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. This study presents a framework of multitask learning deep neural networks (MTLDNNs) to jointly analyze RP and SP, demonstrating its theoretical flexibility and empirical benefits in terms of prediction and economic interpretation. Theoretically, the MTLDNN framework is more generic than the classical nested logit (NL) method because of MTLDNNs' capacity of automatic feature learning and flexible regularization. Empirically, MTLDNNs outperform six benchmark models and particularly the classical NL models by about 5% prediction accuracy in analyzing the adoption of autonomous vehicles (AVs) based on a survey data collected in Singapore. This improvement can be mainly attributed to the soft constraints specific to multitask learning, including its innovative architectural design and regularization methods, but not much to the generic capacity of automatic feature learning endowed by a standard feedforward DNN architecture. Besides prediction, MTLDNNs are also interpretable. MTLDNNs reveal that AVs mainly substitute driving and that the variables specific to AVs are more important than the socio-economic variables in determining AV adoption. Overall, this study presents a new MTLDNN framework in combining RP and SP, demonstrates its theoretical flexibility of architectural design and regularizations, shows its empirical predictive power, and extracts reasonable economic information for interpretation. Future studies can explore other MTLDNN architectures, new choice modeling applications for multitask learning, and deeper theoretical connections between the MTLDNN framework and structural choice modeling.

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. RP and SP are often combined because RP and SP data have their own pros and cons: 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. The common way to combine RP and SP is to use a nested logit (NL) approach, which assigns the alternatives in RP and SP to two nests with different scale factors [29, 11, 6, 7] [1]. However, this NL method heavily relies on handcrafted feature engineering, which can be overly restrictive compared to the underlying complex data generating process. This handcrafted feature engineering is in sharp contrast to the mechanism in deep neural networks (DNNs) [35, 8, 15], which can automatically learn generalizable feature structures. The extraordinary power of DNNs, as demonstrated across domains [17, 34, 35], prompts us to ask whether it is possible to address the classical problem of combining RP and SP for demand analysis in a DNN framework, in a way more generic and flexible than the traditional NL method.

This paper presents a framework of multitask learning deep neural networks (MTLDNNs) to jointly analyze RP and SP, demonstrating its theoretical flexibility compared to the NL method and its empirical power in prediction and interpretation as revealed by an experiment. Figure 1 visualizes a MTLDNN architecture, which starts with shared layers and ends with task-specific layers, capturing both the similarities and differences between the tasks of RP and SP [13]. Our study firstly compares the MTLDNN framework to the classical NL method, demonstrating that MTLDNNs are more generic than NL owing to their automatic feature learning and soft constraints. We further apply this MTLDNN framework to a data set collected in Singapore, which was designed to analyze the adoption of autonomous vehicles (AVs) by adding the travel mode of AVs to four revealed travel modes in a SP survey. The empirical experiments compare two MTLDNN models to six benchmark models, including the NL models with and without parameter constraints (NL-C and NL-NC), the multinomial logit models that are separately or jointly trained for RP and SP (MNL-SPT and MNL-JOINT), and the DNN models that are separately or jointly trained for RP and SP (DNN-SPT and DNN-JOINT). To understand the AV adoption from the MTLDNNs, we visualize the relationship between choice probability functions and input variables and compute the elasticities by using MTLDNNs' gradients information [3, 50]. This process of interpreting MTLDNNs demonstrates that MTLDNNs can not only predict accurately, but also provide key insights similar to the classical NL method.

As far as we know, this is the first study that presents the MTLDNN framework to the community of choice modeling. Our work reveals that the MTLDNN framework is closely related to but also more generic than the classical NL approach in combining RP and SP, thus being able to solve not only the same problems that NL can traditionally solve, such as combining data sources to reduce errors, but also to partially address the problems that implicitly exist in the NL model,

[1] This nested logit method can also be seen as a pooled estimation with heteroscedasticity across RP and SP [38, 28]

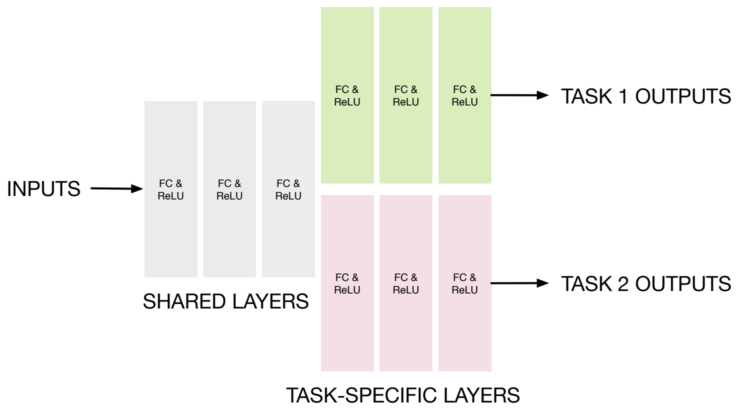


Fig. 1. MTLDNN architecture; this architecture has 3 shared and 3 task-specific layers, but they represent generally M_1 shared and M_2 task-specific layers; FC stands for fully connected layers; ReLU for Rectified Linear Units. This study uses RP as Task 1 and SP as Task 2.

such as function misspecification caused by the linear-in-parameter assumption. Given the power in our simple MTLDNN framework (Figure 1), future studies can improve the predictive power and interpretable information by exploring more advanced MTLDNN architectures [37, 24, 40, 49]. Future studies can also use the MTLDNN framework to explore other innovative applications, such as jointly analyzing car ownership and travel mode choice [56, 71], activity patterns and trip chain choices [33, 21], and many other applications that are traditionally analyzed by structural equation models (SEM). The similarity of the visual structure between MTLDNNs and NL models prompts us to ask whether MTLDNNs can be seen as a generic extension of NL, which we cannot answer here but believe as an interesting question for future studies. To enable future researchers to follow our work, we have uploaded the project to Github https://github.com/cjsyzwsh/Multitask-learning-deep-neural-networks-to-combine-revealed-and-stated-preference-data.git [2].

This paper is organized as following. Section 2 reviews the MTLDNN models and the traditional methods of combining RP and SP. Section 3 presents the MTLDNN model and discusses why it is more generic than the NL method. Section 4 presents the setup of experiments, and Section 5 analyzes model performance, sources of the prediction gain of MTLDNNs, and the economic information in MTLDNNs. Section 6 concludes our findings and discuss future research directions.

2. Literature Review

RP and SP data are both necessary for travel demand analysis, but they suffer from different sources of problems. The RP data can be problematic due to its 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 respondents' sensitivity to survey formats, unrealistic hypothetical scenarios [52], dy-

[2]Unfortunately, we can only upload the codes and paper but without the data set, because of the policy limitation on the data set we collected

namics between RP and reported attitudes [51], or even just measurement errors that happen in the data collection process [26, 25], although SP is the only possible data source to analyze the effects of new pricing strategies, new public transit services, or new travel modes [6, 45].

To address these problems, one common remedy is to jointly estimate RP and SP, with the benefits of gaining efficiency and correcting biases [7]. Specifically, researchers often use the NL model to combine RP and SP by treating RP and SP as two nests of choices [29, 11, 45, 41]. For example, Polydoropoulou and Ben-Akiva (2001) used the NL approach to analyze the travel mode choice for multiple mass transit technologies [45]. Golob et al. (1997) [20] used the same method to examine how vehicle usage depends on the factors of vehicles and fuel types. In the studies with the NL method, researchers need to make parametric assumptions to capture the differences and similarities between RP and SP [11]. To capture the similarities, RP and SP choice models can be designed to share parameters; for instance, price and time coefficients in RP and SP can be specified to be the same [45]. To capture the differences, RP and SP models can be specified to have different randomness in their error terms, causing the different magnitudes of the coefficients [29, 11]. While details of the modeling specifics vary with studies, these assumptions along with the NL approach developed in the 1990s have become one standard way of combining RP and SP in choice modeling [52, 38, 66, 57].

From a machine learning perspective, combining RP and SP can be addressed by using the MTLDNN framework because RP and SP can be treated as two different but related tasks. While rarely seen in the community of choice modeling, this MTLDNN framework has shown a great success in many other fields. In natural language processing, researchers used the MTLDNN framework to jointly learn different levels of semantic components [15, 24]. In image recognition, researchers designed novel MTLDNN architectures by using Bayesian priors and cross-stitch hyper-parameters [37, 40]. In biology, Ramsundar et al. (2015) [46] used MTLDNNs to model a large dataset with 40M features and 200 biological targets, finding that MTLDNNs can significantly outperform single-task DNNs. MTLDNNs can leverage multiple sources of domain-specific information, thus successfully generalizing the results to other contexts [13].

A large number of MTLDNN architectures have been created in the past three decades. Specifically, Caruana (1997) [13] first created a benchmark MTLDNN architecture, which starts with shared layers and ends with task-specific layers, as introduced in this study (Figure 1). This MTLDNN architecture has been applied to natural language processing tasks, showing the state-of-the-art performance in the absence of handcrafted feature engineering [15]. Caruana's initial MTLDNN architecture has been later improved by new MTLDNN architectural designs, such as deep relationship network (DRN) [37], hierarchical MTLDNN [24], cross-stitch network [40], and SLUICE network [49]. Similar to the modeling concerns in the NL method, all the MTLDNN studies focus on how to control the similarities and differences of the multiple tasks, typically by designing specific MTLDNN architectures or regularization methods. In Caruana's framework, the MTLDNN model uses the first several layers to reflect the similarities and the following task-specific layers to capture the differences [13]. Studies after Caruana's work explicitly designed MTLDNN architectures to capture both the similarities and dissimilarities between tasks [40, 49]. In addition to architectural design, researchers also used various regularizations to control the parameter differences between tasks, such as using L_1/L_p group LASSO [2, 70], mean and variance regularization [16, 31], trace norm regularization [68], tree-guided regularization [32], or Bayesian tensor normal priors [37]. While sharing some similarity with the parameter constraints in the classical NL method, these constraints are more generic and flexible in describing the relationship between tasks, as discussed in the following sections.

The MTLDNN framework is quite similar to the models with simultaneous estimation, thus potentially enabling researchers to approach all the classical joint estimation questions under this new MTLDNN framework. Traditionally, researchers used simultaneous estimation to jointly analyze auto ownership and mode choice [56, 71], trip chains and travel modes [69], travel time and vehicle miles traveled (VMT) [19], travel demand and attitudinal factors [39, 41, 55], and activity patterns and travel demands [33, 21]. The MTLDNN framework can be potentially applied to all the cases. Interestingly, while the MTLDNN framework has been developed for decades in the ML community, it is relatively less known in the community of choice modeling.

3. Theory

This section introduces the MTLDNN model and the NL method, discusses the relationship between MTLDNNs, DNNs, NL, and MNL models, and briefly discusses the potential weakness of MTLDNNs by using statistical learning theory. We found that, although MTLDNNs may theoretically lead to small increases of prediction errors, MTLDNNs are overall more generic than the NL method in combining RP and SP, because the MTLDNN framework has soft constraints and the capacity of automatic feature learning, enabled by its flexible architecture design.

3.1. Multitask Learning Deep Neural Network for RP and SP

Let $x_{r,i}$ and $x_{s,t}$ denote the input variables for RP and SP; r and s stand for RP and SP, $i \in \{1, 2, ..., N_r\}$ and $t \in \{1, 2, ..., N_s\}$ are the index of RP and SP observations. $x_{r,i}, x_{s,t} \in R^d$, in which 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 zero or one 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 by the following:

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

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

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. As a result, 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 the T factor, which is referred to as temperature in the DNN literature to adjust the scale of logits [30], and T is trained in the MTLDNN model.

With choice probabilities formulated, we train the model by minimizing the empirical risk (ERM) with regularization terms:

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

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

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

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 simply the negative values of the log likelihood scores in the classical maximum likelihood estimation. 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 explicit regularizations. In total, 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 $\lambda_0 = 1$ [3]. 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 [60]. 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 similarities shared by the coefficients in RP and SP models. 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. This specific ERM formulation and the regularizations in Equation 5 are commonly used in MTLDNN studies [16, 31].

3.2. Nested Logit Model for RP and SP

The NL method comes from the past studies [29, 11, 45, 41]. The utility functions in RP and SP are assumed to be

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

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

in which w_{k_r} and w_{k_s} are the parameters for RP and SP; ϕ denotes the feature transformation based on domain-specific knowledge; 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 one scale factor:

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

[3]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.

The choice probability functions are:

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

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

Here w_r and w_s represent all 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. As a result, 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 "artificial nested logit" model, the details of which are available in [29, 11].

3.3. Relationship between MTLDNN, DNN, NL, and MNL

To understand the relationship between MTLDNN and NL, we added DNN and MNL to form Table 1, summarizing the differences of MTLDNN, DNN, NL, and MNL in terms of constraints and feature learning. This table has an implicitly symmetric structure because the correspondence between MTLDNNs and NL is similar to that between DNN and MNL. As a high-level summary, the MTLDNN framework has the soft constraints and the capacity of automatic feature learning, more generic than the hard constraints and handcrafted feature learning in the NL models.

Table 1: Four models and their differences

Models	**Constraints**	**Feature learning**
MTLDNN	Soft	Automatic
NL	Hard	Handcrafted
DNN	No	Automatic
MNL	No	Handcrafted

3.3.1. Constraints

First, MTLDNNs use soft constraints to capture the similarity between the tasks of RP and SP in the training of ERM, such as the $\lambda_3 ||w_r - w_s||^2$ in Equation 5. The soft constraints are more generic than the hard constraints in NL and no constraints in DNN and MNL [4], because both hard and no constraints are the boundary cases of soft constraints. To facilitate the understanding, we design a proposition as following.

Proposition 1. *The regularized ERM in MTLDNN can be simplified as:*

$$\min_{w_r, w_s} \hat{L}_R(w_r) + \hat{L}_S(w_s) + \lambda ||w_r - w_s||^2 \tag{13}$$

When $\hat{L}_R(w_r)$ and $\hat{L}_S(w_s)$ are convex, there exists a corresponding value c, such that the following constrained ERM problem has the same optimum values w_r^* and w_s^* as the regularized ERM in Equation 13:

$$\min_{w_r, w_s} \hat{L}_R(w_r) + \hat{L}_S(w_s); \quad s.t. \ ||w_r - w_s||^2 \leq c \tag{14}$$

In addition, λ is inversely proportional to the value of c.

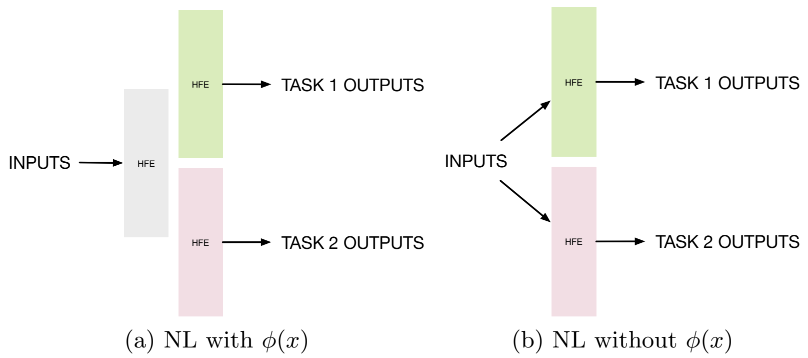
Remarks.

1. This proposition connects the regularized ERM to the constrained ERM. While the regularized ERM in Equation 13 might be seen as a data-driven "trick", the constrained ERM in Equation 14 can be understood in a more intuitive way: c as the constraint captures the distance between the tasks of RP and SP.
2. Intuitively, large λ in the regularized ERM implies a strong penalty on the difference of RP and SP tasks, so it is similar to a small c, which imposes the constraint of the parameters of RP and SP being similar to each other.
3. The parameter c can take three types of values: 0, finite, and $+\infty$. (1) As $c = 0$, the constraint in Equation 14 becomes $||w_r - w_s||^2 \leq 0$ or equally the hard constraint $w_r = w_s$, which is typically specified in the classical NL analysis. (2) As $c = +\infty$, the constraint in Equation 14 disappears. This is similar to the case of training DNN or MNL models separately for RP and SP tasks without constraints. (3) As $0 < c < +\infty$, the constraint is soft, as in the MTLDNN training. Obviously the first two cases are the boundary examples of the third case. In the implementation, researchers seek to find the best soft constraint of λ (or implicitly but equivalently c), by searching in a sequence of λ values as hyper-parameters.
4. We made several simplifications in Proposition 1. Compared to the Equations 5 and 12, the condition of convexity does not hold in the deep architecture; the temperature factor is omitted; the regularizations of $||w_r||^2$ and $||w_s||^2$ are not explicitly discussed. Nonetheless, the key insights still hold that the regularized ERM can be converted to the constrained ERM and that λ and c are roughly inversely correlated.
5. The proof of Proposition 1 uses the Lagrangian form of convex optimization. The details are provided in Appendix I.

[4]Precisely, DNN training often involves the L_1 or L_2 regularizations, but not the soft constraints across multiple tasks, because a standard DNN targets to predict only one task.

3.3.2. *Feature Learning*

Second, MTLDNNs and DNNs rely on automatic feature learning, as opposed to the handcrafted feature learning in NL and MNL. The utility of MTLDNNs (Equations 1 and 2) is specified as a layer-by-layer function form, which makes DNNs a universal approximator and enables the capacity of automatic feature learning. This function form is in sharp contrast to NL and MNL, which relies on the handcrafted feature mapping function $\phi()$ and linear-in-parameter specification w_r and w_s (Equations 6 and 7). The handcrafted feature engineering is problematic because modelers' prior knowledge is rarely complete for the task at hand, and this incompleteness of knowledge can lead to function misspecification error and low prediction accuracy. On the other side, the strong approximation power of MTLDNNs enables it to approximate any underlying behavioral mechanism without relying on the completeness of modelers' domain knowledge. 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 only use the 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, other than the difference in terms of automatic vs. handcrafted feature learning, the difference between MTLDNNs and NL also reflects the difference between deep and shallow neural network. Studies show that DNN can represent the same function as SNN with an exponentially smaller amount of neurons [14, 36, 48], and the benefit of depth can also explain why MTLDNNs have stronger approximation power than NL. Note that this distinction between automatic vs. handcrafted feature learning exists not only in MTLDNNs and NL, but also generally in DNNs and MNL.



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

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

Besides the generic difference of automatic vs. handcrafted feature engineering, MTLDNNs and NL are also different in terms of the architectural design that is specific to the multitask learning problem. For example, the prototype MTLDNN architecture visualized in Figure 1 can vary with the number of shared layers (M_1) and the number of task-specific layers (M_2). The MTLDNN framework can flexibly control to what extent 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.

3.4. Approximation and Estimation Errors of MTLDNNs

MTLDNNs as a more generic model family do not necessarily imply a higher prediction accuracy, since the small approximation error obtained by a model with large complexity can be counteracted by the large estimation error. Based on statistical learning theory, a more complex model typically has smaller approximation errors (bias) but larger estimation error (variance) [62, 60] [5]. This problem can be one potential weakness of MTLDNNs, because MTLDNNs have more function complexity than the NL model. The function complexity of a model is formally measured by the Vapnik-Chervonenkis (VC) dimension [59, 60]. Specifically, 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 [22, 43], it provides sufficient insights for the purpose of this paper[6]. Generally speaking, while DNNs are more generic than multinomial logit models (MNL) in terms of the function class relationship [14, 36, 48], DNNs could perform worse due to its high model complexity and its corresponding large estimation errors. Therefore, we need to conduct empirical experiments to compare the performance of MTLDNNs and NL.

4. Experiment Setup

4.1. Data

An online survey was designed to explore the underlying factors that determine AV demand. The online survey was collected in Singapore with the help of a professional survey company Qualtrics.com. The survey consisted of one section about revealed preference (RP), one section about stated preference (SP), and one section for eliciting socioeconomic variables. In the RP part, we asked the respondents to report the home and working locations and their current travel modes. The revealed preference information was then used to generate the SP scenarios, which were designed by following the standard procedure. Each attribute in the SP section took three levels of values with the middle option equal to the value in the RP section so that the values were realistic and readily understandable to participants, and the other two levels were adjusted by multiplying the middle value by certain constants. All participants reported their socioeconomic information in the last section of the survey. The travel mode alternatives in RP include walking, public transit, driving, and ride sharing; on-demand AV use was added to the SP survey as the additional travel

[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 [63, 61, 4, 1]

mode in the SP part. In total, we collected 1,592 RP and 8,418 SP choice responses. In Appendix II, we provide the key summary statistics of our sample, and the comparison to the population in Singapore.

4.2. Training

RP and SP data are split into training and testing sets with the ratio of 4:1. One challenge in the MTLDNN training is its vast number of hyperparameters, and the performance of MTLDNNs largely depends on hyperparameters. To address this problem, we specify a hyperparameter space and search randomly within this space to identify the hyperparameters that cause high prediction accuracy [10]. Let S_H denote the hyperparameter space. We sample one group of hyperparameters $c_H^{(q)}$ from S_H, and choose the one with the highest prediction accuracy in the testing set. Formally,

$$\hat{c}_H = \underset{c_H \in \{c_H^{(1)}, c_H^{(2)}, \ldots, c_H^{(S)}\}}{\text{argmin}} R(X, Y; \hat{w}_r, \hat{w}_s, \hat{w}_0, \hat{T}; c_H) \tag{15}$$

in which $R(X, Y; \hat{w}_r, \hat{w}_s, \hat{w}_0, \hat{T}; c_H)$ is the estimated empirical risk (Equation 5); $S = 1,500$ represents the total number of random sampling in this study. Details of the hyperparameter space are incorporated in Appendix III, and the topc 10 MTLDNN architectures are provided in Appendix IV.

5. Experiment Results

5.1. Model Performance

Table 2 summarizes the prediction accuracy of MTLDNN (Top 1), MTLDNN ensemble over top 10 models (MTLDNN-E), the DNN separately trained for RP and SP (DNN-SPT), the DNN jointly trained for RP and SP (DNN-JOINT), NL with parameter constraints (NL-C), NL with no parameter constraints (NL-NC), the MNL separately trained for RP and SP (MNL-SPT), and the MNL jointly trained for RP and SP (MNL-JOINT). 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 summarizes the differences between the eight models in terms of feature learning and constraints, reflecting the contents in Table 1. Overall, the six non-MTLDNN models are designed as benchmarks to compare performance and to disentangle the reasons why MTLDNNs perform well.

Two MTLDNNs perform better than all the other six models in terms of joint and separate RP and SP prediction accuracy, as shown in Panel 1 in Table 2. In terms of the joint prediction accuracy, the top 1 MTLDNN model outperforms the NL models with and without parameter constraints by 4.5% and 5.0%. This about 5% prediction gain of MTLDNNs over NL models is consistent in the out-of-sample performance of the separate RP and SP datasets. The top 10 MTLDNN model ensemble also has higher prediction accuracy than all the other models in the testing set of joint RP and SP and separate RP and SP datasets, although MTLDNN-E performs

Table 2: Comparison of eight models

	MTLDNN (Top1)	**MTLDNN-E (Top10)**	**DNN-SPT**	**DNN-JOINT**	**NL-C**	**NL-NC**	**MNL-SPT**	**MNL-JOINT**
Panel 1: Prediction Accuracy								
Joint RP+SP (Testing)	60.0%	58.7%	53.4%	53.8%	55.4%	55.0%	55.0%	51.9%
RP (Testing)	69.9%	66.6%	65.8%	65.8%	65.4%	64.7%	64.5%	44.0%
SP (Testing)	58.2%	57.2%	51.1%	51.5%	53.5%	53.2%	53.2%	53.5%
Joint RP+SP (Training)	60.7%	62.2%	52.5%	52.9%	54.0%	54.5%	54.4%	50.3%
RP (Training)	69.1%	71.9%	59.8%	59.8%	58.9%	62.2%	62.1%	37.0%
SP (Training)	59.1%	60.3%	51.1%	51.5%	53.0%	53.0%	53.0%	52.8%
Panel 2: Different Characteristics of Models								
Feature Learning	A	A	A	A	H	H	H	H
Constraints	S	S	N	N	H	H	N	N

Notes: Feature learning: A, automatic; H, handcrafted. Constraints: S, soft constraints; H, hard constraints; N: no constraints.

about 1.3% worse than the top 1 MTLDNN model. Note that the MTLDNN models not only outperform classical choice modeling methods such as NL and MNL, they also perform better than the DNN models without the MTLDNN architectures, such as DNN-SPT and DNN-JOINT, demonstrating the importance of the soft constraints and architectural design in MTLDNN models. While the performance improvement of MTLDNNs over NL models is clear, the next question is how to attribute this improvement to two potential factors: the automatic feature learning and the soft constraints.

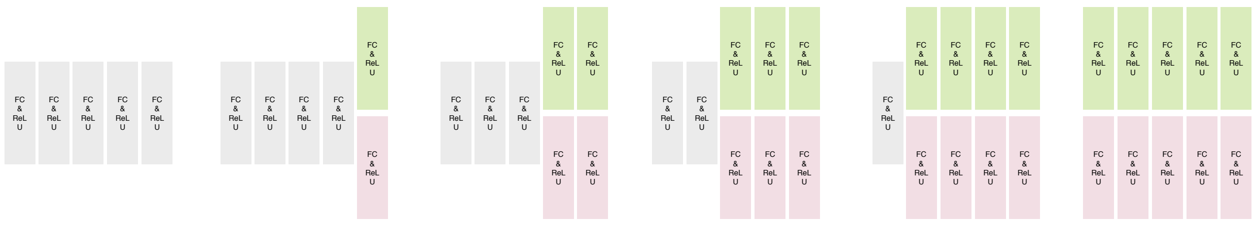
5.2. Sources of Performance Improvement in MTLDNNs

It is difficult to disentangle the two factors by directly comparing MTLDNNs and NL models since they are different by both factors. To disentangle the effects, we compare the eight models in a pair-wise manner, making each pair differ by only one factor. In addition, Figure 3 summarizes how the prediction accuracy of MTLDNNs varies with the regularization constraints and the architectural hyperparameters.

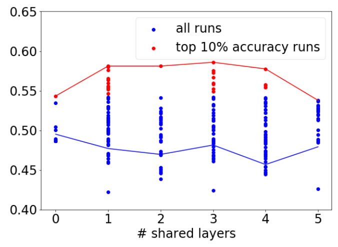
5.2.1. Constraints

When appropriately chosen, the hyperparameters of the soft constraints 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 3c-3e, the MTLDNN model cannot perform well, when λ_3 becomes too large or small, which implies that task-specific layers between RP and SP are either too similar or different. This finding also 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}.

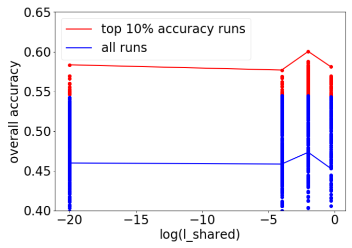
Compared to the soft constraints, the hard constraints in NL-C do not help to improve the model performance of NL-NC. As shown in Table 2, the NL-C and NL-NC perform similarly in terms of the joint performance of RP and SP. The prediction accuracy in NL-C and NL-NC is



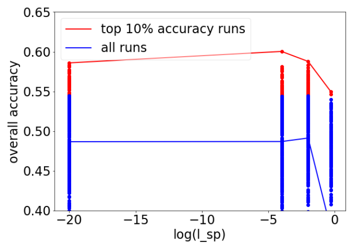
(a) Six Different Architectures: (5-0);(4-1);(3-2);(2-3);(1-4);(0-5)



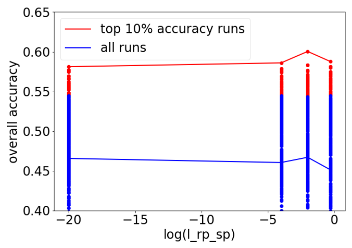
(b) Performance of Six Architectures



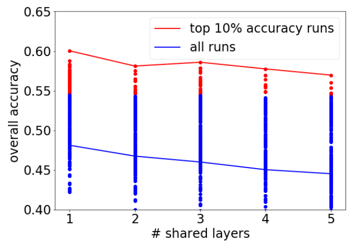
(c) λ_1



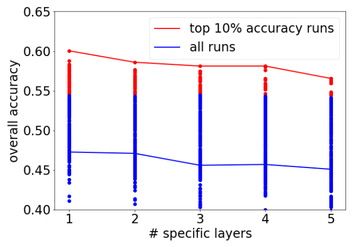
(d) λ_2



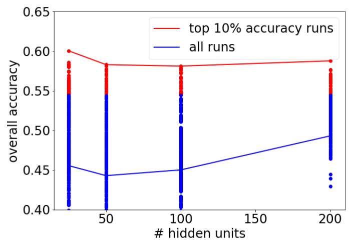
(e) λ_3



(f) Depth (shared)



(g) Depth (specific)



(h) Width

Fig. 3. Prediction accuracy with architectural design and regularization; row 1 visualizes six architectures; row 2 presents the prediction accuracy of the six architectures; from row 2 to 4, 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.

only different by 0.4%, which is much smaller than the 5% accuracy difference between MTLDNN and NL models and about 6.4% accuracy difference between MTLDNNs and DNNs. This result is consistent with our previous discussions: hard constraints imposed by domain experts are often limited, because they are less generic and flexible than the soft constraints in the MTLDNN models.

5.2.2. Feature Learning

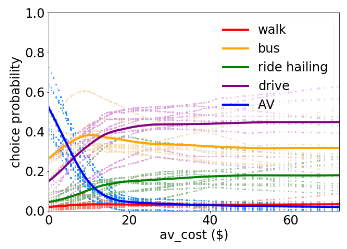
The architectural hyperparameters specific to MTLDNNs are quite effective in improving the model performance, because the MTLDNN architectures with both shared and task-specific layers outperform the DNN models with only shared or task-specific layers. Figure 3a visualizes a spectrum of MTLDNN examples, indexed by their shared vs. task-specific layers. On the left hand of Figure 3a, MTLDNNs become DNN-JOINT, which has only five shared layers without any task-specific layer. On the right hand of Figure 3a, MTLDNNs become DNN-SPT, which has only five task-specific layers without any shared layer. Between DNN-JOINT and DNN-SPT, MTLDNNs take various forms of architectures, varying with the ratio of shared vs. task-specific layers. Figure 3b visualizes the prediction accuracy of the six MTLDNN architectures in Figure 3a. In this set of MTLDNN models with 5 layers in total, the MTLDNN models with non-zero shared or task-specific layers perform substantially better than DNN-SPT and DNN-JOINT, and particularly the MTLDNN architecture with 3 shared layers and 2 task-specific layers performs the best. This result is the same as Table 2, which shows that top 1 MTLDNN outperforms DNN-SPT and DNN-JOINT by 6.6% and 6.2% (average about 6.4%) in the joint performance of RP and SP.

On the other side, naive applications of the feedforward DNN architectures do not contribute to the performance of MTLDNNs, since DNNs do not outperform MNL models and simply increasing depth and width do not improve the MTLDNNs' performance. First, comparing models between MNL-SPT and DNN-SPT, and between MNL-JOINT and DNN-JOINT helps to understand how the generic automatic feature learning of DNNs contributes to the performance improvement. Interestingly, as shown in Table 2, DNN-SPT performs worse than MNL-SPT by 1.6% prediction accuracy and DNN-JOINT performs only slightly better than MNL-JOINT by 1.9%. As a result, the effectiveness of DNNs' general structure is inconclusive. Second, as shown in Figures 3f-3h, the prediction accuracy of the MTLDNNs decreases as the architecture becomes deeper and wider, suggesting that naively increasing the scale of MTLDNNs cannot improve model performance at least in our current data set. This result is actually explainable via our previous discussion regarding how more complex models may lead to worse predictive performance, discussed in Section 3.4. While the approximation errors of deeper and wider MTLDNNs decrease, the large estimation error can exceed the gain. This result can also happen when the underlying data generating process (DGP) is similar to that of MNL, leading to only trivial or zero reduction of approximation error by using DNN to replace MNL. While our results are different from many studies that found DNN outperforming MNL in the travel behavioral analysis [44, 67], our finding is not unseen in previous studies [42]. In other words, our results imply that it cannot effectively help the model performance if researchers only naively apply the default feedforward DNN architecture and hope to solve domain-specific problems without any adjustment. The most effective ways to improve the MTLDNN models are to design architectures and regularization methods specific to the multitask learning problem. In fact, many of the recent new models in the deep learning community arise from the innovation of DNN architectures and regularizations [34, 54, 27], and this MTLDNN case is no exception.

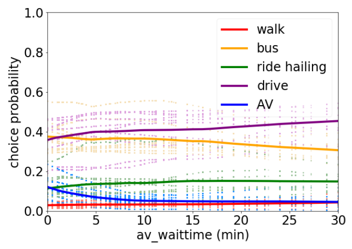
5.3. Interpreting MTLDNNs for AV Adoption

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 [53, 47, 9, 3, 50, 65].

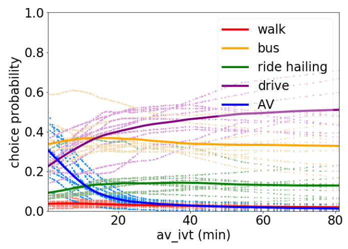
5.3.1. Substitution Patterns of Travel Mode Choice



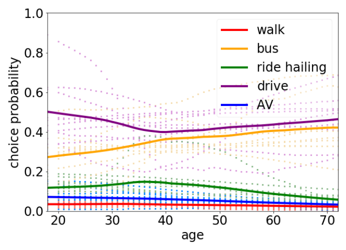
(a) AV cost



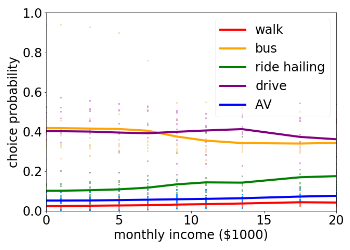
(b) AV wait time



(c) AV in-vehicle time



(d) Age



(e) Income

Fig. 4. Choice probability functions varying with inputs values; light curves are the individual MTLDNN results; dark ones are the average of top 10 models.

As shown in Figure 4, MTLDNNs reveal that the probability of choosing AVs is highly sensitive to the change of AV-specific cost variables, such as costs, waiting time, and in-vehicle travel time of AVs, but much less so to the change of socio-economic variables, such as age and income. In Figure 4, the y-axis represents the probability of choosing AVs; the x-axis represents the change of input variables; each curve represents how the probability varies with input variables holding all the other variables constant at the level of sample average. As shown by Figures 4a, 4b, and 4c, people are highly responsive to the AV-specific cost variables. For example, the probability of choosing AVs drops from about 50% to only 5%, as the AV cost increases from \$0 to \$20; Similarly, the probability drops from about 30% to only 5%, as the AV in-vehicle travel time increases from 0 to 20 minutes. Meanwhile, the two Figures 4a and 4c also show that AVs are primarily substituted by driving, as the cost and in-vehicle travel time of AVs increase. In contrast to the AV-specific variables, the probability of adopting AV is much less sensitive to socio-economic variables, as shown by Figures 4e and 4d: regarding the different values of age and income, the probability curve of adopting AV

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

Panel 1: MTLDNN	Walk	Public Transit	Ride Hailing	Driving	AV
Walk time	**-1.453(1.4)**	0.194(0.5)	0.375(0.7)	0.026(0.4)	0.185(0.4)
Public transit cost	-0.408(0.7)	**-0.631(0.7)**	0.091(0.5)	0.225(0.4)	0.004(0.4)
Public transit walk time	-0.100(0.5)	**-0.217(0.4)**	0.278(0.5)	0.032(0.3)	0.180(0.3)
Public transit wait time	0.036(0.4)	**-0.416(0.4)**	-0.033(0.3)	0.125(0.2)	0.146(0.3)
Public transit in-vehicle time	0.014(0.4)	**-0.370(0.5)**	-0.006(0.5)	0.122(0.4)	0.178(0.4)
Ride hail cost	0.231(0.7)	0.184(0.5)	**-0.634(0.7)**	-0.043(0.5)	0.516(0.4)
Ride hail wait time	-0.048(0.5)	0.248(0.4)	**-0.631(0.6)**	0.082(0.3)	-0.022(0.4)
Ride hail in-vehicle time	0.052(0.6)	-0.101(0.5)	**-1.384(1.0)**	0.197(0.4)	-0.062(0.6)
Drive cost	0.067(0.6)	0.344(0.6)	0.441(0.5)	**-0.571(0.8)**	0.408(0.5)
Drive walk time	0.243(0.3)	0.038(0.3)	0.291(0.3)	**-0.166(0.2)**	0.227(0.2)
Drive in-vehicle time	0.034(0.5)	0.369(0.7)	0.597(0.6)	**-0.517(0.6)**	0.474(0.5)
AV cost	0.023(0.5)	0.025(0.4)	0.422(0.7)	0.216(0.4)	**-0.895(1.0)**
AV wait time	0.194(0.3)	0.018(0.2)	0.214(0.4)	0.084(0.2)	**-0.280(0.4)**
AV in-vehicle time	-0.244(0.5)	-0.039(0.4)	0.379(0.6)	0.232(0.3)	**-0.827(0.8)**
Age	-0.430(1.4)	0.079(1.0)	-1.185(1.8)	0.234(0.6)	-0.768(1.0)
Income	0.495(0.8)	0.155(0.5)	-0.014(0.7)	-0.026(0.3)	0.004(0.5)
Panel 2: NL	Walk	Public Transit	Ride Hailing	Driving	AV
Walk time	**-1.907(1.9)**	0.126(0.1)	0.126(0.1)	0.126(0.1)	0.126(0.1)
Public transit cost	0.128(0.1)	**-0.506(0.4)**	0.128(0.1)	0.128(0.1)	0.128(0.1)
Public transit access time	0.084(0.1)	**-0.322(0.3)**	0.084(0.1)	0.084(0.1)	0.084(0.1)
Public transit transfer time	0.073(0.1)	**-0.260(0.2)**	0.073(0.1)	0.073(0.1)	0.073(0.1)
Public transit in-vehicle time	0.126(0.2)	**-0.458(0.4)**	0.126(0.2)	0.126(0.2)	0.126(0.2)
Ride hail cost	0.029(0.0)	0.029(0.0)	**-0.230(0.2)**	0.029(0.0)	0.029(0.0)
Ride hail wait time	0.036(0.0)	0.036(0.0)	**-0.313(0.2)**	0.036(0.0)	0.036(0.0)
Ride hail in-vehicle time	0.078(0.1)	0.078(0.1)	**-0.684(0.5)**	0.078(0.1)	0.078(0.1)
Drive cost	0.294(0.2)	0.294(0.2)	0.294(0.2)	**-0.787(1.1)**	0.294(0.2)
Drive walk time	0.119(0.1)	0.119(0.1)	0.119(0.1)	**-0.220(0.3)**	0.119(0.1)
Drive in-vehicle time	0.279(0.2)	0.279(0.2)	0.279(0.2)	**-0.450(0.5)**	0.279(0.2)
AV cost	0.047(0.1)	0.047(0.1)	0.047(0.1)	0.047(0.1)	**-0.439(0.4)**
AV wait time	0.029(0.0)	0.029(0.0)	0.029(0.0)	0.029(0.0)	**-0.269(0.2)**
AV in-vehicle time	0.065(0.1)	0.065(0.1)	0.065(0.1)	0.065(0.1)	**-0.628(0.6)**
Age	0.018(0.0)	0.164(0.1)	-0.064(0.0)	0.098(0.1)	-0.051(0.0)
Income	-0.008(0.0)	-0.049(0.0)	0.008(0.0)	0.039(0.0)	0.017(0.0)

is nearly flat everywhere. Overall, MTLDNNs reveal a substitution pattern that is intuitive and reasonable.

5.3.2. Elasticity Values of Travel Mode Choices

Table 3 presents the elasticity values of the five travel modes with respect to the input variables in SP. Panels 1 and 2 report the values in the top MTLDNN model and the NL model. To facilitate our discussion, we highlight the self-elasticity values on the main diagonal, which are the sensitivity of choosing alternatives regarding their own price attributes.

As shown in Table 3, the elasticity values in the MTLDNN model are overall reasonable in terms of their signs and magnitudes. As to the signs, 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 to choose this alternative. 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 in the MTLDNN model. For example, with 1% increase in walking time, people are less likely to choose walking with 1.45% probability; with 1% increase in the cost of public transit, people are less likely to choose public transit with 0.63% probability. Their magnitudes are not very far-off. However, it is impossible to evaluate which one of the two models reveals more realistic pattern because researchers can never know the underlying data generating process in empirical analysis.

Based on the elasticity values, we can rank the importance of the input variables regarding the adoption of AVs, and the ranking is similar to the result presented in Figure 4. Specifically, one percent increase in the AV cost and in-vehicle travel time leads to 0.895 and 0.827 percent decrease of the probability of using AVs, while the impact of age and income is smaller with values of -0.768 and 0.004 respectively. Overall, our results suggest that AV adoption heavily depends on its cost structure relative to the socio-economic information. While 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 and that the results from the MTLDNNs seem consistent with the long-standing findings in travel behavioral analysis.

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 yields the following three major findings.

First, it is theoretically feasible and effective to combine RP and SP data by using the MTLDNN framework, since 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 the tasks of RP and SP, owing to the universal approximation power, the diverse architectures, and flexible regularization methods. Second, MTLDNNs empirically outperform six benchmark models, and particularly outperform the NL models by about 5% prediction accuracy. This improvement can be mainly attributed to the architectures and regularizations specific to the multitask learning problem, but not much to the generic feedforward deep architectures. Third, MTLDNNs are also interpretable. MTLDNNs reveal that AVs can mainly substitute the driving mode as the costs of AVs vary and that the AV-specific variables are more important than socio-economic variables in determining the adoption of AVs.

While these results are promising, many questions still remain concerning MTLDNNs' feature engineering, model interpretation, statistical properties, and achievable prediction accuracy. As to features, it is unclear what the best method is to effectively engineer the input variables of RP and SP to match them as the inputs of the MTLDNNs. This is not a simple question since many different methods can apply [23]. As to model interpretation, this study analyzed substitution patterns and elasticity values. But model interpretability is an ambiguous concept, which leads to both the challenge of creating a set of definitive interpretation methods and the opportunity of extracting novel information that is unforeseen from classical choice modeling methods. The statistical property of MTLDNNs is another challenge [12]; for example, classical methods often model the unobserved random utility of RP and SP as a panel structure, the discussion of which does not exist in DNNs at least in an obvious way yet. Lastly, the MTLDNNs only improved the prediction accuracy of the NL models by 5%, and it is fair to ask whether this moderate improvement is worthy of the strenuous efforts of $1,500$ MTLDNN trainings for hyperparameter searching. In fact, this moderate improvement in prediction accuracy aligns with many of the past studies that find only limited improvement or even no improvement of DNNs over classical discrete choice models [44, 42]. To fully unleash the power of MTLDNNs, future studies need to explore the questions about feature engineering, model interpretation, and MTLDNNs' statistical properties.

It is also natural to ask whether our results are generalizable in two different ways: whether the results can be generalized to other studies in which RP and SP data sets are jointly analyzed, and whether our results about the adoption of AVs based on RP and SP are generalizable to the adoption of real AVs in the future. The first generalization question is relatively easier, because our MTLDNN framework is always generalizable to other studies, although the similarity of others' results to ours depends on specific contexts. The second generalization question is much more difficult, because the answer depends on the uncertainty of AVs and the risk preference of the potential AV users [64]. If the AV attributes in the future deviate significantly from the survey design, then it would be highly unlikely for our results to be generalizable. In short, despite its power, this MTLDNN framework cannot fully resolve the generalization challenge caused by future uncertainty.

Building upon our work, future studies can further develop this MTLDNN framework in choice modeling by investigating the effects of other MTLDNN architectures, applications, data structures, or even just larger sample sizes. 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 [37, 24, 40, 49]. Novel MTLDNN architectures can also be created in an automatic way by using sequential modeling techniques such as the autoML tools [18, 58, 72, 73]. Besides new MTLDNN architectures, future studies should also explore new applications. The new applications can be 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 resemble an advanced version of joint estimation, that has been traditionally used to combine tasks. In fact, combining RP and SP might not be even the most straightforward application for MTLDNNs, because the tasks of RP and SP are *heterogeneous* due to the AV-specific variables in the SP that do not exist in RP. As an alternative example, when MTLDNNs are used to combine the tasks of modeling travel mode choices from two cities, the two tasks are *homogenous* since the input and output dimensions of the two tasks match well. In addition, future studies may further improve the predictive performance of the MTLDNNs by using even larger sample size, which is often necessary to compensate for the high model complexity of DNNs, or by using unstructured data such as images and natural language, because DNNs are particularly powerful in learning the generalizable representations of the unstructured data. Despite some limitations, we hope that our work has demonstrated the possibility and the future opportunities of using this MTLDNN framework in choice modeling for 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] Leo Breiman. "Statistical modeling: The two cultures (with comments and a rejoinder by the author)". In: *Statistical science* 16.3 (2001), pp. 199–231.

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

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

[15] 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.

[16] 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.

[17] 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.

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

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

[20] 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.

[21] 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.

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

[23] Jen J Gong et al. “Predicting clinical outcomes across changing electronic health record systems”. In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* ACM, 2017, pp. 1497–1505.

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

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

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

[31] 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.

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

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

[34] 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.

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

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

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

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

[39] 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.

[40] 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.

[41] 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.

[42] 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.

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

[44] 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.

[45] 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.

[46] Bharath Ramsundar et al. "Massively multitask networks for drug discovery". In: *arXiv preprint arXiv:1502.02072* (2015).

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

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

[50] 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).

[51] Kenneth A Small, Erik T Verhoef, and Robin Lindsey. "Travel Demand". In: *The economics of urban transportation.* Vol. 2. Routledge, 2007.

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

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

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

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

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

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

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

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

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

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

[62] 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.

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

[64] Shenhao Wang and Jinhua Zhao. “Risk preference and adoption of autonomous vehicles”. In: *Transportation Research Part A: Policy and Practice* 126 (2019), pp. 215–229. ISSN: 0965-8564.

[65] 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).

[66] 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.

[67] 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.

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

[69] 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.

[70] 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.

[71] 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.

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

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

Appendix I: Proof of Proposition 1

Proof. The regularized ERM problem in MTLDNN is represented by the following equation:

$$\min_{w_r, w_s} L(w_r, w_s) = \hat{L}_R(w_r) + \hat{L}_S(w_s) + \lambda ||w_r - w_s||^2 \tag{16}$$

Note that the regularization term $\lambda ||w_r - w_s||^2$ penalizes differences between w_r and w_s, functioning in a way similar to the barrier in the interior point method. As the objective function is convex, the optimum values w_r^* and w_s^* satisfies the first order condition:

$$\nabla_{w_r} L(w_r, w_s) = \nabla_{w_r} \hat{L}_R(w_r^*) + 2\lambda(w_r^* - w_s^*) = 0 \tag{17}$$

$$\nabla_{w_s} L(w_r, w_s) = \nabla_{w_s} \hat{L}_S(w_s^*) - 2\lambda(w_r^* - w_s^*) = 0 \tag{18}$$

Now consider the constrained ERM problem:

$$\min_{w_r, w_s} L(w_r, w_s) = \hat{L}_R(w_r) + \hat{L}_S(w_s) \;\; s.t. \; ||w_r - w_s||^2 \leq c \tag{19}$$

in which $c := ||w_r^* - w_s^*||^2$. By solving the Lagrangian form of this constrained ERM problem, we can use the first order condition and the complementary slackness to show:

$$\nabla_{w_r} L(w_r, w_s) = \nabla_{w_r} \hat{L}_R(w_r^*) + 2\lambda(w_r^* - w_s^*) = 0 \tag{20}$$

$$\nabla_{w_s} L(w_r, w_s) = \nabla_{w_s} \hat{L}_S(w_s^*) - 2\lambda(w_r^* - w_s^*) = 0 \tag{21}$$

$$\lambda(||w_r^* - w_s^*||^2 - c) = 0 \tag{22}$$

The first two conditions are the same as the first order conditions of the regularized ERM; the third condition is satisfied by the definition of c. Therefore by construction, the three optimality conditions of the constrained ERM problem are satisfied. In other words, we have proved there always exist a constrained ERM problem that has the same optimum value as the regularized ERM problem when c is chosen wisely. In addition, by using the two first order conditions, we can show

$$c = \frac{\langle \nabla_{w_r} \hat{L}_R(w_r^*), \nabla_{w_r} \hat{L}_R(w_r^*) \rangle}{4\lambda^2} = \frac{\langle \nabla_{w_s} \hat{L}_S(w_s^*), \nabla_{w_s} \hat{L}_S(w_s^*) \rangle}{4\lambda^2} \tag{23}$$

which suggests that c and λ are inversely related.

Appendix II: Descriptive Summary Statistics

The key statistics of our samples are summarized in Table 4, and a comparison of age and income distribution between the sample and the population is summarized in Table 5. 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 all other income groups is close to that of the population.

Table 4: 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 5: 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 III: Hyperparameter Space

Table 6: 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 7. However, the wise choice of regularization hyperparameters helps to improve model performance.

Table 7: 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 8: Elasticities of five travel modes with respect to input variables (RP)

Panel 1: MTLDNN Model	Walk	Public Transit	Ride Hailing	Driving
Walk time	**-2.021(1.6)**	0.600(0.6)	1.478(1.1)	-0.039(0.6)
Public transit cost	-0.356(0.9)	**-0.152(0.4)**	0.678(0.5)	0.140(0.5)
Public transit walk time	-0.234(0.6)	**0.047(0.3)**	0.421(0.4)	0.013(0.3)
Public transit wait time	-0.118(0.4)	**0.050(0.2)**	0.022(0.3)	0.297(0.3)
Public transit in-vehicle time	-0.033(0.3)	**-0.005(0.2)**	0.016(0.3)	0.038(0.4)
Ride hail cost	-0.051(1.0)	0.348(0.6)	**-0.195(0.5)**	-0.300(0.5)
Ride hail wait time	-0.116(0.4)	0.065(0.2)	**-0.871(0.6)**	0.141(0.3)
Ride hail in-vehicle time	0.919(1.0)	-0.274(0.4)	**-0.899(0.4)**	-0.217(0.5)
Drive cost	-0.092(0.4)	-0.001(0.3)	-0.023(0.3)	**-0.073(0.6)**
Drive walk time	0.133(0.2)	-0.100(0.1)	0.174(0.2)	**-0.086(0.2)**
Drive in-vehicle time	-0.507(0.7)	0.229(0.4)	0.490(0.4)	**-0.303(0.5)**
Age	-0.653(1.1)	0.239(0.6)	-0.658(1.1)	0.333(0.6)
Income	0.195(0.5)	-0.056(0.3)	-0.050(0.5)	0.087(0.3)
Panel 2: NL Model	Walk	Public Transit	Ride Hailing	Driving
Walk time	**-0.483(0.4)**	0.177(0.1)	0.177(0.1)	0.177(0.1)
Public transit cost	0.127(0.1)	**-0.088(0.0)**	0.127(0.1)	0.127(0.1)
Public transit access time	0.085(0.1)	**-0.061(0.1)**	0.085(0.1)	0.085(0.1)
Public transit transfer time	0.070(0.0)	**-0.048(0.0)**	0.070(0.0)	0.070(0.0)
Public transit in-vehicle time	0.140(0.1)	**-0.097(0.1)**	0.140(0.1)	0.140(0.1)
Ride hail cost	0.007(0.0)	0.007(0.0)	**-0.088(0.1)**	0.007(0.0)
Ride hail wait time	0.009(0.0)	0.009(0.0)	**-0.117(0.1)**	0.009(0.0)
Ride hail in-vehicle time	0.021(0.0)	0.021(0.0)	**-0.258(0.2)**	0.021(0.0)
Drive cost	0.010(0.0)	0.010(0.0)	0.010(0.0)	**-0.571(0.4)**
Drive walk time	0.002(0.0)	0.002(0.0)	0.002(0.0)	**-0.106(0.1)**
Drive in-vehicle time	0.005(0.0)	0.005(0.0)	0.005(0.0)	**-0.224(0.2)**
Age	0.021(0.0)	0.168(0.1)	-0.018(0.0)	0.002(0.0)
Income	-0.008(0.0)	-0.047(0.0)	0.002(0.0)	0.001(0.0)

Appendix VI: Coefficients of the Nested Logit Model

Table 9: Coefficients of the nested logit model

Variable Name	Value	Std Err	t-test	p-value
ASC_RP_BUS	2.69	0.217	12.4	0
ASC_RP_RS	0.621	0.241	2.58	0.00993
ASC_RP_WALK	1.93	0.234	8.28	2.22e-16
ASC_SP_AV	-0.215	0.0589	-3.64	0.000268
ASC_SP_BUS	0.0411	0.039	1.06	0.291
ASC_SP_DRIVE	0.864	0.139	6.22	4.87e-10
ASC_SP_RS	-0.272	0.0644	-4.22	2.43e-05
ASC_SP_WALK	-0.418	0.0859	-4.87	1.14e-06
B_AGE_AV	-0.0592	0.0308	-1.92	0.0544
B_AGE_BUS	0.0883	0.0254	3.48	0.00051
B_AGE_DRIVE	0.0262	0.0205	1.28	0.202
B_AGE_RS	-0.0769	0.032	-2.41	0.0162
B_AGE_WALK	0.0217	0.0292	0.741	0.459
B_COST_AV	-0.178	0.0339	-5.24	1.62e-07
B_COST_BUS	-0.158	0.0277	-5.71	1.14e-08
B_COST_DRIVE	-0.443	0.0695	-6.38	1.81e-10
B_COST_RS	-0.0843	0.0227	-3.71	0.000204
B_EDU_AV	0.114	0.0559	2.05	0.0407
B_EDU_BUS	-0.0741	0.0429	-1.73	0.0837
B_EDU_DRIVE	0.0218	0.0386	0.564	0.573
B_EDU_RS	0.0137	0.0542	0.254	0.8
B_EDU_WALK	-0.0757	0.0563	-1.35	0.178
B_FULLJOB_AV	-0.0135	0.0452	-0.298	0.766
B_FULLJOB_BUS	0.102	0.0374	2.74	0.00615
B_FULLJOB_DRIVE	-0.077	0.0318	-2.42	0.0154
B_FULLJOB_RS	0.00992	0.0446	0.223	0.824
B_FULLJOB_WALK	-0.0219	0.0456	-0.48	0.631
B_HEDU_AV	-0.0144	0.0479	-0.301	0.763
B_HEDU_BUS	-0.0385	0.0369	-1.05	0.296
B_HEDU_DRIVE	0.00549	0.0324	0.169	0.866
B_HEDU_RS	0.057	0.0488	1.17	0.242
B_HEDU_WALK	-0.0095	0.049	-0.194	0.846
B_INC_AV	0.0322	0.0142	2.26	0.0237
B_INC_BUS	-0.0469	0.013	-3.61	0.000309
B_INC_DRIVE	0.0174	0.0099	1.76	0.0788
B_INC_RS	0.0142	0.0133	1.07	0.286
B_INC_WALK	-0.0168	0.0138	-1.22	0.223
B_IVT_AV	-0.24	0.0427	-5.62	1.93e-08
B_IVT_BUS	-0.179	0.0298	-6	1.93e-09
B_IVT_DRIVE	-0.241	0.0393	-6.13	8.59e-10
B_IVT_RS	-0.227	0.0404	-5.61	1.98e-08
B_LEDU_AV	0.217	0.0923	2.35	0.0188
B_LEDU_BUS	-0.178	0.0716	-2.49	0.0127
B_LEDU_DRIVE	-0.146	0.0674	-2.17	0.0303
B_LEDU_RS	0.0881	0.0907	0.971	0.332
B_LEDU_WALK	0.0192	0.0858	0.224	0.823
B_MALE_AV	0.0314	0.0325	0.968	0.333
B_MALE_BUS	-0.0399	0.0258	-1.54	0.122
B_MALE_DRIVE	-0.00143	0.0217	-0.0659	0.947
B_MALE_RS	-0.0429	0.0333	-1.29	0.198
B_MALE_WALK	0.0528	0.036	1.46	0.143
B_OLD_AV	-0.163	0.0849	-1.92	0.0543
B_OLD_BUS	-0.00337	0.0543	-0.062	0.951
B_OLD_DRIVE	0.0654	0.0505	1.29	0.195
B_OLD_RS	0.0657	0.0764	0.861	0.389
B_OLD_WALK	0.0356	0.0677	0.526	0.599
B_WAITTIME_AV	-0.0954	0.0237	-4.02	5.82e-05
B_WAITTIME_BUS	-0.0879	0.0189	-4.65	3.29e-06
B_WAITTIME_RS	-0.0961	0.0236	-4.08	4.57e-05
B_WALKTIME_BUS	-0.147	0.0255	-5.77	7.81e-09
B_WALKTIME_DRIVE	-0.146	0.0257	-5.67	1.41e-08
B_WALKTIME_WALK	-0.758	0.124	-6.12	9.08e-10
B_YOUNG_AV	-0.0659	0.0491	-1.34	0.18
B_YOUNG_BUS	0.141	0.0416	3.4	0.000686
B_YOUNG_DRIVE	-0.0784	0.0369	-2.12	0.0338
B_YOUNG_RS	-0.00151	0.0474	-0.0318	0.975
B_YOUNG_WALK	0.00468	0.0493	0.0949	0.924
MU_SP	2.46	0.383	6.41	1.44e-10