads to the IIA constraint, which can be considered as a regularization method under the DNN framework. This constraint 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.

One natural question is to what extent our findings are generalizable. This ASU-DNN model is guaranteed to have the IIA-constraint substitution pattern, smaller estimation errors than F-DNN, and more flexibility and higher function approximation power than MNL. These results are always generaliable, owing to the design of ASU-DNN architecture. However, it is neither theoretically nor empirically guaranteed that ASU-DNN always outperforms F-DNN and MNL in terms of prediction accuracy. The prediction performance depends on the sample size, model complexity, and the underlying data generating process (DGP) that is never known to researchers in empirical studies. Loosely speaking, ASU-DNN tends to perform better than F-DNN when sample size becomes smaller, DGP is closer to the IIA-constraint substitution pattern, and the number of alternatives in the choice set becomes larger. ASU-DNN tends to outperform MNL when the utility specification of each alternative is more complicated than simple linear or quadratic forms, although both ASU-DNN and MNL will have misspecification errors when the true DGP deviates from the alternative-specific utility specification. Related to this generalizability discussion, another open question is whether the behavioral pattern revealed in ASU-DNN is realistic. Unfortunately this realism question is hard to answer given that the DGP is never known to researchers in empirical studies. Instead of making a value judgment here, we would encourage future studies to use simulations to answer under what conditions ASU-DNN can approximate the true DGP in a more efficient manner than both F-DNN and MNL.

The alternative-specific utility specification can be incorrect in ASU-DNN. 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.

The ASU-DNN in the family of DNN models is the counterpart of the MNL in the family of discrete choice models. This mapping is enabled by a triangle relationship between the IIA-constraint substitution pattern, choice probability functions taking the Softmax form, and the IID error terms with extreme value distributions. This triangle relationship was neatly established in McFadden's seminal paper [37], which demonstrates any one of the three conditions leads to the other two under the RUM framework. However, the triangle relationship does not explicitly exist for choice models beyond MNL. Whereas researchers can derive the choice probability functions of NL based on the generalized extreme value (GEV) distributions, the proof of the reversed direction is unclear. The mixed logit (MXL) model that allows more flexible correlation between the utility error terms is even more complicated, since the choice probabilities in MXL are computed by sampling, which deviates further away from any analytical approach. Our study has empirically demonstrated that the elasticity coefficients of F-DNN are more flexible than NL as shown in Tables 4 and 6, leading to our conjecture that there exists another regularized DNN model that is corresponding to the NL or GEV models. A valid support for this conjecture is beyond the scope of this study, and we hope future studies can identify the regularization methods that are associated with the NL or even MXL models.

Regardless of certain caveats and remaining questions, 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 NL and MXL) to design novel DNN architectures. Future researchers should also examine the generalizability of ASU-DNN by testing whether it can perform better than F-DNN and choice models in other contexts.

Author Contributions

S.W. conceived of the presented idea, developed the theory, reviewed previous studies, and derived the analytical proofs. S.W. and B.M. designed and conducted the experiments; S.W. drafted the manuscripts; J.Z. provided comments and supervised this work. All authors discussed the results and contributed to the final manuscript.

References

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

[2] David Baehrens et al. “How to explain individual classification decisions”. In: *Journal of Machine Learning Research* 11.Jun (2010), pp. 1803–1831.

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

[4] Peter L Bartlett et al. “Nearly-tight VC-dimension and pseudodimension bounds for piecewise linear neural networks”. In: *arXiv preprint arXiv:1703.02930* (2017).

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

[6] Yoshua Bengio, Aaron Courville, and Pascal Vincent. “Representation learning: A review and new perspectives”. In: *IEEE transactions on pattern analysis and machine intelligence* 35.8 (2013), pp. 1798–1828.

[7] Yves Bentz and Dwight Merunka. “Neural networks and the multinomial logit for brand choice modelling: a hybrid approach”. In: *Journal of Forecasting* 19.3 (2000), pp. 177–200.

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

[9] Axel Börsch-Supan and John Pitkin. “On discrete choice models of housing demand”. In: *Journal of Urban Economics* 24.2 (1988), pp. 153–172.

[10] Robert Brauneis and Ellen P Goodman. “Algorithmic transparency for the smart city”. In: (2017).

[11] Giulio Erberto Cantarella and Stefano de Luca. “Multilayer feedforward networks for transportation mode choice analysis: An analysis and a comparison with random utility models”. In: *Transportation Research Part C: Emerging Technologies* 13.2 (2005), pp. 121–155.

[12] Sander van Cranenburgh and Ahmad Alwosheel. “An artificial neural network based approach to investigate travellers’ decision rules”. In: *Transportation Research Part C: Emerging Technologies* 98 (2019), pp. 152–166.

[13] George Cybenko. “Approximation by superpositions of a sigmoidal function”. In: *Mathematics of control, signals and systems* 2.4 (1989), pp. 303–314.

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

[15] Finale Doshi-Velez and Been Kim. “Towards a rigorous science of interpretable machine learning”. In: (2017).

[16] Stefan Falkner, Aaron Klein, and Frank Hutter. “BOHB: Robust and efficient hyperparameter optimization at scale”. In: *arXiv preprint arXiv:1807.01774* (2018).

[17] Alex A Freitas. “Comprehensible classification models: a position paper”. In: *ACM SIGKDD explorations newsletter* 15.1 (2014), pp. 1–10.

[18] 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*. 2010, pp. 249–256.

[19] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. “Domain adaptation for large-scale sentiment classification: A deep learning approach”. In: *Proceedings of the 28th international conference on machine learning (ICML-11)*. 2011, pp. 513–520.

[20] Noah Golowich, Alexander Rakhlin, and Ohad Shamir. “Size-independent sample complexity of neural networks”. In: *arXiv preprint arXiv:1712.06541* (2017).

[21] Ian Goodfellow et al. *Deep learning*. Vol. 1. MIT press Cambridge, 2016.

[22] Peter M Guadagni and John DC Little. “A logit model of brand choice calibrated on scanner data”. In: *Marketing science* 2.3 (1983), pp. 203–238.

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

[24] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[25] Kaiming He et al. “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification”. In: *Proceedings of the IEEE international conference on computer vision.* 2015, pp. 1026–1034.

[26] Geoffrey E Hinton et al. “Improving neural networks by preventing co-adaptation of feature detectors”. In: *arXiv preprint arXiv:1207.0580* (2012).

[27] Kurt Hornik. “Approximation capabilities of multilayer feedforward networks”. In: *Neural networks* 4.2 (1991), pp. 251–257.

[28] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. “Multilayer feedforward networks are universal approximators”. In: *Neural networks* 2.5 (1989), pp. 359–366.

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

[30] Matthew G Karlaftis and Eleni I Vlahogianni. “Statistical methods versus neural networks in transportation research: Differences, similarities and some insights”. In: *Transportation Research Part C: Emerging Technologies* 19.3 (2011), pp. 387–399.

[31] Pang Wei Koh and Percy Liang. “Understanding black-box predictions via influence functions”. In: *arXiv preprint arXiv:1703.04730* (2017).

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

[33] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. “Deep learning”. In: *Nature* 521.7553 (2015), pp. 436–444.

[34] Qianli Liao and Tomaso Poggio. *When Is Handcrafting Not a Curse?* Tech. rep. 2018.

[35] Zachary C Lipton. “The mythos of model interpretability”. In: *arXiv preprint arXiv:1606.03490* (2016).

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

[37] Daniel McFadden. “Conditional logit analysis of qualitative choice behavior”. In: (1974).

[38] Hrushikesh Mhaskar, Qianli Liao, and Tomaso Poggio. “Learning functions: when is deep better than shallow”. In: *arXiv preprint arXiv:1603.00988* (2016).

[39] Gregoire Montavon, Wojciech Samek, and Klaus-Robert Muller. “Methods for interpreting and understanding deep neural networks”. In: *Digital Signal Processing* 73 (2018), pp. 1–15.

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

[41] Peter Nijkamp, Aura Reggiani, and Tommaso Tritapepe. "Modelling inter-urban transport flows in Italy: A comparison between neural network analysis and logit analysis". In: *Transportation Research Part C: Emerging Technologies* 4.6 (1996), pp. 323–338.

[42] Hichem Omrani. "Predicting travel mode of individuals by machine learning". In: *Transportation Research Procedia* 10 (2015), pp. 840–849.

[43] Miguel Paredes et al. "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*. IEEE, 2017, pp. 780–785.

[44] Tomaso Poggio et al. "Why and when can deep-but not shallow-networks avoid the curse of dimensionality: a review". In: *International Journal of Automation and Computing* 14.5 (2017), pp. 503–519.

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

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

[47] PV Subba Rao et al. "Another insight into artificial neural networks through behavioural analysis of access mode choice". In: *Computers, environment and urban systems* 22.5 (1998), pp. 485–496.

[48] 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*. ACM, 2016, pp. 1135–1144.

[49] 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.

[50] Ch Ravi Sekhar and E Madhu. "Mode Choice Analysis Using Random Forrest Decision Trees". In: *Transportation Research Procedia* 17 (2016), pp. 644–652.

[51] PC Sham and D Curtis. "An extended transmission/disequilibrium test (TDT) for multi-allele marker loci". In: *Annals of human genetics* 59.3 (1995), pp. 323–336.

[52] Jasper Snoek et al. "Scalable bayesian optimization using deep neural networks". In: *International Conference on Machine Learning*. 2015, pp. 2171–2180.

[53] Jianping Sun et al. "Analyzing the Impact of Traffic Congestion Mitigation: From an Explainable Neural Network Learning Framework to Marginal Effect Analyses". In: *Sensors* 19.10 (2019), p. 2254.

[54] Christian Szegedy et al. "Going deeper with convolutions". In: Cvpr, 2015.

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

[56] Vladimir Naumovich Vapnik. "An overview of statistical learning theory". In: *IEEE transactions on neural networks* 10.5 (1999), pp. 988–999.

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

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

[59] Shenhao Wang and Jinhua Zhao. "Using Deep Neural Network to Analyze Travel Mode Choice With Interpretable Economic Information: An Empirical Example". In: *arXiv preprint arXiv:1812.04528* (2018).

[60] Ray Weaver and Shane Frederick. "Transaction disutility and the endowment effect". In: *NA-Advances in Consumer Research Volume 36* (2009).

[61] Xin Wu et al. "Hierarchical travel demand estimation using multiple data sources: A forward and backward propagation algorithmic framework on a layered computational graph". In: *Transportation Research Part C: Emerging Technologies* 96 (2018), pp. 321–346. ISSN: 0968-090X.

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

[63] Chiyuan Zhang et al. "Understanding deep learning requires rethinking generalization". In: *arXiv preprint arXiv:1611.03530* (2016).

[64] Zhenhua Zhang et al. "A deep learning approach for detecting traffic accidents from social media data". In: *Transportation research part C: emerging technologies* 86 (2018), pp. 580–596.

[65] Bolei Zhou et al. "Object detectors emerge in deep scene cnns". In: *arXiv preprint arXiv:1412.6856* (2014).

[66] Barret Zoph and Quoc V Le. "Neural architecture search with reinforcement learning". In: *arXiv preprint arXiv:1611.01578* (2016).

Appendix I. Proof of Propositions 1 and 2

Proof of Proposition 1. This proof can be found in all choice modeling textbooks [55, 5]. With Gumbel distributional assumption, Equation 3 could be solved in an analytical way:

$$\begin{aligned} P_{ik} &= \int_{-\infty}^{+\infty} \prod_{j \neq k} e^{-e^{-(V_{ik}-V_{ij}+\epsilon_{ik})}} f(\epsilon_{ik}) d\epsilon_{ik} \\ &= \int \prod_{j} e^{-e^{-(V_{ik}-V_{ij}+\epsilon_{ik})}} e^{-\epsilon_{ik}} d\epsilon_{ik} \\ &= \int exp(e^{-\epsilon_{ik}} \sum_{j} -e^{-(V_{ik}-V_{ij})}) e^{-\epsilon_{ik}} d\epsilon_{ik} \\ &= \int_{\infty}^{0} exp(-t \sum_{j} e^{-(V_{ik}-V_{ij})}) dt \\ &= \frac{e^{V_{ik}}}{\sum_{j} e^{V_{ij}}} \end{aligned} \tag{17}$$

in which the fourth equation uses $t = e^{-\epsilon_{ik}}$. Note this formula in Equation 17 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 2. This proof can be found in lemma 2 of Mcfadden (1974) [37]. 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 and 17,

$$\begin{aligned} P_{ik} &= \frac{e^{V_{ik}}}{e^{V_{ik}} + T e^{V_{ij}}} \\ &= \int F(\epsilon_{ik} + V_{ik} - V_{ij})^{T} dF(\epsilon_{ik}) \end{aligned} \tag{18}$$

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

$$\begin{aligned} P_{ik} &= \frac{e^{V_{ik}}}{e^{V_{ik}} + e^{V_{il}}} \\ &= \int F(\epsilon_{ik} + V_{ik} - V_{il}) dF(\epsilon_{ik}) \\ &= \int F(\epsilon_{ik} + V_{ik} - V_{ij} - logT) dF(\epsilon_{ik}) \end{aligned} \tag{19}$$

By construction, Equations 18 and 19 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. Summary Statistics of SGP and TRAIN

Table 7: Summary Statistics of SGP data set

Variables			**Variables**		
Name	Mean	Std.	Name	Mean	Std.
Male (Yes = 1)	0.383	0.486	Age <35 (Yes = 1)	0.329	0.470
Age>60 (Yes = 1)	0.075	0.263	Low education (Yes = 1)	0.331	0.471
High education (Yes = 1)	0.480	0.500	Low income (Yes = 1)	0.035	0.184
High income (Yes = 1)	0.606	0.489	Full job (Yes = 1)	0.602	0.490
Walk: walk time (min)	60.50	54.88	Bus: cost ($SG)	2.070	1.266
Bus: walk time (min)	11.96	10.78	Bus: waiting time (min)	7.732	5.033
Bus: in-vehilce time (min)	25.06	18.91	RideSharing: cost ($SG)	14.48	11.64
RideSharing: waiting time (min)	7.108	4.803	RideSharing: in-vehilce time (min)	18.28	13.39
AV: cost ($SG)	16.08	14.60	AV: waiting time (min)	7.249	5.674
AV: in-vehilce time (min)	20.11	16.99	Drive: cost ($SG)	10.49	10.57
Drive: walk time (min)	3.968	4.176	Drive: in-vehilce time (min)	17.43	14.10
Statitics					
Number of samples	8418				
Number of choices	Walk: 874 (10.38%); Bus: 1951 (23.18%); RideSharing: 904 (10.74%); Drive 3774 (44.83%); AV: 915 (10.87%)				

Table 8: Summary Statistics of TRAIN data set

Variables					
Name	Mean	Std.	Name	Mean	Std.
Choice1: price (guilders)	3368.3	1296.6	Choice2: price (guilders)	3367.7	1274.3
Choice1: time (min)	127.52	29.13	Choice2: time (min)	127.17	27.96
Choice1: number of changes	0.664	0.733	Choice2: number of changes	0.681	0.743
Choice1: comfort level (0,1 or 2)	0.899	0.602	Choice2: comfort level (0,1 or 2)	0.885	0.617
Statistics					
Number of samples	2928				
Number of choices	Choice1: 1473 (50.31%); Choice2: 1455 (49.69%)				

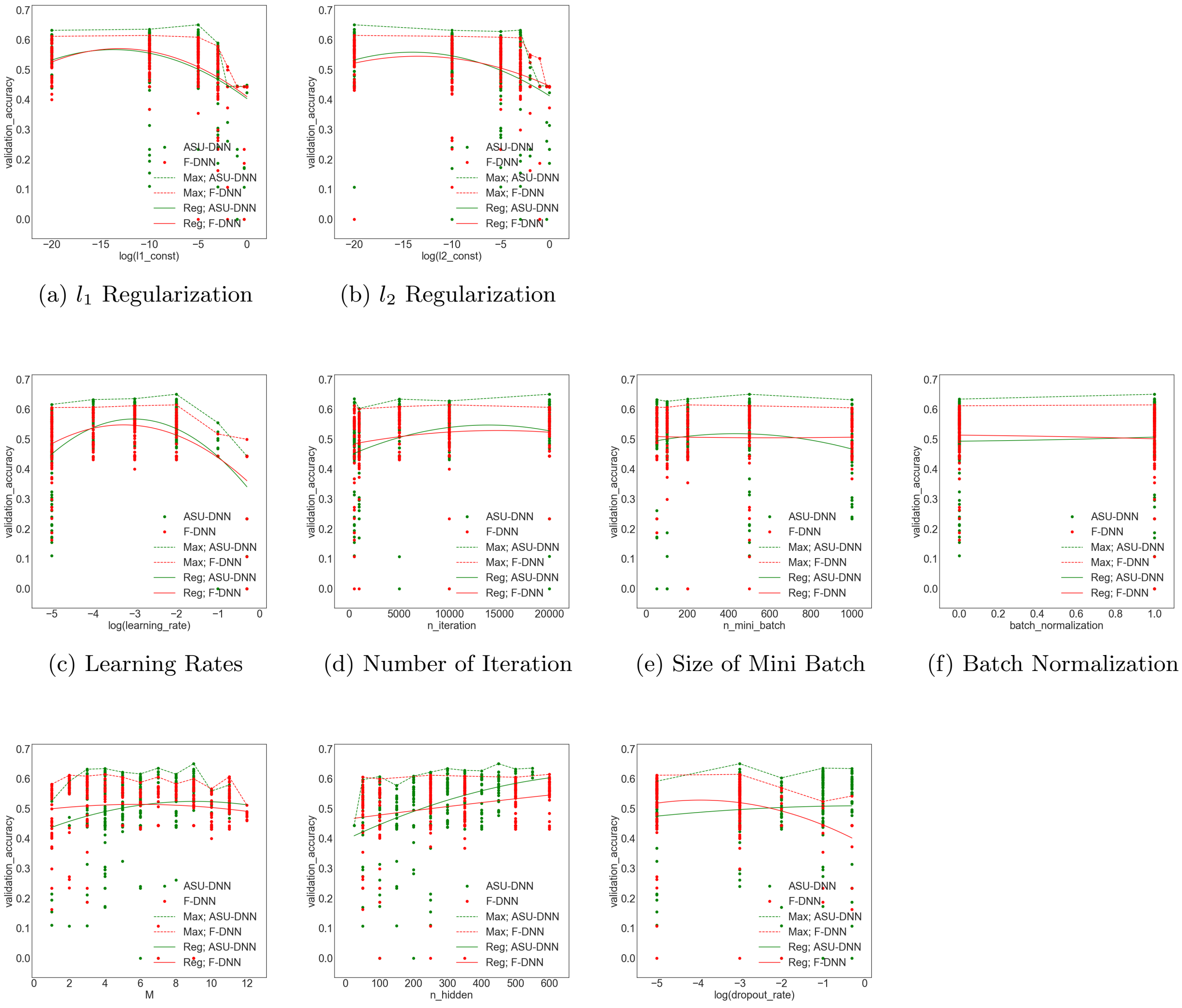
Appendix III. Top Five DNN Architectures

Table 9: Top 5 DNN structures in the SGP data set

	F-DNN					ASU-DNN				
Rank	1	2	3	4	5	1	2	3	4	5
Accuracy (validation)	0.615	0.612	0.609	0.608	0.607	0.651	0.636	0.634	0.633	0.632
M	4	2	3	3	11	-	-	-	-	-
Width n	600	250	350	350	350	-	-	-	-	-
M_1	-	-	-	-	-	5	5	2	3	1
M_2	-	-	-	-	-	3	1	1	5	1
Width n_1	-	-	-	-	-	100	100	60	100	60
Width n_2	-	-	-	-	-	60	100	40	80	40
γ_1 (l_1 penalty)	10^{-10}	10^{-20}	10^{-5}	10^{-5}	10^{-5}	10^{-5}	10^{-10}	10^{-5}	10^{-10}	10^{-20}
γ_2 (l_2 penalty)	$10^{-}20$	10^{-10}	10^{-5}	10^{-5}	10^{-10}	10^{-20}	10^{-20}	$10^{-}20$	10^{-3}	10^{-10}
Dropout rate	10^{-3}	10^{-5}	10^{-3}	10^{-3}	10^{-5}	10^{-3}	0.1	10^{-3}	0.1	10^{-3}
Batch normalization	True	False	True	True	False	True	True	False	True	True
Learning rate	0.01	10^{-3}	10^{-3}	10^{-3}	10^{-4}	0.01	10^{-3}	0.01	10^{-4}	0.01
Num of iteration	10000	500	5000	5000	20000	20000	500	5000	20000	20000
Mini-batch size	200	500	500	500	100	500	500	200	50	1000

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

Figure 6 compares the alternative-specific connectivity regularization to other regularization methods in the validation set of SGP. The results are very similar to Figure 4.



(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

Fig. 6. Comparing Alternative-Specific Connectivity to Explicit Regularizations, Implicit Regularizations, and Architectural Hyperparameters in SGP Validation Set; *First Row*: Explicit regularizations; *Second Row*: Implicit regularizations; *Third Row*: Architectural hyperparameters. All results are similar to those in testing set.