Multitask Learning Deep Neural Networks to Combine Revealed and Stated Preference Data

Shenhao Wang*
Qingyi Wang
Jinhua Zhao
Massachusetts Institute of Technology
77 Mass Ave, Cambridge, Massachusetts, U.S.

Abstract

It is an enduring question how to combine revealed preference (RP) and stated preference (SP) data to analyze individual choices. While the nested logit (NL) model is the classical way to address the question, this study presents multitask learning deep neural networks (MTLDNNs) as an alternative framework, and discusses its theoretical foundation, empirical performance, and behavioral intuition. We first demonstrate that the MTLDNNs are theoretically more general than the NL models because of MTLDNNs' automatic feature learning, flexible regularizations, and diverse architectures. By analyzing the adoption of autonomous vehicles (AVs), we illustrate that the MTLDNNs outperform the NL models in terms of prediction accuracy but underperform in terms of cross-entropy losses. To interpret the MTLDNNs, we compute the elasticities and visualize the relationship between choice probabilities and input variables. The MTLDNNs reveal that AVs mainly substitute driving and ride hailing, and that the variables specific to AVs are more important than the socio-economic variables in determining AV adoption. Overall, this work demonstrates that MTLDNNs are theoretically appealing in leveraging the information shared by RP and SP and capable of revealing meaningful behavioral patterns, although its performance gain over the classical NL model is still limited. To improve upon this work, future studies can investigate the inconsistency between prediction accuracy and cross-entropy losses, novel MTLDNN architectures, regularization design for the RP-SP question, MTLDNN applications to other choice scenarios, and deeper theoretical connections between choice models and the MTLDNN framework.

Keywords: multitask learning deep neural network, machine learning, revealed preference, stated preference, autonomous vehicles

*Corresponding Author; Email: shenhao@mit.edu

1. Introduction

For decades, researchers have been combining revealed preference (RP) and stated preference (SP) data to analyze individual behavior, owing to their complementary properties. RP data are thought to have stronger external validity but often lack the variation in attributes or alternatives, while SP data often incorporate new attributes or alternatives but lack strong external validity. As a classical method, the nested logit (NL) model has been commonly used to combine RP and SP by assigning their alternatives to two nests with different utility scale factors [26, 11, 6, 7].[1] However, in the NL model, researchers need to analyze RP and SP by handcrafting the model structure, which can be too restrictive to capture the complex data generating process. This handcrafted feature engineering is different from the mechanism in deep neural networks (DNNs) [33, 8, 14], which can automatically learn generalizable features to achieve outstanding predictive performance across disciplines [16, 32, 33]. The recent innovations in DNNs prompt us to investigate the possibility of using a DNN framework to address the classical problem of combining RP and SP, as an alternative to the traditional NL method.

This study presents a framework of multitask learning deep neural networks (MTLDNNs) to jointly analyze RP and SP, demonstrating MTLDNNs' theoretical flexibility, empirical performance, and behavioral intuition. A MTLDNN architecture starts with shared layers capturing the similarities between RP and SP, and ends with task-specific layers capturing their differences (Figure 1) [12]. We first demonstrate that MTLDNNs are theoretically more general than NL owing to their automatic feature learning, soft constraints, and diverse architectures. Then we apply the MTLDNN framework to a data set collected in Singapore, which was designed to analyze the adoption of autonomous vehicles (AVs). In the empirical experiments, we compare the MTLDNNs to two NL benchmarks using prediction accuracy and cross-entropy loss.[2] To understand the determinants of AV adoption, we visualize the relationship between choice probabilities and input variables and compute the elasticity values using MTLDNNs' gradients information [3, 47]. Overall, our analysis demonstrates that the MTLDNNs are theoretically appealing in leveraging the shared information between RP and SP, and are capable of revealing meaningful behavioral patterns, although the gain in empirical performance, particularly measured by cross-entropy losses, is still limited.

This study contributes to the choice modeling community by being the first to present the MTLDNN framework in the important context of combining RP and SP. Future studies can investigate deeper theoretical and empirical questions revolving around this topic. Particularly, future researchers should investigate the inconsistency between prediction accuracy and cross-entropy losses, because the two metrics represent the different perspectives from machine learning and classical choice modeling. Researchers can also investigate the classical theoretical question (e.g. modeling the structure of random utility terms of RP and SP) under this MTLDNN framework

[1] This nested logit method can also be seen as a pooled estimation with heteroscedasticity across RP and SP [36, 25]

[2] Cross-entropy loss is the same as negative log likelihood, so minimizing the cross-entropy loss is the same as maximizing log likelihood.

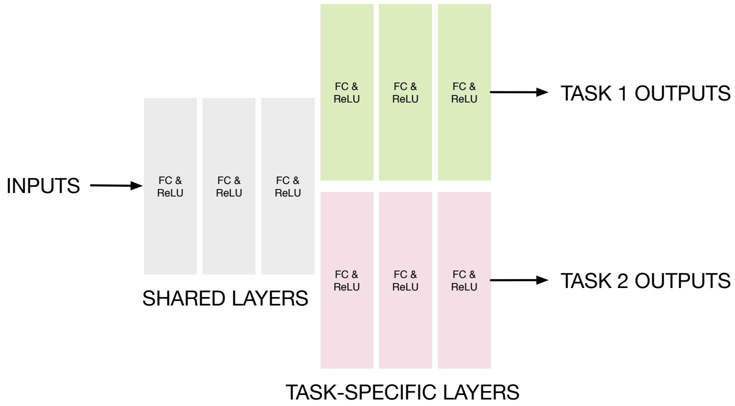


Fig. 1. MTLDNN architecture; this architecture has 3 shared and 3 task-specific layers, but they represent generally M_1 shared and M_2 task-specific layers; FC stands for fully connected layers; ReLU for Rectified Linear Units. This study uses RP as Task 1 and SP as Task 2.

and improve the empirical performance of this study by using advanced MTLDNN architectures [35, 22, 38, 46]. Future studies can also apply the MTLDNN framework to other choice scenarios, such as jointly analyzing car ownership and travel mode choice [51, 67], activity patterns and trip chain choices [31, 20], and many others that are traditionally analyzed by structural equation models (SEM). For future researchers to replicate and improve upon our work, we have uploaded the project to Github.[3]

This paper is organized as following. Section 2 reviews the MTLDNN and NL models. Section 3 presents the MTLDNN framework and compares its theoretical properties to the NL models. Section 4 presents data and methods, and Section 5 analyzes model performance and presents the economic information in MTLDNNs. Section 6 concludes our findings and discusses future research directions.

2. Literature Review

For travel demand analysis, RP and SP data are important but associated with different sources of problems. The RP data can have limited coverage of values, high correlation between attributes, and poor quality of background information [7], although it typically has better external validity. In the SP data, respondents could fail to provide valid answers because of their sensitivity to survey formats, unrealistic hypothetical scenarios [48], or even just measurement errors that happen in nearly all the data collection processes [24, 23]. However, SP is the only viable way to analyze new pricing strategies, new public transit services, or new travel modes [6, 43].

To address these problems, one common remedy is to jointly estimate RP and SP, thus gaining efficiency and correcting biases [7]. The NL model has been used as a classical method by treating RP and SP as two nests of choices [26, 11, 43, 39]. For example, Polydoropoulou and Ben-Akiva

[3] https://github.com/cjsyzwsh/Multitask-learning-deep-neural-networks-to-combine-revealed-and-stated-preference-data.git

(2001) used the NL approach to analyze the travel mode choices for multiple mass transit technologies [43]. Golob et al. (1997) [19] used the same method to examine how vehicle usage depends on the factors of the vehicles and fuel types. In these studies, researchers make parametric assumptions in the NL models to capture the differences and similarities between RP and SP [11]. To capture the similarities, RP and SP choice models can share parameters, such as the same price and time coefficients [43]. To capture the differences, RP and SP models can have different randomness in their error terms, causing the different magnitudes of coefficients [26, 11]. While modeling details vary with studies, the NL approach developed in the 1990s has become a standard way of combining RP and SP [48, 36, 62, 52].

From a machine learning perspective, we can use a MTLDNN framework to combine RP and SP, because the MTLDNN framework can be widely applied to any *related* tasks. For example, researchers used MTLDNNs to jointly analyze the steering directions and road conditions for driving, jointly locate doorknobs and identify door types by using shared door images [12], and jointly analyze the ratings of 20 computers by using 13 shared binary attributes [2]. In natural language processing, researchers used the MTLDNN framework to jointly learn different levels of semantic components, such as part-of-speech tags, word chunks, and named entity tags [14, 22]. In image recognition, researchers used the MTLDNN to jointly analyze semantic segmentation and surface normal prediction, and jointly detect objects and predict attributes [38]. Sometimes the multiple tasks are different only in collection procedures. For example, the four tasks in Long et al. (2015) are the same in terms of inputs and outputs but different because they were collected through four different online channels [35]. The 259 tasks in Ramsundar et al. (2015) refer to the 259 data sets that were collected differently but share similar biological purposes, such as predicting drug toxicity and protein molecules. Interestingly, while the MTLDNN framework seems intuitive and has been developed for decades in the ML community, it is relatively less known in the community of choice modeling.

The MTLDNN framework is similar to the simultaneous estimation of choice models, although the similarity has never been explicitly discussed. In fact, it is possible to use the MTLDNN framework to jointly analyze auto ownership and mode choice [51, 67], trip chains and travel modes [65], travel time and vehicle miles traveled (VMT) [18], travel mode choices and attitudinal factors [37, 39, 50], and activity patterns and travel demands [31, 20]. It is because all these tasks are similar travel behavior, thus containing valuable shared information. Despite the intuition, no study has applied the MTLDNN framework to jointly analyze these travel behaviors in choice modeling yet.

Many MTLDNN architectures have been created in the past three decades. Caruana (1997) [12] first created a benchmark MTLDNN architecture, which starts with shared layers and ends with task-specific layers (Figure 1). Caruana's initial MTLDNN architecture was further improved by recent studies [35, 22, 38, 46], which designed regularizations and network components to control the similarities and differences of the multiple tasks in a more specific manner [38, 46, 2, 66, 15, 29, 64, 30, 35]. Despite the vast number of MTLDNN architectures, our study uses only the basic

MTLDNN architecture because it allows a straightforward comparison with the NL method, as discussed below.

3. Theory

3.1. Multitask Learning Deep Neural Network for RP and SP

Let $x_{r,i}, x_{s,t} \in R^d$ denote the input variables for RP and SP respectively, where r and s stand for RP and SP, $i \in \{1, 2, ..., N_r\}$ and $t \in \{1, 2, ..., N_s\}$ are the indices of RP and SP observations, and d represents the input dimension. The output choices of RP and SP are denoted by $y_{r,i}$ and $y_{s,t}$; $y_{r,i} \in \{0, 1\}^{K_r}$ and $y_{s,t} \in \{0, 1\}^{K_s}$; K_r and K_s are the dimensions of the outputs. In our case, SP has more alternatives than RP since SP includes a new product that is not available in the existing market ($K_s > K_r$). Both $y_{r,i}$ and $y_{s,t}$ are vectors taking binary values, and each component in $y_{r,i}$ and $y_{s,t}$ is denoted by $y_{k_r,i} \in \{0, 1\}$ and $y_{k_s,t} \in \{0, 1\}$. Due to the constraint of mutually exclusive and collectively exhaustive alternatives, $\sum_{k_s} y_{k_s,t} = 1$ and $\sum_{k_r} y_{k_r,i} = 1$. k_r and k_s are the index of alternatives in RP and SP, so $k_r \in \{1, 2, ..., K_r\}$ and $k_s \in \{1, 2, ..., K_s\}$. As represented by Figure 1, the feature transformation of RP and SP can be represented as:

$$V_{k_r,i} = (g_r^{M_2,k_r} \circ g_r^{M_2-1} \circ ... \circ g_r^1) \circ (g_0^{M_1} \circ g_0^{M_1-1} \circ ... \circ g_0^1)(x_{r,i}) \tag{1}$$

$$V_{k_s,t} = (g_s^{M_2,k_s} \circ g_s^{M_2-1} \circ ... \circ g_s^1) \circ (g_0^{M_1} \circ g_0^{M_1-1} \circ ... \circ g_0^1)(x_{s,t}) \tag{2}$$

in which M_1 represents the depth of the shared layers and M_2 the depth of the task-specific layers; g_0 represents the transformation of one shared layer; g_r and g_s represent the transformation of one layer in RP and SP. Specifically, g functions (including g_r, g_s, and g_0) are the composition of ReLU and linear transformation: $g^l(x) = \max\{W^l x, 0\}$, $\forall l \neq M_2$. Equations 1 and 2 describe precisely the MTLDNN architecture in Figure 1: $(g_0^{M_1} \circ g_0^{M_1-1} \circ ... \circ g_0^1)$ represent the shared layers, while $(g_r^{M_2,k_r} \circ g_r^{M_2-1} \circ ... \circ g_r^1)$ and $(g_s^{M_2,k_s} \circ g_s^{M_2-1} \circ ... \circ g_s^1)$ represent task-specific layers. The choice probability functions in RP and SP can be represented by

$$P(y_{k_r,i}; w_r, w_0) = \frac{e^{V_{k_r,i}}}{\sum_{j_r=1}^{K_r} e^{V_{j_r,i}}} \tag{3}$$

$$P(y_{k_s,t}; w_s, w_0, T) = \frac{e^{V_{k_s,t}/T}}{\sum_{j_s=1}^{K_s} e^{V_{j_s,t}/T}} \tag{4}$$

in which w_r and w_s represent the task-specific parameters in g_r and g_s; w_0 the shared parameters in g_0. Equation 3 takes the form of a standard Softmax activation function, while that of SP (Equation 4) is adjusted by a T factor, which is referred to as temperature in the DNN literature to adjust the scale of logits [28].

With choice probabilities formulated, we train the model by minimizing the empirical risk

(ERM) with regularization terms:

$$
\begin{aligned}
\min_{w_r,w_s,w_0,T} R(X,Y;w_r,w_s,w_0,T;c_H) &= \min_{w_r,w_s,w_0,T}\Big\{\hat{L}_R(w_r,w_0)+\lambda_0\hat{L}_S(w_s,w_0)+\lambda^T g(w_r,w_s,w_0)\Big\} \\
&= \min_{w_r,w_s,w_0,T}\Big\{-\frac{1}{N_r}\sum_{i=1}^{N_r}\sum_{k_r=1}^{K_r} y_{k_r}\log P(y_{k_r,i};w_r,w_0;c_H) \\
&\qquad -\frac{\lambda_0}{N_s}\sum_{t=1}^{N_s}\sum_{k_s=1}^{K_s} y_{k_s}\log P(y_{k_s,t};w_s,w_0,T;c_H) \\
&\qquad +\lambda_1||w_0||_2^2+\lambda_2||w_s||_2^2+\lambda_3||\tilde{w}_s-w_r||_2^2\Big\}
\end{aligned} \tag{5}
$$

Equation 5 consists of three parts. The first part

$$\hat{L}_R(w_r,w_0) = -\frac{1}{N_r}\sum_{i=1}^{N_r}\sum_{k_r=1}^{K_r} y_{k_r}\log P(y_{k_r,i};w_r,w_0;c_H)$$

is the empirical risk of RP. The second part

$$\lambda_0\hat{L}_S(w_s,w_0) = -\frac{\lambda_0}{N_s}\sum_{t=1}^{N_s}\sum_{k_s=1}^{K_s} y_{k_s}\log P(y_{k_s,t};w_s,w_0,T;c_H)$$

is the empirical risk of SP with λ_0 weight. Note that the empirical risks $\hat{L}_R(w_r,w_0)$ and $\hat{L}_S(w_s,w_0)$ are taking the form of cross-entropy losses, which are simply the negative values of the log likelihood in the classical maximum likelihood estimation. Hence minimizing the cross-entropy losses is exactly the same as maximizing the log likelihood. The third part

$$\lambda^T g(w_r,w_s,w_0) = \lambda_1||w_0||_2^2+\lambda_2||w_s||_2^2+\lambda_3||\tilde{w}_s-w_r||_2^2$$

is the regularization. Equation 5 incorporates four hyperparameters (λ_0, λ_1, λ_2, λ_3) for explicit regularizations. λ_0 adjusts the ratio of empirical risks between RP and SP. This study treats equally one observation in RP and SP by fixing $\lambda_0 = 1$.[4] λ_1 and λ_2 jointly adjust the absolute magnitudes of the shared layers and SP-specific layers: larger λ_1 and λ_2 lead to larger weight decay, reducing the estimation error of the complex DNN models [56]. λ_3 controls the degree of similarity between RP- and SP-specific layers. As λ_3 becomes very large, ERM penalizes more the large differences between RP- and SP-specific layers, leading to more shared RP-SP similarities. Since w_s and w_r do not match perfectly in our case, $\tilde{w}_s$ is used to denote the SP-specific weights that are corresponding to those RP-specific weights. The ERM formulation and the regularizations in Equation 5 are commonly used in MTLDNN studies [15, 29].

[4] Researchers are free to choose the value of λ_0, since there is no clear-cut rule for its value specification. Our choice reflects our belief that each individual counts as equal in RP and SP.

3.2. Nested Logit Model for RP and SP

Similar to the past studies [26, 11, 43, 39], the utility functions of the NL model are assumed to be the following:

$$U_{k_r,i} = V_{k_r,i} + \epsilon_{k_r} = w_{k_r}^T \phi(x_{r,i}) + \epsilon_{k_r,i} \tag{6}$$

$$U_{k_s,t} = V_{k_s,t} + \epsilon_{k_s} = w_{k_s}^T \phi(x_{s,t}) + \epsilon_{k_s,t} \tag{7}$$

in which w_{k_r} and w_{k_s} are the parameters for RP and SP; ϕ denotes the handcrafted feature transformation; for example, ϕ can represent the quadratic transformation, when researchers believe there exists nonlinear relationship between utilities and input variables. $\epsilon_{k_r,i}$ and $\epsilon_{k_s,t}$ are random utility terms. It is commonly assumed that $\epsilon_{k_r,i}$ and $\epsilon_{k_s,t}$ are off by a scale factor:

$$Var(\epsilon_{k_r,i})/Var(\epsilon_{k_s,t}) = 1/\theta^2 \tag{8}$$

The choice probability functions thus become:

$$P(y_{k_r,i}; w_r) = \frac{e^{w_{k_r}^T \phi(x_{r,i})}}{\sum_{j_r=1}^{K_r} e^{w_{j_r}^T \phi(x_{r,i})}} \tag{9}$$

$$P(y_{k_s,t}; w_s, \theta) = \frac{e^{w_{k_s}^T \phi(x_{s,t})/\theta}}{\sum_{j_s=1}^{K_s} e^{w_{j_s}^T \phi(x_{s,t})/\theta}} \tag{10}$$

Here w_r and w_s represent the parameters in RP and SP. Note that θ is similar to the temperature factor T in the MTLDNN framework, although θ arises from the assumption about the variance of the random error terms while T does not. The ERM in the NL model is

$$\min_{w_r,w_s,\theta} R(X,Y;w_r,w_s,\theta) = \min_{w_r,w_s,\theta} \Big\{ \hat{L}_R(w_r) + \hat{L}_S(w_s,\theta) \Big\} \tag{11}$$

$$= \min_{w_r,w_s,\theta} \Big\{ -\frac{1}{N} \big[\sum_{i=1}^{N_r} \sum_{k_r=1}^{K_r} y_{k_r,i} \log P(y_{k_r,i}; w_r) + \sum_{t=1}^{N_s} \sum_{k_s=1}^{K_s} y_{k_s,t} \log P(y_{k_s,t}; w_s, \theta) \big] \Big\} \tag{12}$$

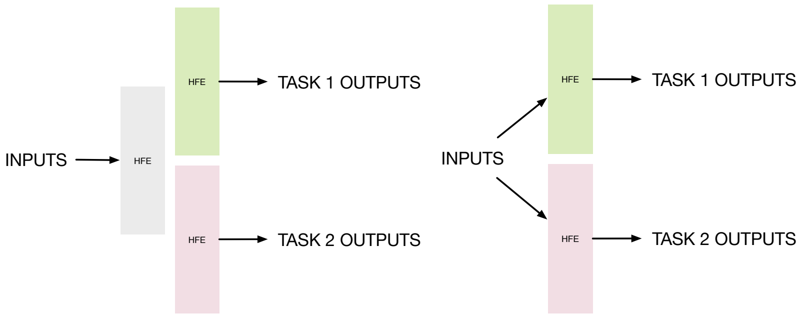
This NL formulation is not the same as the standard NL model, since respondents do not face all the RP and SP alternatives in one choice scenario. Therefore, researchers named this NL approach as an "artificial nested logit" model, the details of which are available in [26, 11].

3.3. Similarities and Differences between MTLDNN and NL

MTLDNN and NL are similar in terms of the underlying behavioral intuition, but they are different in terms of parameter constraints, learning capacity, and estimation errors. The following four subsections respectively introduce the four perspectives.

3.3.1. Similar Behavioral Intuition

MTLDNN and NL share a similar behavioral intuition because both improve the training efficiency by leveraging the shared information between multiple tasks. In fact, Equations 6 and 7 can be visualized in Figure 2a, in which the grey layer represents the $\phi()$ transformation and the green and red layers represent w_r and w_s multiplication. When researchers use only an identity mapping $\phi(x) = x$, Equations 6 and 7 can be visualized in Figure 2b, in which inputs are directly fed into task-specific layers. Therefore, NL models can be treated as shallow MTLDNNs, which use a single shared layer to capture the similarities of RP and SP and a single task-specific layer to capture their differences. With the shared information captured by the shared layers, both MTLDNNs and NL models can fit the RP and SP data efficiently.



(a) NL with $\phi(x)$ (b) NL without $\phi(x)$

Fig. 2. Visualization of NL; HFE stands for handcrafted feature engineering

3.3.2. Different Parameter Constraints

MTLDNNs use soft constraints to capture the similarities between RP and SP, while the NL models rely on handcrafted hard constraints. By defining $w_{k_{sh}}, w_{k_r}, w_{k_s}$ as the shared, RP-specific, and SP-specific parameters and using x_{sh}, x_r and x_s to denote the shared, RP-specific, and SP-specific variables, the utility functions of the NL model (Equations 6 and 7) can be rewritten as:

$$U_{k_r,i} = V_{k_r,i} + \epsilon_{k_r} = w_{k_{sh}}^T \phi(x_{sh,i}) + w_{k_r}^T \phi(x_{r,i}) + \epsilon_{k_r,i} \tag{13}$$

$$U_{k_s,t} = V_{k_s,t} + \epsilon_{k_s} = w_{k_{sh}}^T \phi(x_{sh,t}) + w_{k_s}^T \phi(x_{s,t}) + \epsilon_{k_s,t} \tag{14}$$

The shared and domain-specific variables/parameters in NL models are designed by using domain knowledge; for example, the coefficients for travel time can be specified as the same between RP and SP. On the contrary, MTLDNNs use soft constraints without handcrafted adjustments; for example, $\lambda_3||w_r - w_s||^2$ in Equation 5 controls the overall distance between the vectors of w_r and w_s without specifying how the individual elements of w_r and w_s differ. Since the hard constraints can be seen as the boundary cases of the soft ones, the soft constraints in MTLDNNs are more generic than the hard ones.

3.3.3. Different Learning Capacities

MTLDNNs have a much stronger learning capacity than NL, because MTLDNNs have a deep structure while the NL is shallow. The deep layer-by-layer structure is more general than the shallow one-layer NL models (Equations 6 and 7), because a deep structure can be reduced to a shallow one when the layers after the first layer become identity mapping. Studies show that deep architectures can represent the same function as shallow ones with an exponentially smaller number of neurons [13, 34, 45], which explains why MTLDNNs have stronger learning capacity than NL. In addition, MTLDNNs' deep architectures enable more flexibility to capture the RP-SP similarities. For example, the prototype MTLDNN framework (Figure 1) can flexibly control the extent to which the models of RP and SP are similar by varying the numbers of shared layers M_1 and task-specific layers M_2. This flexibility exists owing to the depth of the MTLDNN framework, whereas the NL architecture in Figure 2 does not have this flexibility because of its shallowness.

3.3.4. Different Estimation Errors

MTLDNNs as a more generic model family do not necessarily imply a higher prediction accuracy, since small approximation errors obtained by a model with large learning capacity can be counteracted by large estimation errors. Based on statistical learning theory, a more complex model (e.g. MTLDNN) typically has a smaller approximation error (bias) but a larger estimation error (variance) than a simple one (e.g. NL) [58, 56].[5] The learning capacity of a model is formally measured by the Vapnik-Chervonenkis (VC) dimension [55, 56]. The VC dimension of DNNs is roughly proportional to its number of parameters and its depth, so the VC dimension of a simple 5-layer DNN with 100 neurons in each layer is $c_0 \times 250,000$ ($O(100^2 \times 5 \times 5)$) [5]. On the other side, the VC dimension of a NL model is proportional to its number of parameters: with about 20 input variables, the VC dimension of NL is only about $c_1 \times 20$. While this VC dimension perspective is not the optimum upper bound on the estimation error [21, 41], it provides adequate insights for the purpose of this paper.[6] While MTLDNNs are more generic than NL in terms of the function class relationship [13, 34, 45], MTLDNNs could perform worse due to its high model complexity and the corresponding large estimation errors. Therefore, empirical experiments are necessary to compare the performance of MTLDNNs and NL.

4. Data and Methods

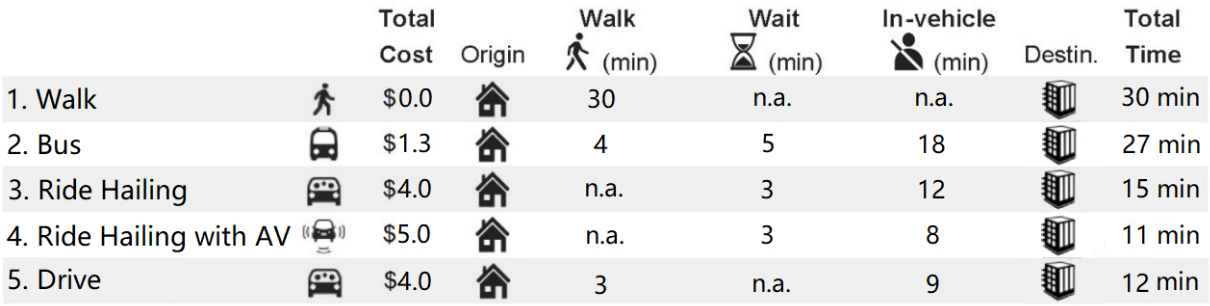
4.1. Data Collection

A survey was conducted through Qualtrics.com in July 2017 to analyze the mode choice preferences to autonomous vehicles (AVs) in Singapore. The survey consisted of three sections: RP, SP, and

[5] This tradeoff is traditionally known as bias-variance tradeoff. Bias is similar to the approximation error, and variance is similar to the estimation error

[6] For a more general introduction, readers could refer to the recent studies in the fields of high dimensional probability and statistics [59, 57, 4, 1]

respondents' demographics. In the first part (RP), the respondents reported the zip codes of the origin and destination (OD) of their most recent trip with a specific trip purpose, which was randomly drawn from commuting, shopping, or recreation, along with the travel mode choice of the trip that was chosen from walking, public transit, driving, and ride hailing. By using the OD information, we computed the trip characteristics from Google Map API, including walking time for walking, public transit, and driving; waiting time for public transit and ride hailing; in-vehicle travel time for public transit, ride hailing, and driving; and travel cost for public transit, ride hailing, and driving. In the second part (SP), hypothetical choice scenarios (Figure 3) were automatically created in the survey and on-demand AV was added to the choice set using the computed trip characteristics. In the choice scenarios, trip-specific attributes took three values with the middle level equal to the value generated from the RP section so that the scenarios were realistic to the survey respondents, and the other two values were 50% and 150% of the middle levels. The trip attributes of AVs were taken to be of a similar magnitude as ride hailing. Among all possible choice scenarios, six scenarios were randomly drawn for each respondent to make mode choice decisions, following a random experimental design. This random experiment design is not as efficient as the SP design with fractional factorial design, and the SP experiment deriving values from the RP (SP-off-RP) can lead to inconsistent estimates due to the unobserved factors shared by RP and SP [53, 27]. But it is still unclear how these statistical concerns influence the predictive performance and the interpretation of the MTLDNNs. In the last section of the survey, the respondents reported their socioeconomic information, such as age, gender, and income.



	Total Cost	Origin	Walk (min)	Wait (min)	In-vehicle (min)	Destin.	Total Time
1. Walk	$0.0	🏠	30	n.a.	n.a.	🏢	30 min
2. Bus	$1.3	🏠	4	5	18	🏢	27 min
3. Ride Hailing	$4.0	🏠	n.a.	3	12	🏢	15 min
4. Ride Hailing with AV	$5.0	🏠	n.a.	3	8	🏢	11 min
5. Drive	$4.0	🏠	3	n.a.	9	🏢	12 min

Fig. 3. Example choice scenario in the survey

4.2. Data Summary

A significantly larger number of respondents chose driving and a significantly smaller number chose public transit in SP than RP. As shown by the market shares of the travel mode choices in Table 1, only 1.73% of the total respondents drove as reported in RP, as opposed to 35.6% in SP; about 58.8% chose public transit in RP, as opposed to 28.8% in SP. Given that the average values of the alternative-specific variables (e.g. cost, travel time, etc.) are the same between RP and SP, the difference of mode shares is likely to be caused by the constraints in the RP setting, such as the availability issue of automobiles in Singapore.

To investigate closely the mode switching behavior, Table 1 cross tabulates the choices in RP (columns) and SP (rows), with the proportion of the respondents who chose consistent travel modes highlighted in bold. Table 1 reveals that a considerable number of people changed from public transit in RP to driving in SP. The drivers in RP are highly likely to choose driving in SP and people from each of the other three modes switch to driving in SP for around 35% of the chance. This change can be caused by the strict restrictions on car ownership in Singapore. To address this challenge, our models explicitly use car availability as a constraint in both NL and MTLDNN models, which will be detailed in the next section. The summary statistics of the independent variables are provided in Appendix I.

Table 1: Cross tabulation of mode choice shares in RP and SP

	Walk (RP)	**Public Transit (RP)**	**Ride Hailing (RP)**	**Drive (RP)**	**SP Share**
Walk (SP)	**463**	265	11	20	759 (12.9%)
Public Transit (SP)	375	**1229**	80	10	1694 (28.8%)
Ride Hailing (SP)	179	397	**104**	8	688 (11.7%)
Drive (SP)	642	1206	194	**53**	2095 (35.6%)
AV (SP)	147	365	127	11	650 (11.0%)
RP Share	1806 (30.7%)	3462 (58.8%)	516 (8.77%)	102 (1.73%)	(100.00%)

4.3. Experiment Setup

Our experiments compare the MTLDNNs to two NL models that specify the utility functions as introduced in Appendix II. Both NL models take linear forms, but differ in the parameter sharing between RP and SP. The NL models with and without parameter constraints (NL-C and NL-NC) represent respectively the NL with the most and the least parameter sharing between RP and SP. The two NL models were designed to represent the two boundary NL models to guarantee a fair comparison between MTLDNNs and NLs.

One challenge in training MTLDNNs is its vast number of hyperparameters that define their regularization and architecture, on which the performance of MTLDNNs largely depends. To address this challenge, we specified a hyperparameter space and searched randomly within this space to identify the hyperparameters that lead to the high prediction accuracy [10]. The hyperparameter space is presented in Appendix III, and the hyperparameters associated with the top 10 MTLDNNs are provided in Appendix IV. To evaluate the models, RP and SP data were split into training and testing sets with the ratio of 5:1. The training set was used for training and the testing set for evaluation.

Non-availability of alternatives is addressed in NL and MTLDNN models by excluding the driving alternative from the Softmax activation function when the respondent does not own a driver licence and a car. The unavailable driving options account for about 39.2% of the RP training set and 37.1% of the RP testing set. For implementation, we used the non-availability option in Python Biogeme for the NL models, and modified the choice probability functions (Equation 3) in the ERM of MTLDNNs using the Tensorflow module in Python.

5. Experiment Results

5.1. Model Performance

Table 2 summarizes the prediction accuracy and cross-entropy losses of MTLDNN (Top 1), MTLDNN ensemble over top 10 models (MTLDNN-E), NL with parameter constraints (NL-C), and NL with no parameter constraints (NL-NC). In Table 2, Panel 1 reports the joint prediction accuracy for RP and SP, individual RP, and individual SP data in the testing and training sets; Panel 2 reports the cross-entropy loss for the joint RP and SP data set.

Table 2: Comparison of four models

	MTLDNN (Top1)	**MTLDNN-E (Top10)**	**NL-C**	**NL-NC**
Panel 1: Prediction Accuracy (Hit-Rate)				
Joint RP+SP (Testing)	60.3%	59.3%	54.7%	55.3%
RP (Testing)	64.6%	61.0%	57.8%	59.7%
SP (Testing)	59.5%	59.0%	54.1%	54.5%
Joint RP+SP (Training)	62.7%	68.9%	53.9%	54.5%
RP (Training)	70.7%	83.3%	61.3%	64.3%
SP (Training)	61.2%	66.2%	52.5%	52.6%
Panel 2: Cross Entropy Loss (Negative Log Likelihood)				
Joint RP+SP (Training)	1.77	1.78	0.948	0.928
Joint RP+SP (Testing)	1.91	1.94	0.984	0.985

The MTLDNNs outperform the NL models in terms of prediction accuracy, but underperform in terms of cross-entropy losses. Measured by the prediction accuracy, the top 1 MTLDNN model outperforms the NL-C and NL-NC by 5.6% and 5.0%. This about 5% prediction gain of MTLDNNs over NL models is consistent in the separate RP and SP datasets. The MTLDNN-E also has higher prediction accuracy than the NL models in the testing sets, although MTLDNN-E performs slightly worse than the top 1 MTLDNN model. Comparing between two NL models, the NL-NC model with fewer parameter constraints outperforms the NL-C model by 0.6% in both the testing and training sets, which is reasonable in the classical statistical framework because releasing constraints improves the capacity of models fitting the data. However, MTLDNNs perform worse than the two NL models in terms of cross entropy losses. In both the training and testing sets, the cross-entropy losses of both MTLDNN and MTLDNN-E are higher than NL-C and NL-NC: that of the best MTLDNN is larger than the NL-C and NL-NC by 0.925 and 0.945 respectively. Interestingly, this worse performance of MTLDNNs seems not caused by overfitting, because the cross-entropy losses of MTLDNNs in the training and testing sets are of a similar magnitude (1.77 vs. 1.91 for the best MTLDNN).

We can understand the inconsistency between prediction accuracy and cross-entropy losses by their different weighting mechanisms. The prediction accuracy uses zero-one losses, which essentially assign the same weights to all the observations; however, the cross-entropy loss is computed by using the formula $-y_i \log P_i$, which assigns a very large weight to the confident but incorrect predictions. For example, when an observation has $y_i = 1$ but predicted choice probability is $P_i = 0.001$ (0.001 chance to be one), the cross-entropy loss is $-y_i \log P_i = 6.91$ while the zero-one loss (prediction error) only equals to one. Therefore, the cross-entropy losses significantly penalize the confident but incorrect predictions with the $\log P_i$ weighting, while the zero-one losses do not.

It is difficult to argue for the importance of one metric over the other. From a theoretical perspective, cross-entropy losses are the same as the negative log likelihood in the classical maximum likelihood estimation, so they have better information theoretical interpretation than prediction accuracy. In fact, Train (2009) [52] illustrated a clear preference of cross-entropy losses over prediction accuracy, which was referred to as "percent correctly predicted" and "should actually be avoided" in practice. But on the other side, prediction accuracy is the most widely adopted metric in the machine learning field, and it is the only metric that allows a wide comparison across data sets and variable types.[7] Despite the difficulty of choosing a definitive metric, it is important to recognize that prediction accuracy and cross-entropy losses can lead to different or even opposite model selection results.

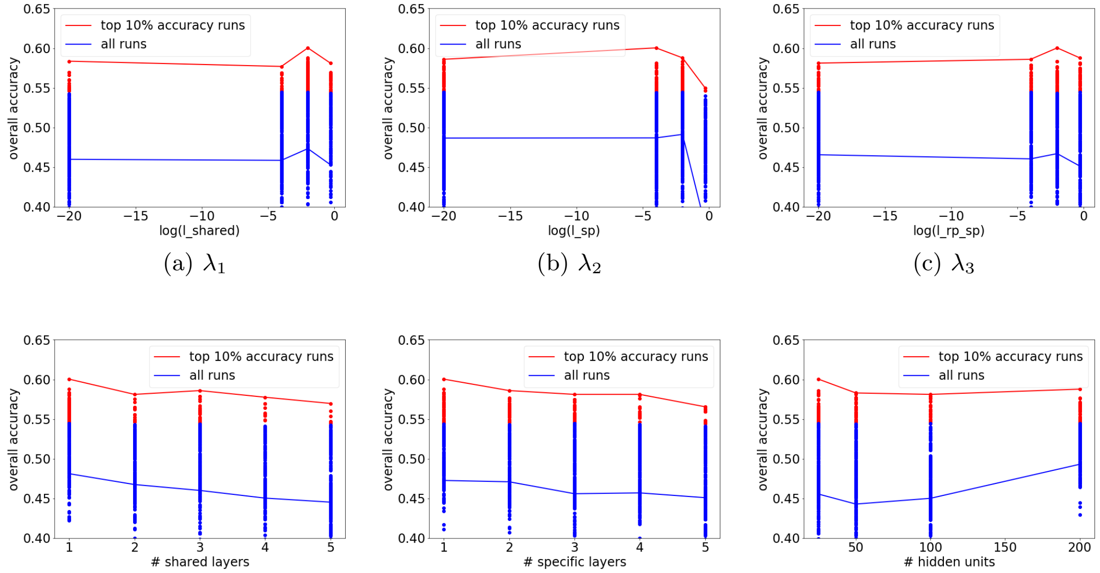
5.2. Hyperparameters in MTLDNNs

Unlike the NL models, hyperparameters of the MTLDNNs have a significant impact on their performance. Figure 4 summarizes how the prediction accuracy of MTLDNNs varies with the regularization constraints and the architectural hyperparameters. It assists in identifying the effective hyperparameters that contribute to MTLDNNs' empirical performance.

When appropriately chosen, the regularization hyperparameters, such as λ_1, λ_2, and λ_3, can effectively improve the prediction accuracy of MTLDNNs. As discussed in Section 3, λ_1 and λ_2 control the absolute scales of RP and SP models, and λ_3 is the penalty term on the similarity between the task-specific layers of RP and SP. As shown in Figures 4a-4c, when λ_3 becomes too large or small, implying that RP- and SP-specific layers are either too similar or different, the MTLDNN model cannot perform well. This finding also similarly applies to λ_1 and λ_2. In our case, the best λ_1, λ_2, and λ_3 values are respectively 10^{-2}, 10^{-4}, and 10^{-2}.

Interestingly, the architectural hyperparameters, such as depth and width of MTLDNNs, do not contribute to MTLDNNs' prediction accuracy, suggesting that naively increasing the scale of MTLDNNs cannot improve model performance. This result can be explained via the previous discussion regarding how more complex models can lead to worse predictive performance in Section 3.3.4. While the approximation errors of deeper and wider MTLDNNs decrease, the large estimation error can counteract the approximation gain, particular when the sample size is small. While our results are different from many studies that found DNNs outperforming MNL in the travel behavioral analysis [42, 63], the finding of only limited performance improvement from deeper and wider DNN architectures was not unseen [40]. Our results imply that it cannot effectively help the model performance if researchers only naively apply the default feedforward DNN architecture

[7] Cross-entropy losses do not allow very wide comparison. For example, when the choice sets of two models have different numbers of alternatives, the two models are not directly comparable using cross-entropy losses.



(a) λ_1

(b) λ_2

(c) λ_3

(d) Depth (shared)

(e) Depth (specific)

(f) Width

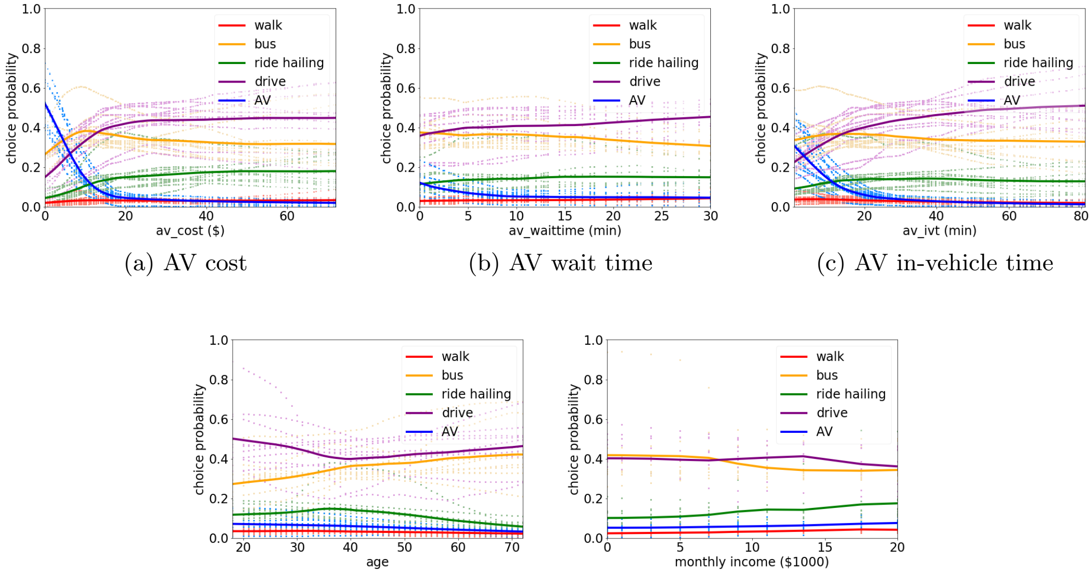
Fig. 4. Prediction accuracy with regularization and architectural hyperparameters; red and blue dots are the results of individual models; blue lines connect average prediction accuracy of all models, and red ones connect the models with the maximum prediction accuracy.

without architectural innovations adjusted to the RP-SP problem.

5.3. Interpreting Substitution Patterns

MTLDNNs are not only predictive, but also interpretable. DNNs can be interpreted in at least two ways: visualizing the substitution patterns of choice alternatives or computing the elasticity values with the gradients' information. The gradient information is commonly used for interpreting DNNs [49, 44, 9, 3, 47, 61, 60]. In Figure 5, the y-axis represents the probability of choosing AVs; the x-axis represents the change of input variables; each curve represents how the choice probability varies with input variables holding all the other variables constant at the level of sample average.

As shown in Figure 5, the probability of choosing AVs is highly sensitive to the change of AV-specific attributes, such as costs, waiting time, and in-vehicle travel time of AVs, but much less so to the socio-economic variables, such as age and income. For example, as the AV cost increases from \$0 to \$20, the probability of choosing AVs drops from about 50% to only 5%; similarly, as the AV in-vehicle travel time increases from 0 to 20 minutes, the probability drops from about 30% to only 5%. In contrast to the AV-specific variables, the probability of adopting AV is much less sensitive to socio-economic variables (Figures 5e and 5d): the probability curve of adopting AV is nearly flat everywhere. Meanwhile, Figures 5a and 5c reveal that as the cost and in-vehicle travel time of AVs increase, AVs are substituted primarily by driving and secondarily by ride hailing. This substitution pattern is intuitive, because among the five travel mode alternatives, AV is presumably more similar to driving and ride hailing than walking and taking buses.



(a) AV cost

(b) AV wait time

(c) AV in-vehicle time

(d) Age

(e) Income

Fig. 5. Choice probability functions varying with inputs values; the curves are generated by varying the targeting input variables while holding all the other variables constant; light curves are the individual MTLDNN results; dark ones are the average of top 10 models.

5.4. Interpreting Elasticity Values

Table 3 presents the elasticity values of the five travel modes with respect to the input variables in SP. The elasticity values in MTLDNNs are generated by computing the gradient information of each variable, while holding all the other variables constant. The detailed coefficient tables of the NL models are reported in Appendix VI. Panels 1 and 2 report the values in the top MTLDNN model and the NL model. To facilitate the discussion, we highlight the self-elasticity values on the main diagonal, which are the sensitivity of the choice probabilities of the alternatives regarding their own attributes.

The elasticity values from the MTLDNN model are overall reasonable in terms of their signs and magnitudes. The self-elasticity values are all negative, which are the same as those in the NL model; the majority of the cross-elasticity values in the MTLDNN are positive, which are slightly different from the completely positive values in the NL model. These signs are reasonable because higher prices of one alternative should lead to fewer people choosing this alternative and more people choosing its substitutes. As to the magnitude, economics theories suggest that the elasticity values should be around -1.0, which are similar to the value range of the self-elasticity values found in the MTLDNN model.[8] For example, with 1% increase in walking time, the probability of people choosing to walk decreases by 1.45%; with 1% increase in the cost of public transit, the

[8] When the price elasticity of demand is smaller than -1, service providers can increase revenues by reducing prices; when it is larger than -1, service providers can increase revenues by increasing prices. Therefore, the price elasticity should converge to -1 in an ideal theoretical setting, around which the service providers cannot improve their revenue by increasing or reducing prices.

Table 3: Elasticities of five travel modes with respect to input variables in SP; elasticities of the RP part are attached in Appendix V.

Panel 1: MTLDNN	Walk	Public Transit	Ride Hailing	Driving	AV
Walk time	**-1.653(1.6)**	0.240(0.4)	0.371(0.4)	0.065(0.3)	0.166(0.4)
Public transit cost	-0.202(0.7)	**-0.699(0.5)**	0.345(0.5)	0.182(0.3)	0.146(0.4)
Public transit walk time	0.000(0.5)	**-0.227(0.5)**	0.092(0.4)	0.054(0.3)	0.223(0.4)
Public transit wait time	0.039(0.3)	**-0.323(0.5)**	-0.129(0.4)	0.096(0.2)	0.028(0.4)
Public transit in-vehicle time	0.023(0.6)	**-0.518(0.6)**	0.215(0.6)	0.076(0.4)	0.055(0.6)
Ride hail cost	0.198(0.9)	0.303(0.6)	**-0.624(0.7)**	-0.034(0.4)	0.683(0.5)
Ride hail wait time	-0.029(0.6)	0.076(0.4)	**-0.644(0.7)**	-0.001(0.4)	0.126(0.5)
Ride hail in-vehicle time	0.005(0.7)	0.022(0.4)	**-0.911(0.7)**	0.096(0.3)	-0.110(0.4)
Drive cost	0.166(0.7)	0.391(0.6)	0.500(0.6)	**-0.535(0.7)**	0.496(0.5)
Drive walk time	0.335(0.5)	0.124(0.4)	0.320(0.4)	**-0.211(0.3)**	0.297(0.4)
Drive in-vehicle time	0.241(0.6)	0.560(0.6)	0.648(0.6)	**-0.659(0.8)**	0.599(0.6)
AV cost	0.034(0.4)	-0.053(0.4)	0.263(0.4)	0.192(0.4)	**-0.854(0.8)**
AV wait time	0.039(0.4)	-0.004(0.4)	0.355(0.4)	0.069(0.3)	**-0.378(0.4)**
AV in-vehicle time	-0.242(0.4)	-0.085(0.4)	0.140(0.4)	0.293(0.4)	**-0.902(0.7)**
Panel 2: NL	Walk	Public Transit	Ride Hailing	Driving	AV
Walk time	**-1.907(1.9)**	0.125(0.1)	0.125(0.1)	0.125(0.1)	0.125(0.1)
Public transit cost	0.137(0.1)	**-0.529(0.4)**	0.137(0.1)	0.137(0.1)	0.137(0.1)
Public transit access time	0.079(0.1)	**-0.287(0.3)**	0.079(0.1)	0.079(0.1)	0.079(0.1)
Public transit transfer time	0.067(0.1)	**-0.229(0.2)**	0.067(0.1)	0.067(0.1)	0.067(0.1)
Public transit in-vehicle time	0.124(0.2)	**-0.436(0.4)**	0.124(0.2)	0.124(0.2)	0.124(0.2)
Ride hail cost	0.025(0.0)	0.025(0.0)	**-0.197(0.2)**	0.025(0.0)	0.025(0.0)
Ride hail wait time	0.036(0.0)	0.036(0.0)	**-0.314(0.2)**	0.036(0.0)	0.036(0.0)
Ride hail in-vehicle time	0.077(0.1)	0.077(0.1)	**-0.680(0.5)**	0.077(0.1)	0.077(0.1)
Drive cost	0.292(0.2)	0.292(0.2)	0.292(0.2)	**-0.790(1.1)**	0.292(0.2)
Drive walk time	0.116(0.1)	0.116(0.1)	0.116(0.1)	**-0.218(0.3)**	0.116(0.1)
Drive in-vehicle time	0.263(0.2)	0.263(0.2)	0.263(0.2)	**-0.433(0.5)**	0.263(0.2)
AV cost	0.045(0.1)	0.045(0.1)	0.045(0.1)	0.045(0.1)	**-0.410(0.4)**
AV wait time	0.029(0.0)	0.029(0.0)	0.029(0.0)	0.029(0.0)	**-0.260(0.2)**
AV in-vehicle time	0.065(0.1)	0.065(0.1)	0.065(0.1)	0.065(0.1)	**-0.629(0.6)**

probability of people choosing public transit decreases by 0.63%. Although the magnitudes between MTLDNNs and NLs are different, it is difficult to evaluate which one approximates reality better because the underlying true data generating process is always unknown to researchers.

The elasticity values enable us to rank the importance of the input variables regarding the adoption of AVs, and the result is similar to that from Figure 5. Specifically, one percent increase in the AV cost and in-vehicle travel time leads to 0.854 and 0.902 percent decrease of the probability of using AVs, overall larger than the other variables' impacts on AV adoption. The results suggest that AV adoption heavily depends on its cost structure as opposed to the socio-economic information and other alternatives' attributes. While it is hard to rigorously evaluate the reliability of the economic information from the MTLDNN models, the bottomline is that it is at least feasible to extract intuitive economic information from MTLDNNs.

6. Conclusions and Discussions

This study introduces the MTLDNN framework to combine RP and SP for demand analysis. It is fueled by the practical importance of combining RP and SP for prediction and the theoretical interest of using deep learning to analyze individual demand. This study investigates the theoretical, empirical, and behavioral dimensions of tackling the RP-SP problem under the MTLDNN framework, yielding the following findings.

Theoretically, it is feasible and appealing to combine RP and SP data using the MTLDNN framework. It is because the MTLDNN framework takes advantage of the capacity of automatic feature learning in DNNs and imposes flexible constraints to capture the similarities and differences between tasks. MTLDNNs are more generic than the classical NL in combining RP and SP, owing to the approximation power and diverse architectures. Empirically however, the gain of model performance in using the MTLDNN framework is still limited. MTLDNNs outperform the NL models in terms of prediction accuracy, but underperform the NL models in terms of cross-entropy losses. MTLDNNs' performance can be mainly attributed to the regularizations specific to the multitask learning problem, but not much to the feedforward deep architectures. Behaviorally, MTLDNNs can reveal reasonable substitution patterns and elasticity values. MTLDNNs reveal that AVs mainly substitute the driving and ride hailing modes and that the AV-specific variables are more important than socio-economic variables in determining the adoption of AVs.

The study poses intriguing questions about how to evaluate, improve, and interpret the MTLDNNs. First, our results show the inconsistent model evaluation between prediction accuracy and cross-entropy losses. Since the two metrics represent two disciplinary views, currently the papers adopting the machine learning perspective tend to emphasize the prediction accuracy while those adopting the choice modeling perspective emphasize the cross-entropy losses (a.k.a. negative log likelihood). However, owing to their potential conflicts, future researchers should at least report both and then seek to reconcile any conflict between the two metrics. As machine learning permeates into the choice modeling practices, this reconciliation will become an imperative for future researchers. Second, even the prediction accuracy of MTLDNNs is only modestly higher than the NL models. Since this study uses only the simplest MTLDNN architecture, the limited empirical improvement in model performance is not unexpected. Future studies should continue the efforts of improving the MTLDNN performance by using larger sample size or advanced MTLDNN architectures, or creating MTLDNN architectures specific for the RP-SP problem. In fact, many novel MTLDNN architectures are already created to capture the similarities and differences of multiple tasks in a way subtler than the architecture used in this study [35, 22, 38, 46]. Novel MTLDNN architectures can also be created in an automatic way by using sequential modeling techniques such as the autoML tools [17, 54, 68, 69]. Lastly, this study analyzed substitution patterns and elasticity values to interpret MTLDNNs. But model interpretability is an ambiguous concept, which leads to both the challenge of creating definitive interpretation methods and the opportunity of extracting novel information that is unforeseen from classical choice modeling methods.

The MTLDNN framework can create a vast number of empirical and theoretical research opportunities. Future studies should explore the application of MTLDNNs to the classical topics in urban transportation, such as jointly analyzing auto ownership and mode choice, trip chains and travel modes, and travel time and VMT, because multitask learning appears to be a viable and intuitive framework for these situations of joint analysis. Future studies can approach the MTLDNN framework from the classical statistical perspective; for example, classical methods often model the structure of the unobserved random utility of RP and SP, which is not yet explored in the MTLDNN framework. In short, we hope that our work has connected the novel MTLDNN framework to the classical choice models, and revealed the tremendous opportunities of using the MTLDNN framework for choice modeling, behavioral analysis, and policy discussions.

Acknowledgements

We thank Singapore-MIT Alliance for Research and Technology (SMART) for partially funding this research. We thank Mary Rose Fissinger for her careful proofreading.

Contributions of Authors

S.W. conceived of the presented idea; S.W. developed the theory and reviewed previous studies; S.W. and Q.W. designed and conducted the experiments; S.W. drafted the manuscripts; Q.W. and J.Z. provided comments; J.Z. 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] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. "Multi-task feature learning". In: *Advances in neural information processing systems.* 2007, pp. 41–48.

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

[4] 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.

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

[6] Moshe Ben-Akiva and Takayuki Morikawa. "Estimation of switching models from revealed preferences and stated intentions". In: *Transportation Research Part A: General* 24.6 (1990), pp. 485–495.

[7] Moshe Ben-Akiva et al. "Combining revealed and stated preferences data". In: *Marketing Letters* 5.4 (1994), pp. 335–349.

[8] 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.

[9] 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.

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

[11] Mark A Bradley and Andrew J Daly. "Estimation of logit choice models using mixed stated preference and revealed preference information". In: *Understanding travel behaviour in an era of change* (1997), pp. 209–232.

[12] Rich Caruana. "Multitask learning". In: *Machine learning* 28.1 (1997), pp. 41–75.

[13] Jonathan D Cohen et al. *Measuring time preferences.* Tech. rep. National Bureau of Economic Research, 2016.

[14] Ronan Collobert and Jason Weston. "A unified architecture for natural language processing: Deep neural networks with multitask learning". In: *Proceedings of the 25th international conference on Machine learning.* ACM, 2008, pp. 160–167.

[15] Theodoros Evgeniou, Charles A Micchelli, and Massimiliano Pontil. "Learning multiple tasks with kernel methods". In: *Journal of Machine Learning Research* 6.Apr (2005), pp. 615–637.

[16] Manuel Fernández-Delgado et al. "Do we need hundreds of classifiers to solve real world classification problems". In: *Journal of Machine Learning Research* 15.1 (2014), pp. 3133–3181.

[17] Matthias Feurer and Frank Hutter. *Chapter 1. Hyperparameter Optimization.* 2018.

[18] Thomas F Golob. "Structural equation modeling for travel behavior research". In: *Transportation Research Part B: Methodological* 37.1 (2003), pp. 1–25.

[19] Thomas F Golob, David S Bunch, and David Brownstone. "A vehicle use forecasting model based on revealed and stated vehicle type choice and utilisation data". In: *Journal of Transport Economics and Policy* (1997), pp. 69–92.

[20] Thomas F Golob and Michael G McNally. "A model of activity participation and travel interactions between household heads". In: *Transportation Research Part B: Methodological* 31.3 (1997), pp. 177–194.

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

[22] Kazuma Hashimoto et al. "A joint many-task model: Growing a neural network for multiple nlp tasks". In: *arXiv preprint arXiv:1611.01587* (2016).

[23] Jerry Hausman. "Mismeasured variables in econometric analysis: problems from the right and problems from the left". In: *The Journal of Economic Perspectives* 15.4 (2001), pp. 57–67.

[24] Jerry A Hausman, Jason Abrevaya, and Fiona M Scott-Morton. "Misclassification of the dependent variable in a discrete-response setting". In: *Journal of Econometrics* 87.2 (1998), pp. 239–269.

[25] John Paul Helveston, Elea McDonnell Feit, and Jeremy J Michalek. "Pooling stated and revealed preference data in the presence of RP endogeneity". In: *Transportation Research Part B: Methodological* 109 (2018), pp. 70–89.

[26] David A Hensher and Mark Bradley. "Using stated response choice data to enrich revealed preference discrete choice models". In: *Marketing Letters* 4.2 (1993), pp. 139–151.

[27] Stephane Hess and John M Rose. "Should reference alternatives in pivot design SC surveys be treated differently?" In: *Environmental and Resource Economics* 42.3 (2009), pp. 297–317. ISSN: 0924-6460.

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

[29] Laurent Jacob, Jean-philippe Vert, and Francis R Bach. "Clustered multi-task learning: A convex formulation". In: *Advances in neural information processing systems.* 2009, pp. 745–752.

[30] Seyoung Kim and Eric P Xing. "Tree-guided group lasso for multi-task regression with structured sparsity". In: (2010).

[31] Ryuichi Kitamura et al. “A comparative analysis of time use data in the Netherlands and California”. In: (1992).

[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] Shiyu Liang and R Srikant. “Why deep neural networks for function approximation?” In: *arXiv preprint arXiv:1610.04161* (2016).

[35] Mingsheng Long and Jianmin Wang. “Learning multiple tasks with deep relationship networks”. In: *arXiv preprint arXiv:1506.02117* 2 (2015).

[36] Jordan J Louviere et al. “Combining sources of preference data for modeling complex decision processes”. In: *Marketing Letters* 10.3 (1999), pp. 205–217.

[37] Patricia K Lyon. “Time-dependent structural equations modeling: A methodology for analyzing the dynamic attitude-behavior relationship”. In: *Transportation Science* 18.4 (1984), pp. 395–414.

[38] Ishan Misra et al. “Cross-stitch networks for multi-task learning”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 3994–4003.

[39] Taka Morikawa, Moshe Ben-Akiva, and Daniel McFadden. “Discrete choice models incorporating revealed preferences and psychometric data”. In: *Advances in Econometrics* 16 (2002), pp. 29–56.

[40] Mikhail Mozolin, J-C Thill, and E Lynn Usery. “Trip distribution forecasting with multilayer perceptron neural networks: A critical evaluation”. In: *Transportation Research Part B: Methodological* 34.1 (2000), pp. 53–73.

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

[42] 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.

[43] Amalia Polydoropoulou and Moshe Ben-Akiva. “Combined revealed and stated preference nested logit access and mode choice model for multiple mass transit technologies”. In: *Transportation Research Record: Journal of the Transportation Research Board* 1771 (2001), pp. 38–45.

[44] 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.

[45] David Rolnick and Max Tegmark. "The power of deeper networks for expressing natural functions". In: *arXiv preprint arXiv:1705.05502* (2017).

[46] Sebastian Ruder12 et al. "Sluice networks: Learning what to share between loosely related tasks". In: *stat* 1050 (2017), p. 23.

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

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

[49] AH Sung. "Ranking importance of input parameters of neural networks". In: *Expert Systems with Applications* 15.3-4 (1998), pp. 405–411.

[50] Timothy J Tardiff. "Causal inferences involving transportation attitudes and behavior". In: *Transportation Research* 11.6 (1977), pp. 397–404.

[51] Kenneth Train. "A structured logit model of auto ownership and mode choice". In: *The Review of Economic Studies* 47.2 (1980), pp. 357–370.

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

[53] Kenneth Train and Wesley W Wilson. "Estimation on stated-preference experiments constructed from revealed-preference choices". In: *Transportation Research Part B: Methodological* 42.3 (2008), pp. 191–203.

[54] Joaquin Vanschoren. *Chapter 2. Meta-Learning*. 2018.

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

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

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

[60] Shenhao Wang, Baichuan Mo, and Jinhua Zhao. "Deep neural networks for choice analysis: Architecture design with alternative-specific utility functions". In: *Transportation Research Part C: Emerging Technologies* 112 (2020), pp. 234–251. ISSN: 0968-090X.

[61] Shenhao Wang, Qingyi Wang, and Jinhua Zhao. "Deep Neural Networks for Choice Analysis: Extracting Complete Economic Information for Interpretation". In: *arXiv preprint arXiv:1812.04528* (2018).

[62] John C Whitehead et al. “Combining revealed and stated preference data to estimate the nonmarket value of ecological services: an assessment of the state of the science”. In: *Journal of Economic Surveys* 22.5 (2008), pp. 872–908.

[63] 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.

[64] Yongxin Yang and Timothy M Hospedales. “Trace norm regularised deep multi-task learning”. In: *arXiv preprint arXiv:1606.04038* (2016).

[65] Xin Ye, Ram M Pendyala, and Giovanni Gottardi. “An exploration of the relationship between mode choice and complexity of trip chaining patterns”. In: *Transportation Research Part B: Methodological* 41.1 (2007), pp. 96–113.

[66] Ming Yuan and Yi Lin. “Model selection and estimation in regression with grouped variables”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 68.1 (2006), pp. 49–67.

[67] Christopher Zegras. “The built environment and motor vehicle ownership and use: Evidence from Santiago de Chile”. In: *Urban Studies* 47.8 (2010), pp. 1793–1817.

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

[69] Barret Zoph et al. “Learning transferable architectures for scalable image recognition”. In: *arXiv preprint arXiv:1707.07012* 2.6 (2017).

Appendix I: Descriptive Summary Statistics

The key statistics of our samples are summarized in Table A1. Even though driving is not available to some people in their revealed preference, the survey backend calculated the driving parameters for these observations nonetheless and is reflected in the table. A comparison of age and income distribution between the sample and the population is summarized in Table A2. In terms of age, the sample overrepresents young people, and underrepresents the elderly. In terms of monthly income, individuals with no income and very high income more than $20,000$ are underrepresented in the sample, although the distribution of other income groups is close to that of the population.

Table A1: Summary statistics of the Singapore dataset

Panel 1. Continuous Variables							
	mean	std	min	25%	50%	75%	max
Walk_walktime (min)	60.504	54.875	2.0	28.00	40.0	75.00	630.0
Bus_cost (S$)	2.069	1.266	0.0	1.12	1.8	2.52	7.0
Bus_walktime (min)	11.964	10.782	0.0	4.20	8.0	15.00	84.0
Bus_waittime (min)	7.732	5.033	0.0	4.00	7.0	10.00	42.0
Bus_ivt (min)	25.064	18.911	0.0	10.00	21.0	31.20	168.0
Ridesharing_cost (S$)	14.485	11.636	0.0	7.00	12.0	17.60	140.0
Ridesharing_waittime (min)	7.108	4.803	0.0	4.00	5.6	9.00	42.0
Ridesharing_ivt (min)	18.283	13.389	0.8	9.80	15.4	23.20	147.0
AV_cost (S$)	16.076	14.598	0.0	7.70	12.1	18.70	180.0
AV_waittime (min)	7.249	5.675	0.0	3.00	6.0	8.00	48.0
AV_ivt (min)	20.115	16.989	0.6	9.00	16.2	25.20	189.0
Drive_cost (S$)	10.494	10.568	0.0	3.20	7.0	16.00	70.0
Drive_walktime (min)	3.968	4.176	0.0	1.40	2.8	4.80	42.0
Drive_ivt (min)	17.430	14.101	0.8	8.00	14.4	22.40	168.0
Age (year)	41.349	12.478	18.0	31.00	41.0	50.00	82.0
Income (K S$)	9.827	5.013	0.0	7.00	9.0	13.50	20.0
Education	3.063	2.698	0.0	0.0	4.0	5.00	7.0
Panel 2. Discrete Variables (Counts)							
Gender	5,190 (1: Male); 3,228 (0: Female)						
Employment	5,064 (1: Employed); 3,354 (0: Unemployed)						

Table A2: Comparison of sample and population

Age Group	Population (%)	Sample (%)	Income Group	Population (%)	Sample (%)
20 – 24	8.42	16.31	No income	10.79	1.46
25 – 29	9.04	17.32	Below \$2,000	7.49	7.19
30 – 34	9.22	15.45	\$2,000 – \$3,999	10.69	14.9
35 – 39	9.75	14.08	\$4,000 – \$5,999	11.29	17.35
40 – 44	10.12	10.09	\$6,000 – \$7,999	10.89	15.57
45 – 49	9.72	10.2	\$8,000 – \$9,999	9.49	14.77
50 – 54	10.19	7.42	\$10,000 – \$11,999	8.39	10.07
55 – 59	9.67	4.93	\$12,000 – \$14,999	9.09	8.22
60 – 64	8.13	2.49	\$15,000 – \$19,999	9.49	4.78
65 – 69	6.39	0.67	Over \$20,000	12.39	5.69
70 – 74	3.35	0.91			
75 – 79	2.84	0			
80 – 84	1.73	0.13			
85+	1.43	0			

Appendix II: Utility Specifications of NLs

Table A3 summarizes the utility specifications of the NL-C and NL-NC models. The utility functions are presented as a table rather than functions because the mathematical formula can be unnecessarily complicated in presentation, which include about 20 equations (5 alternatives * 2 (RP and SP) * 2 (NL-C and NL-NC)). For both NL-C and NL-NC models, the utility specifications follow a linear structure:

$$U_{k_r,i} = V_{k_r,i} + \epsilon_{k_r} = w_{k_{sh}}^T x_{sh,i} + w_{k_r}^T x_{r,i} + \epsilon_{k_r,i} \tag{15}$$

$$U_{k_s,t} = V_{k_s,t} + \epsilon_{k_s} = w_{k_{sh}}^T x_{sh,t} + w_{k_s}^T x_{s,t} + \epsilon_{k_s,t} \tag{16}$$

In Table A3, the first column presents the coefficients, in which ASC implies alternative specific constant. The second column indicates whether the coefficient exists for the specific alternatives. As walking is used as the reference alternative, the demographic variables are included in all the utility functions except for walking. In the third and fourth columns, 'SP' implies that the coefficient only exists for SP's utility functions, corresponding to w_{k_s} in the equations above; 'SP,RP' implies that the coefficient enters the utility functions of both RP and SP without parameter sharing; 'SH' implies that the coefficient enters the utility functions of both RP and SP and their coefficients are constrained to be shared, corresponding to $w_{k_{sh}}$ in the equations above. As noted before, the NL-C has many shared parameters as constraints while the NL-NC model does not include parameter sharing. Since AV only exists for SP, the attributes of AV do not appear in RP specifications.

Table A3: Utility specifications of NL-C and NL-NC

Coefficient	Alternative(s)	NL-C	NL-NC
ASC_walk	Walk	SP	SP
Walk_walktime (min)	Walk	SH	SP,RP
ASC_public_transit	Public Transit	SP,RP	SP,RP
Bus_cost (S$)	Public Transit	SH	SP,RP
Bus_walktime (min)	Public Transit	SH	SP,RP
Bus_waittime (min)	Public Transit	SH	SP,RP
Bus_ivt (min)	Public Transit	SH	SP,RP
ASC_Ridehail	Ride hail	SP,RP	SP,RP
Ridesharing_cost (S$)	Ride hail	SH	SP,RP
Ridesharing_waittime (min)	Ride hail	SH	SP,RP
Ridesharing_ivt (min)	Ride hail	SH	SP,RP
ASC_AV	AV	SP	SP
AV_cost (S$)	AV	SP	SP
AV_waittime (min)	AV	SP	SP
AV_ivt (min)	AV	SP	SP
ASC_drive	Drive	SP,RP	SP,RP
Drive_cost (S$)	Drive	SH	SP,RP
Drive_walktime (min)	Drive	SH	SP,RP
Drive_ivt (min)	Drive	SH	SP,RP
Age (year)	All except walk	SH	SP,RP
Income (K S$)	All except walk	SH	SP,RP
Education	All except walk	SH	SP,RP
License	All except walk	SH	SP,RP
Auto Ownership	All except walk	SH	SP,RP

Appendix III: Hyperparameter Space

Table A4: Hyperparameter space of MTLDNNs

Hyperparameter Dimensions	**Values**
Shared M1	$[1, 2, 3, 4, 5]$
Domain-specific M2	$[1, 2, 3, 4, 5]$
λ_1 constant	$[10^{-20}, 10^{-4}, 10^{-2}, 0.5]$
λ_2 constant	$[10^{-20}, 10^{-4}, 10^{-2}, 0.5]$
λ_3 constant	$[10^{-20}, 10^{-4}, 10^{-2}, 0.5]$
n hidden	$[25, 50, 100, 200]$
n iteration	20000
n mini batch	200

Appendix IV: Top 10 MTLDNN Architectures

It appears that naively increasing depth and width cannot improve the predictive power of the MTLDNN models, as shown in Table A5. However, the wise choice of regularization hyperparameters helps to improve model performance.

Table A5: Top 10 MTLDNN Architectures

Shared M1	Domain-specific M2	n hidden	λ_1	λ_2	λ_3
1	1	25	10^{-2}	10^{-2}	10^{-4}
3	2	25	10^{-2}	10^{-4}	10^{-20}
1	1	25	10^{-20}	10^{-2}	10^{-2}
1	1	25	10^{-2}	10^{-1}	10^{-4}
1	1	100	10^{-2}	10^{-20}	10^{-4}
1	4	25	10^{-2}	0.5	10^{-2}
1	1	200	10^{-2}	0.5	10^{-2}
1	1	50	10^{-2}	10^{-2}	10^{-20}
3	1	100	10^{-2}	10^{-4}	10^{-4}
2	3	50	10^{-2}	0.5	10^{-20}

Appendix V: Elasticities in RP

Table A6: Elasticities of travel modes with respect to input variables (RP)

Panel 1: MTLDNN Model	Walk	Public Transit	Ride Hailing	Driving
Walk time	**-1.235(1.3)**	0.384(0.4)	0.868(0.6)	0.094(0.4)
Public transit cost	-0.108(0.7)	**-0.198(0.3)**	0.939(0.7)	0.378(0.4)
Public transit walk time	-0.100(0.4)	**0.014(0.2)**	0.098(0.2)	0.065(0.2)
Public transit wait time	0.022(0.3)	**-0.042(0.2)**	0.181(0.4)	0.237(0.3)
Public transit in-vehicle time	-0.066(0.4)	**-0.020(0.2)**	0.059(0.4)	-0.152(0.3)
Ride hail cost	-0.151(0.7)	0.190(0.3)	**-0.369(0.4)**	-0.031(0.5)
Ride hail wait time	-0.274(0.4)	0.148(0.2)	**-0.913(0.5)**	-0.271(0.3)
Ride hail in-vehicle time	0.474(0.5)	-0.106(0.2)	**-0.469(0.5)**	0.052(0.2)
Drive cost	-0.117(0.5)	0.034(0.2)	0.102(0.4)	**-0.436(0.4)**
Drive walk time	0.103(0.3)	-0.064(0.1)	0.213(0.4)	**-0.123(0.3)**
Drive in-vehicle time	-0.514(0.5)	0.155(0.2)	0.210(0.3)	**-0.754(0.8)**
Panel 2: NL Model	Walk	Public Transit	Ride Hailing	Driving
Walk time	**-0.533(0.4)**	0.219(0.2)	0.219(0.2)	0.219(0.2)
Public transit cost	0.159(0.1)	**-0.098(0.1)**	0.159(0.1)	0.159(0.1)
Public transit access time	0.091(0.1)	**-0.056(0.1)**	0.091(0.1)	0.091(0.1)
Public transit transfer time	0.075(0.0)	**-0.046(0.0)**	0.075(0.0)	0.075(0.0)
Public transit in-vehicle time	0.161(0.1)	**-0.100(0.1)**	0.161(0.1)	0.161(0.1)
Ride hail cost	0.007(0.0)	0.007(0.0)	**-0.085(0.1)**	0.007(0.0)
Ride hail wait time	0.011(0.0)	0.011(0.0)	**-0.133(0.1)**	0.011(0.0)
Ride hail in-vehicle time	0.025(0.0)	0.025(0.0)	**-0.292(0.2)**	0.025(0.0)
Drive cost	0.007(0.0)	0.007(0.0)	0.007(0.0)	**-0.649(0.4)**
Drive walk time	0.001(0.0)	0.001(0.0)	0.001(0.0)	**-0.121(0.1)**
Drive in-vehicle time	0.003(0.0)	0.003(0.0)	0.003(0.0)	**-0.244(0.2)**

Appendix VI: Estimation Results of the Nested Logit Models

Table A7: Performance of the nested logit models

	Constrained	Unconstrained
Null Loglikelihood	-14247.66	-14247.66
Final Loglikelihood	-9669.246	-9603.553
Number of parameters	78	122
Rho square	0.321	0.326

Table A8: Coefficients of the nested logit model with parameter constrains

Variable Name	Value	Std Err	t-test	p-value
ASC_RP_BUS	-2.06	13198.443	-0.000	1.000
ASC_RP_DRIVE	-4.38	1.468	-2.986	0.003
ASC_RP_RS	-3.31	1.387	-2.383	0.017
ASC_SP_AV	4.13	1.931	2.136	0.033
ASC_SP_BUS	5.29	13198.442	0.000	1.000
ASC_SP_DRIVE	6.90	2.742	2.517	0.012
ASC_SP_RS	5.67	2.861	1.982	0.047
ASC_SP_WALK	4.04	2.097	1.925	0.054
B_AGE_AV	-0.27	0.147	-1.800	0.072
B_AGE_BUS	-0.13	0.139	-0.900	0.368
B_AGE_DRIVE	-0.20	0.143	-1.406	0.160
B_AGE_RS	-0.31	0.147	-2.136	0.033
B_AGE_WALK	-0.26	0.148	-1.785	0.074
B_AUTOOWN_AV	-0.05	0.075	-0.721	0.471
B_AUTOOWN_BUS	-0.12	0.068	-1.728	0.084
B_AUTOOWN_DRIVE	-0.02	0.072	-0.246	0.805
B_AUTOOWN_RS	0.01	0.074	0.074	0.941
B_AUTOOWN_WALK	-0.11	0.076	-1.416	0.157
B_COST_AV	-0.19	0.036	-5.273	0.000
B_COST_BUS	-0.19	0.032	-5.947	0.000
B_COST_DRIVE	-0.50	0.077	-6.590	0.000
B_COST_RS	-0.08	0.023	-3.515	0.000
B_EDU_AV	0.90	0.271	3.335	0.001
B_EDU_BUS	0.60	0.253	2.361	0.018
B_EDU_DRIVE	0.79	0.263	3.012	0.003
B_EDU_RS	0.73	0.265	2.763	0.006
B_EDU_WALK	0.71	0.268	2.635	0.008
B_FULLJOB_AV	-0.36	0.213	-1.700	0.089
B_FULLJOB_BUS	-0.17	0.199	-0.840	0.401
B_FULLJOB_DRIVE	-0.40	0.207	-1.926	0.054
B_FULLJOB_RS	-0.24	0.209	-1.132	0.258
B_FULLJOB_WALK	-0.38	0.214	-1.767	0.077
B_HEDU_AV	-0.19	0.228	-0.851	0.395
B_HEDU_BUS	-0.14	0.216	-0.649	0.517
B_HEDU_DRIVE	-0.18	0.222	-0.810	0.418
B_HEDU_RS	-0.08	0.227	-0.372	0.710
B_HEDU_WALK	-0.19	0.230	-0.839	0.401
B_INC_AV	0.11	0.073	1.479	0.139
B_INC_BUS	0.04	0.070	0.511	0.609
B_INC_DRIVE	0.09	0.071	1.255	0.209
B_INC_RS	0.09	0.072	1.193	0.233
B_INC_WALK	0.06	0.073	0.858	0.391
B_IVT_AV	-0.27	0.047	-5.829	0.000
B_IVT_BUS	-0.20	0.031	-6.224	0.000
B_IVT_DRIVE	-0.26	0.041	-6.375	0.000
B_IVT_RS	-0.26	0.044	-5.865	0.000
B_LEDU_AV	1.09	0.436	2.491	0.013
B_LEDU_BUS	0.57	0.408	1.396	0.163
B_LEDU_DRIVE	0.68	0.421	1.612	0.107
B_LEDU_RS	0.96	0.431	2.238	0.025
B_LEDU_WALK	0.88	0.434	2.036	0.042
B_LICENSE_AV	4.13	2.159	1.911	0.056
B_LICENSE_BUS	3.23	13198.443	0.000	1.000
B_LICENSE_DRIVE	2.52	1.482	1.700	0.089
B_LICENSE_RS	2.37	1.392	1.700	0.089
B_LICENSE_WALK	4.04	2.099	1.923	0.055
B_MALE_AV	-0.48	0.159	-3.003	0.003
B_MALE_BUS	-0.53	0.151	-3.518	0.000
B_MALE_DRIVE	-0.49	0.155	-3.143	0.002
B_MALE_RS	-0.51	0.158	-3.234	0.001
B_MALE_WALK	-0.43	0.161	-2.697	0.007
B_OLD_AV	0.38	0.381	0.986	0.324
B_OLD_BUS	0.68	0.359	1.902	0.057
B_OLD_DRIVE	0.70	0.367	1.902	0.057
B_OLD_RS	0.72	0.374	1.933	0.053
B_OLD_WALK	0.80	0.376	2.141	0.032
B_WAITTIME_AV	-0.11	0.026	-4.011	0.000
B_WAITTIME_BUS	-0.09	0.020	-4.477	0.000
B_WAITTIME_RS	-0.11	0.026	-4.183	0.000
B_WALKTIME_BUS	-0.15	0.026	-5.872	0.000
B_WALKTIME_DRIVE	-0.16	0.028	-5.853	0.000
B_WALKTIME_WALK	-0.86	0.133	-6.450	0.000
B_YOUNG_AV	-0.67	0.249	-2.689	0.007
B_YOUNG_BUS	-0.40	0.234	-1.725	0.084
B_YOUNG_DRIVE	-0.72	0.245	-2.929	0.003
B_YOUNG_RS	-0.62	0.245	-2.548	0.011
B_YOUNG_WALK	-0.66	0.250	-2.660	0.008
MU_SP	2.15	0.322	6.679	0.000

Table A9: Coefficients of the nested logit model without parameter constrains

Variable Name	Value	Std Err	t-test	p-value
ASC_RP_BUS	0.73	0.119	6.096	0.000
ASC_RP_DRIVE	-1.24	0.636	-1.951	0.051
ASC_RP_RS	-0.58	0.221	-2.631	0.009
ASC_SP_AV	0.73	11.285	0.064	0.949
ASC_SP_BUS	1.51	11.135	0.136	0.892
ASC_SP_DRIVE	4.39	11.309	0.388	0.698
ASC_SP_RS	0.11	11.457	0.010	0.992
ASC_SP_WALK	0.24	11.416	0.021	0.983
B_RP_AGE_BUS	-0.09	0.146	-0.590	0.555
B_RP_AGE_DRIVE	-0.33	0.661	-0.494	0.621
B_RP_AGE_RS	-0.17	0.267	-0.653	0.514
B_RP_AUTOOWN_BUS	-0.16	0.072	-2.228	0.026
B_RP_AUTOOWN_DRIVE	0.21	0.379	0.548	0.584
B_RP_AUTOOWN_RS	0.04	0.120	0.373	0.709
B_RP_COST_BUS	-0.09	0.114	-0.778	0.436
B_RP_COST_DRIVE	-0.23	0.324	-0.720	0.472
B_RP_COST_RS	0.13	0.120	1.118	0.264
B_RP_EDU_BUS	0.64	0.262	2.432	0.015
B_RP_EDU_DRIVE	1.49	1.014	1.473	0.141
B_RP_EDU_RS	0.51	0.487	1.050	0.294
B_RP_FULLJOB_RP_BUS	-0.29	0.206	-1.413	0.158
B_RP_FULLJOB_RP_DRIVE	-0.62	0.813	-0.760	0.447
B_RP_FULLJOB_RP_RS	0.43	0.428	1.013	0.311
B_RP_HEDU_BUS	-0.11	0.227	-0.486	0.627
B_RP_HEDU_DRIVE	-0.04	1.286	-0.030	0.976
B_RP_HEDU_RS	-0.03	0.412	-0.068	0.946
B_RP_INC_BUS	0.03	0.074	0.401	0.688
B_RP_INC_DRIVE	0.76	0.304	2.491	0.013
B_RP_INC_RS	0.11	0.126	0.875	0.381
B_RP_IVT_BUS	0.08	0.083	0.966	0.334
B_RP_IVT_DRIVE	-0.73	0.568	-1.286	0.198
B_RP_IVT_RS	0.08	0.152	0.548	0.584
B_RP_LEDU_BUS	0.54	0.423	1.288	0.198
B_RP_LEDU_DRIVE	2.86	1.853	1.546	0.122
B_RP_LEDU_RS	0.89	0.764	1.162	0.245
B_RP_LICENSE_BUS	0.73	0.119	6.096	0.000
B_RP_LICENSE_DRIVE	-1.24	0.636	-1.951	0.051
B_RP_LICENSE_RS	-0.58	0.221	-2.631	0.009
B_RP_MALE_BUS	-0.48	0.159	-3.020	0.003
B_RP_MALE_DRIVE	-1.58	0.835	-1.895	0.058
B_RP_MALE_RS	-1.06	0.286	-3.690	0.000
B_RP_OLD_BUS	0.59	0.368	1.607	0.108
B_RP_OLD_DRIVE	1.23	1.357	0.906	0.365
B_RP_OLD_RS	-0.26	0.877	-0.302	0.763
B_RP_WAITTIME_BUS	-0.01	0.086	-0.103	0.918
B_RP_WAITTIME_RS	-0.13	0.140	-0.908	0.364
B_RP_WALKTIME_BUS	0.14	0.081	1.687	0.092
B_RP_WALKTIME_DRIVE	-1.64	0.866	-1.889	0.059
B_RP_WALKTIME_WALK	-1.29	0.160	-8.042	0.000
B_RP_YOUNG_BUS	-0.38	0.247	-1.534	0.125
B_RP_YOUNG_DRIVE	-1.85	1.238	-1.495	0.135
B_RP_YOUNG_RS	-0.17	0.426	-0.411	0.681
B_SP_AGE_AV	0.27	11.717	0.023	0.982
B_SP_AGE_BUS	1.14	11.721	0.097	0.923
B_SP_AGE_DRIVE	0.67	11.706	0.057	0.954
B_SP_AGE_RS	-0.06	11.741	-0.005	0.996
B_SP_AGE_WALK	0.31	11.715	0.026	0.979
B_SP_AUTOOWN_AV	1.04	17.605	0.059	0.953
B_SP_AUTOOWN_BUS	0.68	17.598	0.039	0.969
B_SP_AUTOOWN_DRIVE	1.26	17.614	0.072	0.943
B_SP_AUTOOWN_RS	1.37	17.621	0.078	0.938
B_SP_AUTOOWN_WALK	0.69	17.599	0.039	0.969
B_SP_COST_AV	-1.20	1.483	-0.807	0.420
B_SP_COST_BUS	-1.21	1.499	-0.809	0.418
B_SP_COST_DRIVE	-3.16	3.902	-0.811	0.418
B_SP_COST_RS	-0.55	0.690	-0.797	0.425
B_SP_EDU_AV	0.72	17.489	0.041	0.967
B_SP_EDU_BUS	-1.24	17.936	-0.069	0.945
B_SP_EDU_DRIVE	0.01	17.613	0.000	1.000
B_SP_EDU_RS	-0.29	17.681	-0.016	0.987
B_SP_EDU_WALK	-0.49	17.729	-0.028	0.978
B_SP_FULLJOB_SP_AV	0.35	16.961	0.021	0.984
B_SP_FULLJOB_SP_BUS	1.65	17.011	0.097	0.923
B_SP_FULLJOB_SP_DRIVE	0.12	16.965	0.007	0.995
B_SP_FULLJOB_SP_RS	0.97	16.968	0.057	0.954
B_SP_FULLJOB_SP_WALK	0.23	16.965	0.014	0.989

Table A10: Coefficients of the nested logit model without parameter constrains (continued)

Variable Name	Value	Std Err	t-test	p-value
B_SP_HEDU_AV	0.29	22.537	0.013	0.990
B_SP_HEDU_BUS	0.63	22.582	0.028	0.978
B_SP_HEDU_DRIVE	0.39	22.547	0.017	0.986
B_SP_HEDU_RS	0.98	22.639	0.043	0.966
B_SP_HEDU_WALK	0.32	22.542	0.014	0.989
B_SP_INC_AV	1.20	6.436	0.186	0.852
B_SP_INC_BUS	0.76	6.428	0.119	0.905
B_SP_INC_DRIVE	1.08	6.429	0.167	0.867
B_SP_INC_RS	1.06	6.429	0.164	0.870
B_SP_INC_WALK	0.91	6.426	0.141	0.888
B_SP_IVT_AV	-1.72	2.129	-0.809	0.419
B_SP_IVT_BUS	-1.32	1.632	-0.809	0.418
B_SP_IVT_DRIVE	-1.64	2.024	-0.810	0.418
B_SP_IVT_RS	-1.66	2.047	-0.809	0.418
B_SP_LEDU_AV	2.18	29.082	0.075	0.940
B_SP_LEDU_BUS	-1.10	29.278	-0.038	0.970
B_SP_LEDU_DRIVE	-0.42	29.189	-0.014	0.989
B_SP_LEDU_RS	1.42	29.079	0.049	0.961
B_SP_LEDU_WALK	0.89	29.096	0.030	0.976
B_SP_LICENSE_AV	0.73	11.285	0.064	0.949
B_SP_LICENSE_BUS	1.51	11.135	0.136	0.892
B_SP_LICENSE_DRIVE	4.39	11.309	0.388	0.698
B_SP_LICENSE_RS	0.11	11.457	0.010	0.992
B_SP_LICENSE_WALK	0.24	11.416	0.021	0.983
B_SP_MALE_AV	0.41	21.286	0.019	0.985
B_SP_MALE_BUS	0.01	21.276	0.000	1.000
B_SP_MALE_DRIVE	0.35	21.283	0.017	0.987
B_SP_MALE_RS	0.29	21.282	0.014	0.989
B_SP_MALE_WALK	0.67	21.299	0.032	0.975
B_SP_OLD_AV	-1.65	40.331	-0.041	0.967
B_SP_OLD_BUS	0.28	40.054	0.007	0.994
B_SP_OLD_DRIVE	0.36	40.045	0.009	0.993
B_SP_OLD_RS	0.63	40.022	0.016	0.988
B_SP_OLD_WALK	1.02	39.989	0.025	0.980
B_SP_WAITTIME_AV	-0.67	0.831	-0.801	0.423
B_SP_WAITTIME_BUS	-0.57	0.711	-0.803	0.422
B_SP_WAITTIME_RS	-0.68	0.848	-0.801	0.423
B_SP_WALKTIME_BUS	-1.01	1.250	-0.809	0.419
B_SP_WALKTIME_DRIVE	-1.02	1.260	-0.809	0.418
B_SP_WALKTIME_WALK	-4.97	6.127	-0.811	0.418
B_SP_YOUNG_AV	-0.01	20.331	-0.000	1.000
B_SP_YOUNG_BUS	1.65	20.204	0.082	0.935
B_SP_YOUNG_DRIVE	-0.31	20.375	-0.015	0.988
B_SP_YOUNG_RS	0.18	20.307	0.009	0.993
B_SP_YOUNG_WALK	0.07	20.322	0.003	0.997
MU_SP	0.34	0.425	0.811	0.418