Incorporating Domain Knowledge in Deep Neural Networks for Mode Choice Analysis

Shadi Haj Yahia*, Omar Mansour, Tomer Toledo

Faculty of Civil and Environmental Engineering,
Technion – Israel Institute of Technology, Haifa 32000, Israel
Emails: shadi8@campus.technion.ac.il
omar@technion.ac.il
toledo@technion.ac.il

SHORT SUMMARY

Discrete choice models (DCM) are widely used in travel demand analysis to understand and predict choice behaviors. However, a priori specification of the utility functions is required for model estimation, leading to subjectivity and potential inaccuracies. Machine learning (ML) approaches have emerged as a promising solution but lack interpretability and may not capture expected relationships. This study proposes a framework that supports the development of interpretable models that incorporate domain knowledge and prior beliefs. The framework includes pseudo data samples and a loss function to measure relationship fulfillment. This approach combines the flexibility of ML structures with econometrics and interpretable behavioral analysis, improving model interpretability. The proposed framework's potential is demonstrated through a case study, providing a promising avenue for the advancement of data-driven approaches in DCM.

Keywords: Deep neural networks, discrete choice models, domain knowledge, interpretability

1. INTRODUCTION

Discrete choice models (DCMs) are used in travel demand analysis to understand individuals' decision-making processes. Most DCMs are formulated as random utility models (RUMs) that assume individuals make decisions based on maximizing utility. However, specifying a plausible utility model that captures these complexities is a challenging task (Torres et al., 2011). Recently, data-driven approaches using machine learning (ML) methods have emerged as a promising avenue to overcome the limitations of RUM specifications. Deep neural networks (DNNs) are an increasingly popular data-driven approach that has shown higher prediction accuracy in many tasks.

Unlike RUM, DNN models require essentially no a priori beliefs about the nature of the true underlying relationships among variables. DNN models can find complex non-linear specifications, and their high flexibility means that the role of the analyst is minimized. However, their "black box" form limits their interpretability, and the extracted relationships may not be consistent with domain knowledge (Van Cranenburgh et al., 2021; Wang et al., 2020b).

To address these limitations, some studies combine RUM and ML. For example, Sifringer et al., (2020, 2018) added a DNN-learned utility term to the traditional interpretable RUM utility function. This improves the model's fit to the data but the unbounded DNN term may dominate and prevent interpretation. The decision on which variables enter each part is subjective.

Another approach that was proposed by Wang et al., (2020a) is to use an alternative-specific utility deep neural network (ASU-DNN) architecture, which maintains separate utility functions for each alternative that depend only on its own attributes, resembling RUM. The model is more interpretable compared to fully connected DNNs and achieved comparable or better fit to the data. However, it still might suffer from unreasonable relationships among explanatory variables and choices.

Current methods for interpretability lack control over the relationships among variables and choices, making them inconsistent with domain knowledge and limiting their application in predicting new policies (Alwosheel et al., 2021, 2019). This study proposes a framework that incorporates domain knowledge through constraints and a loss function to penalize violations. The proposed approach preserves flexibility and can be implemented on any model architecture, providing control over the model's behavior for better travel choice predictions.

2. METHODOLOGY

The idea behind incorporating domain knowledge in DNNs involves augmenting the data given to the model and modifying the loss function that the model optimizes. To achieve this, additional data, termed pseudo data, is generated to hold the targeted knowledge that the model is expected to capture. The loss function is then formulated to include terms that use this data, in combination with the original model loss function, such as the negative log-likelihood. The additional loss terms measure the extent to which the trained model is consistent with the domain knowledge.

The overall framework for incorporating domain knowledge into DNNs is shown in Figure 1 and is independent of the model structure, allowing for seamless integration with existing DNN architectures. The model is trained on two sets of inputs: the originally available observed data and domain knowledge, which is mathematically formulated as a set of constraints on the outcomes of the trained model. The observed data represents the available dataset collected, including socio-economic characteristics of decision makers, attributes of the alternatives, and choices. The domain knowledge represents the knowledge that the modeler wants to incorporate into the model and expects to be captured (e.g., directions of sensitivities).

In this work, the modeler's prior expectations are related to signs of the effects of an alternative's attributes on its own utility. For example, these may be negative effects of mode travel times and costs on the utilities of these modes. In this case , the model is constrained to learn a monotonically decreasing probability of choosing an alternative with respect to its travel time and cost and, consequently, monotonically increasing probabilities of choosing the remaining alternatives.

Consider a training set consists of N samples $\{(x_i, y_i)\}, i = 1, \dots, N$, where x_i is a feature vector in $x \in \mathbb{R}^{\mathcal{D}}$, and y_i is the discrete choice among C alternatives, $y_i \in \{1, .., C\}$. Let $p_c(x_i)$ be the probability of choosing alternative c given input x_i, and $x_i[m]$ is the value of feature m in the feature vector. The estimated model is considered to be monotonically increasing in $p_c$with respect to feature m if $p_c(x_i) \geq p_c(x_j)$ for any two feature vectors x_i, x_j, such that $x_i[m] \geq x_j[m]$ and $x_i[h] = x_j[h]$, for all $h \in \mathcal{D} \backslash m$. The opposite applies for decreasing monotonicity. The rest of the components are described as follows.

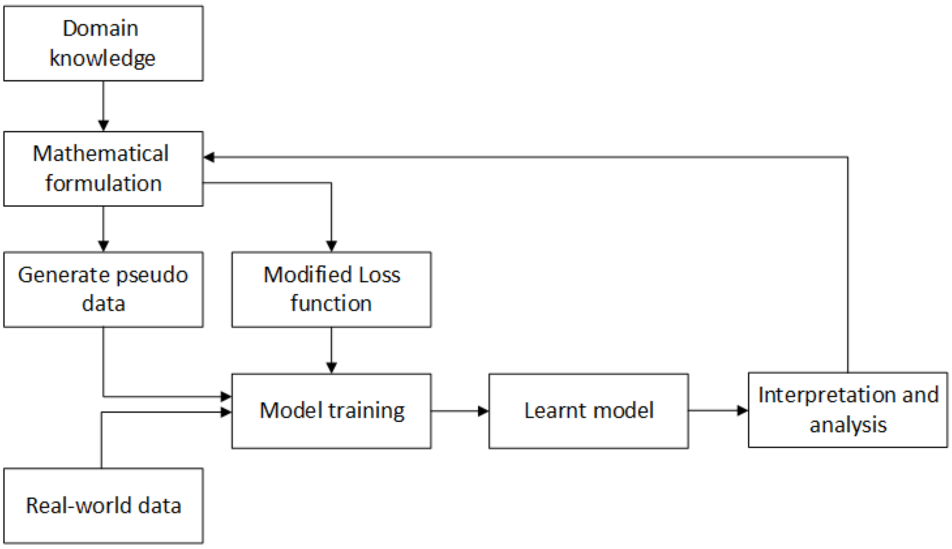


Figure 1. Overall framework for incorporating domain knowledge

Generating Pseudo Data

Following monotonicity constraints above, pseudo data can be generated as pairs of samples to numerically approximate the probabilities' derivatives that are constrained. For each monotonicity constraint with respect to a feature m, K pseudo samples are generated uniformly along the region values of that feature $x^{*}_{k,1}$. Each pseudo sample is then paired with another pseudo one, such that the second pseudo sample has a positive incremental change applied to feature m. The relationship required for an increasing monotonicity constraint of probability of choosing alternative c with respect to feature m is $p_c\left(x^{*}_{k,2}\right) - p_c\left(x^{*}_{k,1}\right) \geq 0$.

The pseudo data does not require labels (i.e., chosen alternatives), as they are only used for capturing domain knowledge, not for predicting the chosen alternative. This ability to generate pseudo samples enhances the model in three ways:

1. When the dataset is small, the pseudo dataset helps increase the dataset size to learn the model's parameters.
2. When the input feature region is imperfectly covered, the pseudo data helps fill gaps and enforce the model to learn along the full range of possible values.
3. Generating pseudo data outside the range of current values for specific features helps enforcing better learning, hence enabling extrapolation in the outer regions (i.e., unseen scenarios).

Loss Function

The loss function includes two components: prediction loss and domain knowledge loss. The prediction loss quantifies the accuracy of predictions and can be calculated for example using the negative log-likelihood ($\mathcal{L}_{NLL}$) method commonly used in RUMs. This calculation is performed only for samples with observed choices and is represented by the following formula:

$$\mathcal{L}_{NLL} = -\sum_{i=1}^{N} \sum_{c \in C} g_{i,c} \cdot log\left(p_{i,c}\right) \tag{1}$$

Where $g_{i,c}$ equals 1 if alternative c is chosen by individual i and 0 otherwise.

The domain knowledge loss measures the violation of monotonicity constraints on the probability of choosing alternative c with respect to feature m. This is determined using pseudo sample pairs that estimate the derivatives of the probabilities, represented by the following formula:

$$\mathcal{L}_{c,m} = \sum_{k=1}^{K} \max\left(0, d_{c,m} \cdot \frac{p_c(x^*_{k,2}) - p_c(x^*_{k,1})}{\Delta x^*_m}\right) \tag{2}$$

Where $d_{c,m}$ equals 1 if the probability of choosing alternative c with respect to feature m should be increasing and -1 otherwise.

If it is assumed that when the probability of choosing alternative c with respect to feature m is in one direction, the probability of choosing other alternatives should be in the opposite direction, the total loss to be minimized can be expressed as follows:

$$\min \mathcal{L}_{total} = \mathcal{L}_{NLL} + \sum_{m \in M} \sum_{c \in C} w_{c,m} \cdot \mathcal{L}_{c,m} \tag{3}$$

Where M represents the indices of the features that constrain the probabilities, and $w_{c,m}$ represents the weight of each constraint violation penalty.

Model Training

The training process is illustrated in Figure 2. Observed data, represented as vector $\mathbf{x}$, and a vector of pseudo sample pairs $\mathbf{x}^* = \{(x^*_{1,1}, x^*_{1,2}), \ldots, (x^*_{k,1}, x^*_{k,1})\}$ are fed into the model. The total loss, calculated as a combination of the prediction loss from the observed samples $\mathbf{x}$ and the domain knowledge loss from the pseudo data $\mathbf{x}^*$, is minimized using the backpropagation technique. This process continues iteratively until convergence is achieved.

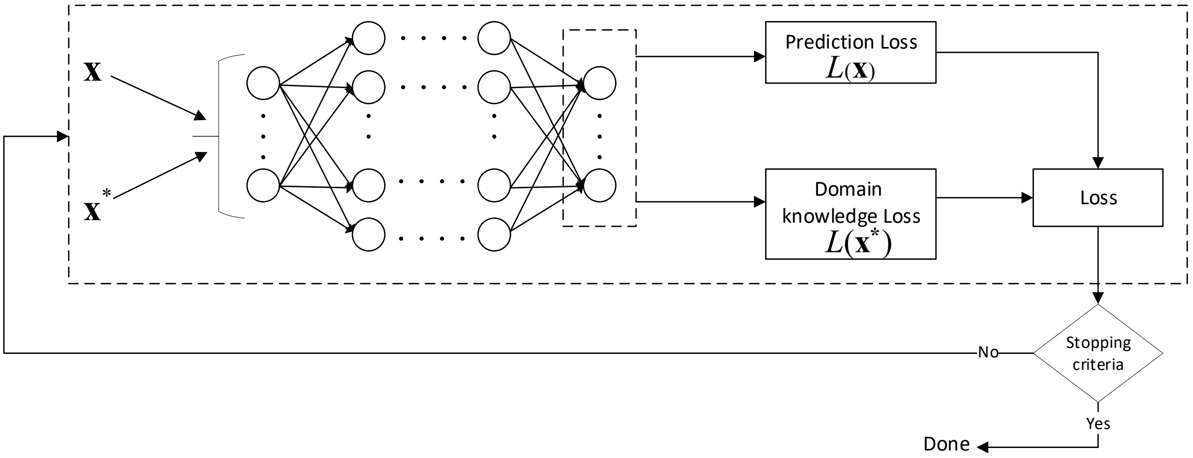


Figure 2. Model training process

3. RESULTS AND DISCUSSION

The methodology outlined above was applied to a mode choice dataset to assess the potential of incorporating domain knowledge in a DNN model and examine the impact of such knowledge on the resulting economic information.

Dataset

The experiment was based on the Swissmetro dataset, which is a publicly available stated preference survey collected in Switzerland in 1998 (Bierlaire et al., 2001). Participants were asked to provide information regarding their preferred mode of transportation between the new Swissmetro (SM) mode, car, and train. Travel time and cost were considered as the key descriptive variables for each alternative mode. Observations with missing alternatives or outliers were removed. The dataset was then divided into training, validation, and testing sets in the ratio of 60:20:20.

Experimental Design

The proposed methodology was implemented on two model architectures: DNN and ASU-DNN. The DNN model was an off-the-shelf model, while the ASU-DNN model was proposed by Wang et al., (2020a) and calculates alternative-specific utilities. Both models were estimated in both an unconstrained and a constrained (i.e., with domain knowledge) version. The constrained models are referred to as C-DNN and C-ASU-DNN, respectively. In addition, a Multinomial Logit (MNL) model was also estimated for comparison.

The domain knowledge incorporated in the constrained models includes negative own-sensitivities of choice probability to travel time and cost and positive cross-sensitivities. All constraints are incorporated simultaneously. The models' negative log-likelihood and prediction accuracy were measured on each of the datasets. Predicted market shares were also calculated for each model. Choice probabilities with respect to each feature were then presented to demonstrate the fulfillment of domain knowledge.

Results

Prediction performance

Table 1 presents the negative log-likelihood (NLL) and accuracy of each estimated model. The results indicate that the DNN model provides the best NLL and accuracy, thanks to its high ability of empirical fit to data. The ASU-DNN also demonstrates good fit to data. When domain knowledge is introduced, constrained models become less flexible and achieve lower fit to data compared to unconstrained ones. This is expected since the introduction of constraints to the models limits the search space for the optimal fit and might restrict the flexibility of the model. Nonetheless, the decrease in accuracy in testing is only 2.1%.

Table 1. Negative log-likelihood (NLL) and prediction accuracy

	Training		Testing	
Model	NLL	Acc [%]	NLL	Acc [%]
DNN	3182	70.5	1133	69.2
Constrained DNN	3336	68.7	1189	67.1
ASU-DNN	3438	68.3	1188	67.7
Constrained ASU-DNN	3577	66.7	1235	65.6
MNL	3508	67.9	1209	66.2

Market shares

While prediction accuracy relates to predicting choices at the level of individuals, transportation policy planners are mainly interested in prediction at the market level. Table 2 shows the predicted market shares by the different models and the root mean square error (RMSE) in each model. The constrained models provide better market shares in terms of RMSE in the training set compared to the unconstrained models but perform worse in the testing set. The DNN and ASU-DNN models outperform the MNL model in terms of RMSE in the testing set, which was guaranteed to provide exact market shares in training. Although the constrained models have worse performance than the unconstrained models, the RMSE values are within a range of 2.4% and are not much different from the observed shares in the sample.

Table 2. Market shares of travel modes

Training set

	DNN	C-DNN	ASU-DNN	C-ASU-DNN	MNL	Observed
Train	5.5%	6.0%	7.6%	4.6%	6.2%	6.2%
SM	55.0%	58.3%	54.7%	56.3%	56.9%	56.9%
Car	39.5%	35.7%	37.7%	39.1%	36.9%	36.9%
RMSE	1.9%	1.1%	1.6%	1.6%	0%	

Testing set

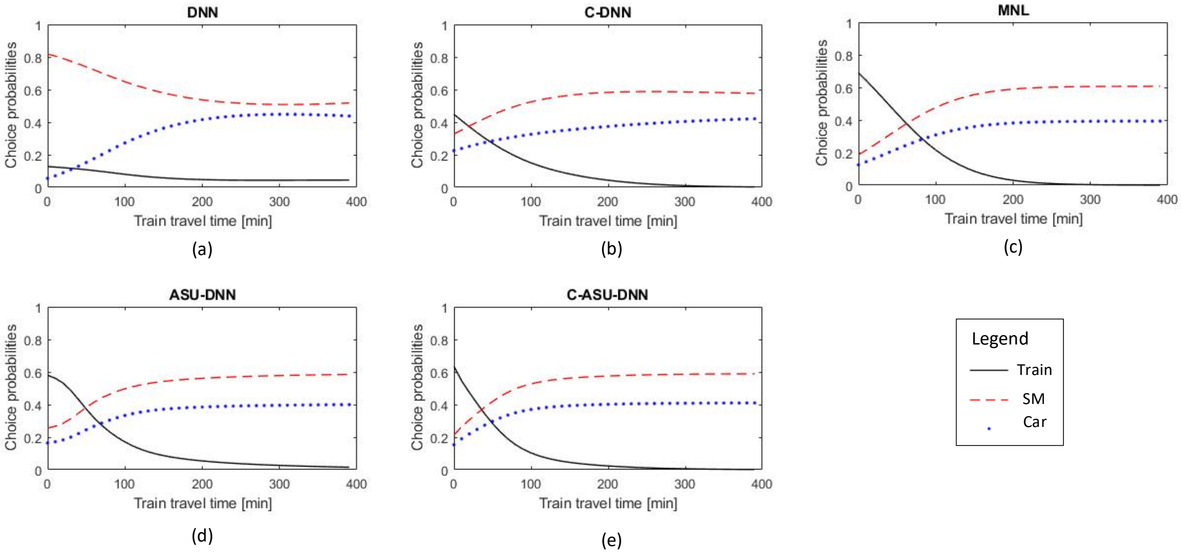
	DNN	C-DNN	ASU-DNN	C-ASU-DNN	MNL	Observed
Train	5.4%	6.1%	7.5%	4.5%	6.2%	7.4%
SM	55.9%	58.7%	55.2%	56.8%	57.7%	55.4%
Car	38.7%	35.1%	37.4%	38.6%	36.1%	37.2%
RMSE	1.5%	2.4%	0.2%	2.0%	1.6%	

Choice probabilities

To demonstrate the consistency with expected domain knowledge, choice probability functions provided by the different models were calculated as a function of each of the six variables. They were calculated using the partial dependence plots (PDP) method which calculates choice probabilities for every possible value of the variable for each observation (Friedman, 2001). Three of them are illustrated in Figure 3-5. In remaining three, the constrained models satisfied the constraints while unconstrained ones did not. They are not presented due to paper length constraints.

The estimated coefficients in the MNL are with the expected sign (i.e., negative coefficients of travel time and cost in all utility functions), therefore, the directions of choice probabilities are consistent with domain knowledge as can be seen in Figure 3-5. However, choice probabilities may not always be consistent with domain knowledge when derived from unconstrained models, even in ASU-DNN where utilities are calculated independently from the others following RUM. This inconsistency could be restrained when domain knowledge is incorporated into the models.

For example, Figure 3 presents choice probabilities as a function of train travel time. It is expected that SM and car shares would increase at the expense of the decrease in train shares, as train travel time increases. Figure 3(a) shows that DNN, while the most accurate, reveals unreasonable decreasing of SM choice probability. This finding is unreasonable since train becomes less attractive, and SM shares should not be negatively affected. This knowledge is considered in C-DNN, and choice probabilities become more reasonable as illustrated in Figure 3(b). The rest of the models behave as expected.



(a)

(b)

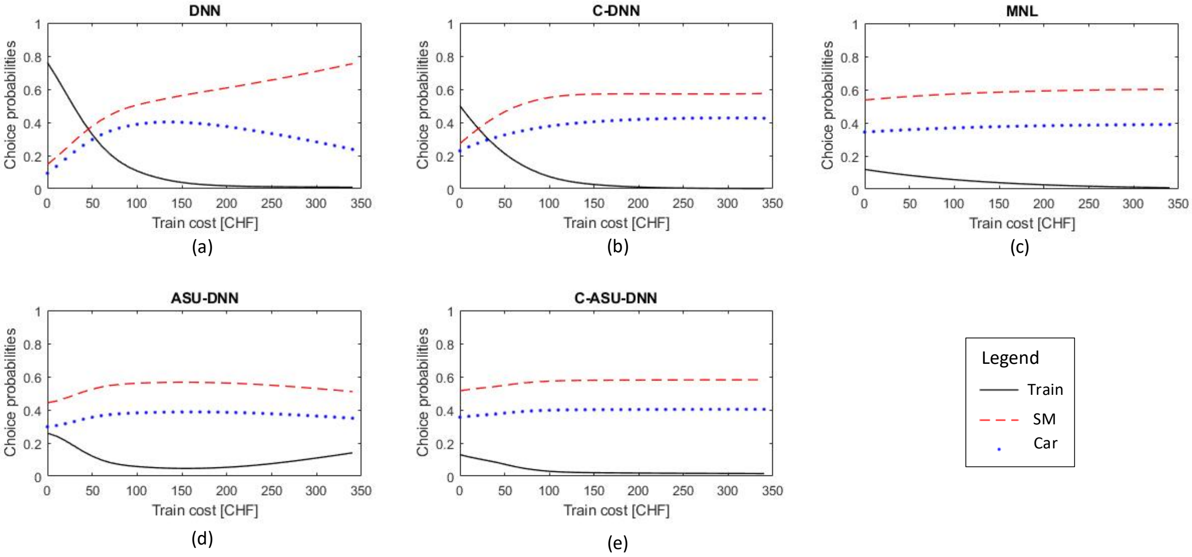
(c)

(d)

(e)

Figure 3. Alternatives' choice probabilities as a function of train travel time

In Figure 4, choice probabilities are calculated as a function of train cost. It is expected that SM and car shares would increase at the expense of the decrease in train shares, as train cost increases. However, DNN fulfills this expectation only up to about 150 CHF train cost, as shown in Figure 4(a). At this point, SM shares increase drastically at the expense of car shares, which start decreasing. This is unexpected since increased train cost must not negatively affect car shares. At worst, car shares would not change (i.e., would not increase), and train users would shift only to SM. Another unexpected finding can be found in the ASU-DNN model in Figure 4(d). Around a train cost of 150 CHF, all choice probabilities switch directions. Both models were corrected by incorporating knowledge, as shown in Figure 4(b) and Figure 4(e) for C-DNN and C-ASU-DNN, respectively.



(a)

(b)

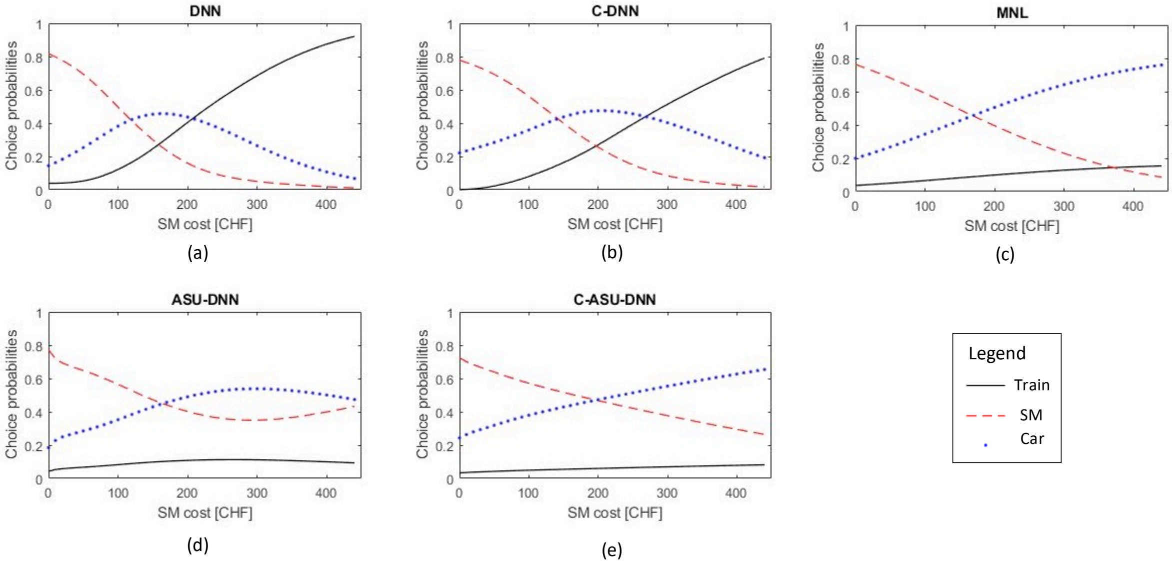
(c)

(d)

(e)

Figure 4. Alternatives' choice probabilities as a function of train cost

Figure 5 presents choice probabilities as a function of SM cost, where train and car shares are expected to increase at the expense of decrease in SM shares. In Figure 5(a), DNN fails to fulfill this knowledge at costs above 170 CHF, where car shares start decreasing, as if higher SM cost makes using the car less attractive. Although domain knowledge was incorporated, C-DNN was not corrected. In ASU-DNN, however, all choice probabilities switch directions at 300 CHF SM cost, as illustrated in Figure 5(d), which had been overcome by incorporating knowledge, as shown in Figure 5(e).



(a) (b) (c)

(d) (e)

Figure 5. Alternatives' choice probabilities as a function of SM cost

In conclusion, while accurate, unconstrained models that rely solely on data may produce unreasonable interpretations of choice probabilities, making them unsuitable for use in policy-making processes. The results obtained through the proposed methodology of incorporating domain knowledge into these models demonstrate the potential of achieving more interpretable results while still relying on data in a controllable manner. In C-DNN, only one constraint out of 18 was not fulfilled (i.e., increasing car choice probability as a function of SM cost, Figure 5 (b)), whereas all constraints were fulfilled in C-ASU-DNN. While constraints may not always be fully satisfied, they can significantly enhance the models' consistency with domain knowledge, making them more useful for choice analysis and planning purposes.

4. CONCLUSIONS

This study addresses the limitations of uncontrollable DNN application in discrete choice analysis. Incorporating domain knowledge into DNN is crucial for its interpretation and usability. The proposed framework enhances model consistency with domain knowledge by introducing constraints, making it easy to implement on different architectures. The case study on Swissmetro dataset demonstrates a tradeoff between accuracy and interpretability, showing promising results in combining domain knowledge with DNN models for choice analysis. Future research could explore the proposed framework's generalizability to other discrete choice modeling problems and datasets. The proposed framework could also be extended to incorporate other types of domain knowledge, such as prior distributions on model parameters or constraints on the functional form of the model.

REFERENCES

Alwosheel A., van Cranenburgh S., and Chorus C. G. (2021). “Why did you predict that? Towards explainable artificial neural networks for travel demand analysis”. Transportation Research Part C: Emerging Technologies, 128, 103143.

Alwosheel A., Van Cranenburgh S., and Chorus C. G. (2019). “‘Computer says no’is not enough: Using prototypical examples to diagnose artificial neural networks for discrete choice analysis”. Journal of choice modelling, 33, 100186.

Bierlaire M., Axhausen K. and Abay G. (2001). “The acceptance of modal innovation: The case of Swissmetro”. In ‘Proceedings of the Swiss Transport Research Conference’, Ascona, Switzerland.

Friedman J. H. (2001). “Greedy function approximation: a gradient boosting machine”. Annals of statistics, 1189-1232.

Sifringer B., Lurkin V., and Alahi A. (2018). “Let me not lie: Learning multinomial logit”. arXiv preprint arXiv:1812.09747.

Sifringer B., Lurkin V., and Alahi A. (2020). “Enhancing discrete choice models with representation learning”. Transportation Research Part B: Methodological, 140, 236-261.

Torres C., Hanley N., and Riera A. (2011). “How wrong can you be? Implications of incorrect utility function specification for welfare measurement in choice experiments”. Journal of Environmental Economics and Management, 62(1), 111-121.

Van Cranenburgh S., Wang S., Vij A., Pereira F., and Walker J. (2021). “Choice modelling in the age of machine learning”. arXiv preprint arXiv:2101.11948.

Wang S., Mo B., and Zhao J. (2020a). “Deep neural networks for choice analysis: Architecture design with alternative-specific utility functions”. Transportation Research Part C: Emerging Technologies, 112, 234-251.

Wang S., Wang Q., and Zhao J. (2020b). “Deep neural networks for choice analysis: Extracting complete economic information for interpretation”. Transportation Research Part C: Emerging Technologies, 118, 102701.