A new flexible and interpretable choice model with monotonicity constraints, non-linearity, and taste heterogeneity

Eui-Jin Kim*[1], Prateek Bansal[2]✉

[1] Assistant Professor, Department of Transportation System Engineering, Ajou University, South Korea
[2] Assistant Professor, Department of Civil and Environmental Engineering, National University of Singapore, Singapore (prateekb@nus.edu.sg)

SHORT SUMMARY

This study proposes a flexible and interpretable discrete choice model (DCM) capturing key behavioural mechanisms simultaneously: (i) interactions between alternative-specific and individual-specific attributes (e.g., taste heterogeneity), (ii) interactions between alternative-specific attributes, (iii) inherent non-linear utility of alternative-specific attributes (e.g., diminishing marginal utility of travel cost). Deep neural networks (DNNs) have been considered as candidates to flexibly capture these mechanisms, but they fail to provide trustworthy and explainable economic information (i.e., interpretability) obeying domain-specific knowledge (e.g., decrease in utility of travel mode due to an increase in its travel cost). We propose a DCM based on a lattice network (LN) that efficiently imposes attribute-specific monotonicity constraints in the utility specification while ensuring the trustworthy interpretation of DNNs. The proposed LN-based DCM is benchmarked against DNN in a Monte Carlo study. The results show that it outperforms even the parametric DCM in terms of interpretability while slightly underperforming the DNN in terms of predictability.

Keywords: discrete choice model; monotonicity; deep neural network; lattice network; interpretability.

1. INTRODUCTION

In discrete choice models (DCMs), correctly specifying the systematic utility is critical to achieving good predictability and interpretability. Interpretability indicates the extent to which it provides trustworthy and explainable economic information at an individual level. Ensuring the monotonicity of the utility relative to a subset of alternative-specific attributes is crucial to maintain interpretability. For instance, the utility should monotonically decrease with the increase in cost in most situations.

Traditional parametric DCMs rely on linear-in-parameter utility specifications, with hand-crafted interactions between attributes. Such models are appealing due to the ease of associating the meaning with parameter estimates. However, misspecifications of parametric utility not only result in poor prediction accuracy, but also biased parameter estimates for interaction effects, leading to counterintuitive willingness to pay (WTP) estimates (e.g., negative WTP to save travel time).

To address the issues of parametric DCMs, researchers have adopted deep neural networks (DNNs) (Cranenburgh et al., 2022). The DNNs improve the DCM's predictability by considering complex non-linear and interaction effects of attributes, and WTP and elasticity estimates can also be extracted (Wang, Wang, et al., 2020). While ensuring monotonicity requirements is challenging in DCM-DNN, ignoring it might lead to counterintuitive interpretations in certain attribute domains (Wang, Wang, et al., 2020).

This study contributes with a theory-constrained DCM where the systematic utility is specified using a lattice network (DCM-LN henceforth). First, the lattice network (LN) segments the input space into grids or cells. The input attribute vector is transformed into a vector of interpolation weights (i.e., model parameters to be estimated) over the vertices of the cell that represents the input space. Second, the function value is obtained as a linear transformation of the first step's interpolation weights (Gupta et al., 2016). While the linear transformation in the second step leads to easy-to-implement theoretical conditions for monotonicity, the transformation of the input attribute vector into the first step captures non-linear effects and interactions of attributes. We also add a calibration layer before and after the lattice network to improve the ability of DCM-LN to capture non-linearities in attribute-specific effects, obviating the need to create a fine-grained lattice that requires much more model parameters. DCM-LN can thus simultaneously infer underlying non-linear effects of all alternative- and individual-specific attributes and interactions between them while achieving the monotonic effect of a subset of attributes for every individual.

Figure 1 symbolically benchmarks the systematic utility of the DCM-LN against the traditional DCM with linear utility specification without interactions (DCM-linear) and DCM-DNN at a population and an individual level. Figure 1(a) shows that the overly complex DCM-DNN model represents abrupt changes in marginal utility and even incorrect signs. Such behavioural irrationalities are even worse at an individual level, as indicated by potentially higher heterogeneity (see Figure 1b). On the other hand, the overly simplified DCM-linear model causes serious bias in the marginal utility estimates. In contrast, the DCM-LN can recover true marginal utilities over the domain of input space at an individual level since the monotonicity constraints prevent the incorrect sign of the attribute effects at the individual level; thus, the population-level effect is naturally corrected.

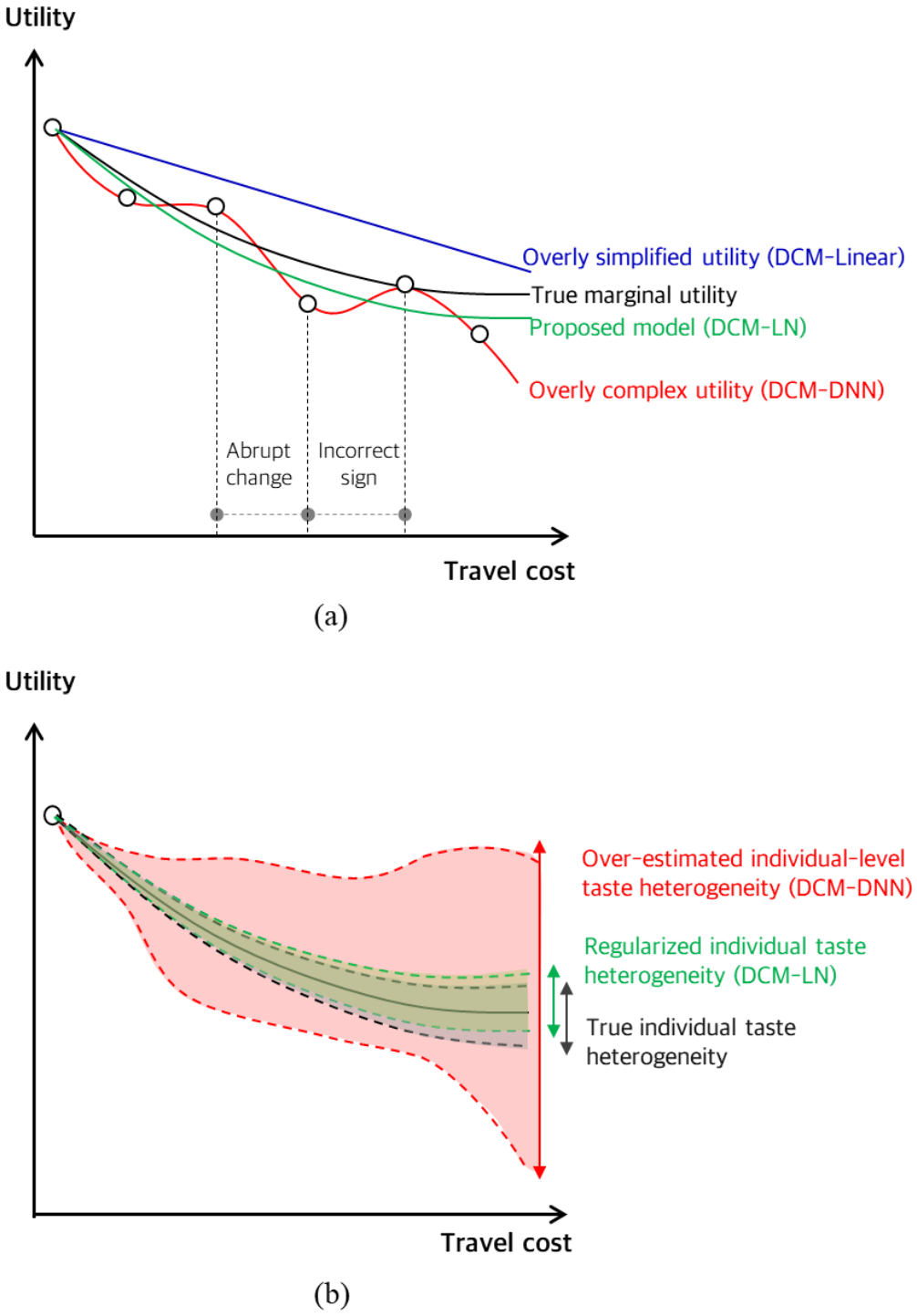

Figure 1. Symbolic benchmarking of the proposed model against existing DCM models

2. METHODOLOGY

Formulation

The indirect utility for individual $n \in \{1, \dots, N\}$ for alternative $o \in \{1, \dots, O\}$ is a sum of systematic utility (V_{on}) and error term (ε_{on}) as denoted in **Equation (1),** and the V_{on} can be represented as a function $\boldsymbol{X}_{on}$, with parameters $\boldsymbol{\theta}$.

$$U_{on} = V_{on} + \varepsilon_{on} = F_{on}(\boldsymbol{X}_{on}; \boldsymbol{\theta}) + \varepsilon_{on} \tag{1}$$

where $\boldsymbol{X}_{on} = (\boldsymbol{x}_{on}, \boldsymbol{z}_n)$ is a vector of input attributes, such that $\boldsymbol{x}_{on} \in R^{M-Q}$ is a vector of alternative-specific attributes and the $\boldsymbol{z}_n \in R^Q$ is a vector of individual-specific attributes. If the ε_{on} is assumed to be identically and independently Gumbel-distributed, the choice probability for the alternative *o* takes the form of the *Softmax* function as in **Equation (2)**.

$$P_{on} = e^{F_{on}(\boldsymbol{X}_{on};\boldsymbol{\theta})} \Big/ \sum_{j=1}^{O} e^{F_{jn}(\boldsymbol{X}_{jn};\boldsymbol{\theta})} \tag{2}$$

Based on the *Softmax* form of choice probability, the DCMs can be estimated by standard empirical risk minimization as in **Equation (3)**.

$$\boldsymbol{\theta}^* = arg \min_{\boldsymbol{\theta}} \sum_{n=1}^{N} \mathcal{L}(\boldsymbol{y}_n, \boldsymbol{P}_{on}) \tag{3}$$

where $\mathcal{L}$ is the standard cross-entropy loss function and $\boldsymbol{y}_n$ is the choice made by individual n. The DCM-Linear assume the F_{on} as a linear function F_{on}^{Li} with $\boldsymbol{\theta}^{Li}$ that is a vector of coefficients for $\boldsymbol{X}_{on}$. The $\boldsymbol{\theta}^{Li}$ directly relates to the main effects $\boldsymbol{\beta}$ as in **Equation (4).**

$$F_{on}^{Li}(\boldsymbol{X}_{on}; \boldsymbol{\theta}^{Li}) = \boldsymbol{\beta}^T \boldsymbol{X}_{on} \tag{4}$$

The DNN represent the F_{on}^{DNN} by multiple neurons in the multiple hidden layers ($\boldsymbol{h}$) as in **Equation (5)**.

$$F_{on}^{DNN}(\boldsymbol{X}_{on}; \boldsymbol{\theta}^{DNN}) = \boldsymbol{A}_{\boldsymbol{on}}^{\boldsymbol{T}}(\boldsymbol{h}_{\boldsymbol{H}} \circ \dots \circ \boldsymbol{h}_{\boldsymbol{2}} \circ \boldsymbol{h}_{\boldsymbol{1}})(\boldsymbol{X}_{on}) \tag{5}$$

where *H* is the number of layers in the DNN, and $\boldsymbol{A}_{\boldsymbol{on}}^{\boldsymbol{T}}$ is the last layer before the Softmax function to make the alternative-specific utility. $\boldsymbol{\theta}^{DNN}$ are the weights (i.e., parameters) connecting neurons and hidden layers.

Empirical studies have shown that F_{on}^{DNN} minimizes the empirical risk with overly complex models (i.e., over-estimated interactions) (Wang, Mo, et al., 2020). To address this issue, this study proposes a flexible but constrained form of LN-based utility function as in **Equation (6).**

$$F_{on}^{LN}(\boldsymbol{X}_{on};\boldsymbol{\theta}^{LN}) = \sum_{m=1}^{M} \boldsymbol{g_m}(\boldsymbol{X}_{on}[m]) + \sum_{m'\neq m} \boldsymbol{r}^{1}_{m,m'}(\boldsymbol{g_m}(\boldsymbol{X}_{on}[m]), \boldsymbol{g_{m'}}(\boldsymbol{X}_{on}[m'])) + \cdots + \sum_{m'\neq m} \boldsymbol{r}^{M}_{m,\ldots,m'}(\boldsymbol{g_m}(\boldsymbol{X}_{on}[m]), \ldots, \boldsymbol{g_{m'}}(\boldsymbol{X}_{on}[m'])) \quad (6)$$

where $\boldsymbol{g_m}$ denotes an attribute-specific utility function that capture the inherent non-linear effect and $\boldsymbol{X}_{on}[m]$ is m-th attribute of $\boldsymbol{X}_{on}$. The $\boldsymbol{r}^{1}_{m,m'}$ indicates the first order interaction between the non-linear effects of m-th and m'-th attributes, and the $\boldsymbol{r}^{M}_{m,\ldots,m'}$ captures the M-th order interactions. To ensure the trustworthy attribute-specific effect, we need to impose partial monotonicity constraints on $\boldsymbol{g_m}$ and $\boldsymbol{r}^{M}_{m,\ldots,m'}$ at individual level.

Lattice network

The key requirement for trustworthy attribute-specific effect is the partial monotonicity of utility function relative to a subset of attributes. For example, increase in travel cost never increases the utility of travel mode if all other attributes are unchanged, regardless of the level of travel cost and the individual attributes. The monotonicity constraints can be implemented by restricting a sum of interaction effects , which requires considering several inequality constraints during training. The LN captures the attribute-specific non-linear effect as segmented effects for each cell (i.e., piecewise linear effect) in the lattice and the interactions of these non-linear effects using multilinear-interpolation. Such combination of piecewise linear functions and multi-linear interpolation enable LN to drastically reduce the number of inequality constraints to be evaluated for the monotonicity constraints (Gupta et al., 2016). Figure 2 shows the LN framework consisting of the calibrator layer and lattice layer.

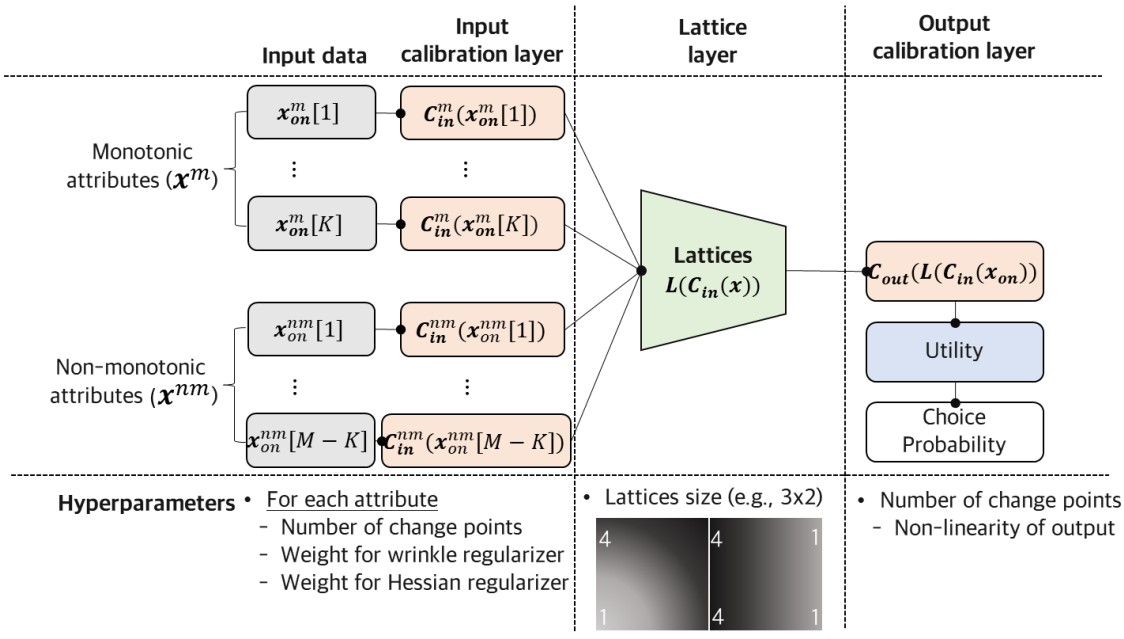

Figure 2. Lattice network framework

The input calibration layers $\boldsymbol{C_{in}}$ in **Figure 2** implements the *K* attribute-specific transformation to capture the non-linear effect before the lattice layer, using one-dimensional monotonic piecewise linear function. These *K* transformation functions are estimated jointly with the lattice in the training. **Figure 3** illustrates the examples of transformation function in the calibration layer. The only hyperparameter for *k*-th attribute in the calibration layer is the number of changing points CP_k if we set the equally distanced cells.

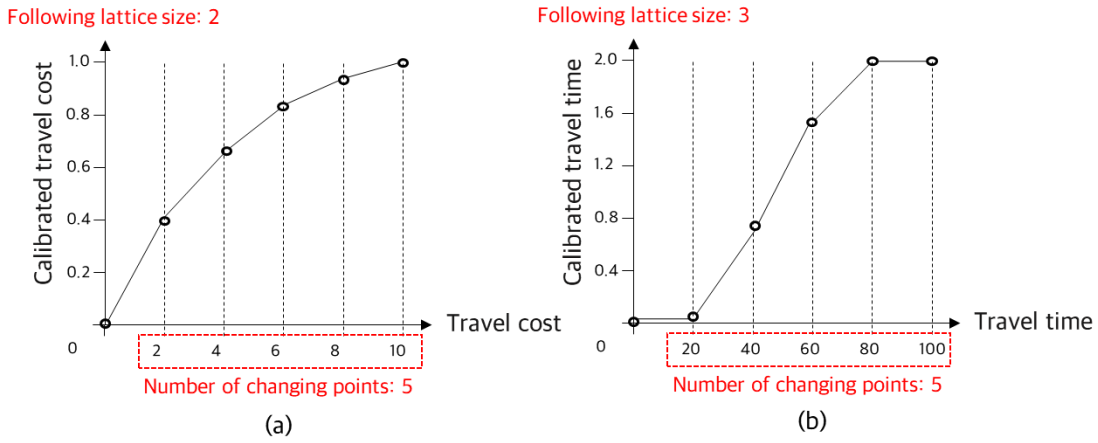

Figure 3. Examples of attribute-wise non-linear transformation through the calibration layer.

For the input attributes, $\boldsymbol{X}_{on} \in R^M$, we define the lattice size S_k for each attribute dimension, which is the number of vertices along the *k*-th attribute dimension. Then, the lattice can be represented by $S = S_1 \times S_2 \times \ldots \times S_K$ parameters and spans the $(S_1 - 1) \times (S_2 - 2) \times \ldots \times (S_K - 1)$ hyper-rectangle. The lattice estimates the value of function $\boldsymbol{L}(\boldsymbol{C_{in}}(\boldsymbol{x}))$ by *S* parameters that is the value of function at each vertex. The larger lattice size can represent more flexible utility function. However, even if the lattice size is two for an attribute, the non-linear effect can be captured by an input calibration layer before the lattice. $\boldsymbol{C_{out}}(\boldsymbol{L}(\boldsymbol{C_{in}}(\boldsymbol{x_{on}}))$ in **Figure 2** estimates the $F_{on}^{LN}(\boldsymbol{X}_{on}; \boldsymbol{\theta}^{LN})$ in **Equation (6)**, and $\boldsymbol{\theta}^{LN}$ consists of (i) the slopes of piecewise linear function in the attribute-specific calibration layers $\boldsymbol{\theta}_{inCal}^{LN}$ (ii) *S* parameters representing the value of function in the vertices $\boldsymbol{\theta}_{Lat}^{LN}$, and (iii) the slope of piecewise linear function in the output calibration layer $\boldsymbol{\theta}_{outCal}^{LN}$.

For the discrete choice datasets consisting of input attributes $\boldsymbol{X}_{on}$ and output choices $\boldsymbol{y}_n$, the objective of the training LN is to estimate the $\boldsymbol{\theta}_{Lat}^{LN}$ while ensuring the monotonicity constraints. For the *k*-th attributes (x_{on}^m[k]), the increasing monotonicity is ensured if $\theta_{Lat,s}^{LN} > \theta_{Lat,r}^{LN}$ for all adjacent vertices *s* and *r* along the *k*-th attribute dimension. Similarly, the monotonicity constraints are also imposed on the attribute-specific input calibration layer. For the parameters of calibration layer for *k*-th monotonic attributes, $\theta_{inCal}^{LN}[k]$ (i.e., the slopes of piecewise linear function in each segment), $\theta_{inCal,u}^{LN}[k] > \theta_{inCal,v}^{LN}[k]$ should be maintained for all adjacent u and v, to make the $C_{in}(x_{on}^m)$ be a piecewise monotonic linear function. With these two levels of inequality constraints, the LN is estimated using a structural risk minimization. Then, the updated parameters are projected to ensure their monotonic constraints. The estimation of the $\boldsymbol{\theta}^{LN}$ is formulated as **in Equations (7-9)**.

$$F_{on}^{LN}(\boldsymbol{X}_{on};\boldsymbol{\theta}^{LN}) = \boldsymbol{C_{out}}\left(\boldsymbol{L}(\boldsymbol{C_{in}}\left(\boldsymbol{x_{on}};\boldsymbol{\theta}_{inCal}^{LN}\right);\boldsymbol{\theta}_{Lat}^{LN}\right);\boldsymbol{\theta}_{outCal}^{LN}) \tag{7}$$

$$P_{on}^{LN} = e^{F_{on}^{LN}(\boldsymbol{X}_{on};\boldsymbol{\theta}^{LN})} \Big/ \sum_{o=1}^{O} e^{F_{on}^{LN}(\boldsymbol{X}_{on};\boldsymbol{\theta}^{LN})} \tag{8}$$

$$\begin{aligned} &\arg\min_{\boldsymbol{\theta}^{LN}} \sum_{n=1}^{N} \mathcal{L}(y_n, P_{on}^{LN}) + R\left(\boldsymbol{\theta}_{inCal}^{LN}\right) \\ &s.t\ \boldsymbol{A\theta}_{Lat}^{LN} \le 0, \boldsymbol{B\theta}_{inCal}^{LN} \le 0, and\ \boldsymbol{C\theta}_{outCal}^{LN} \le 0 \end{aligned} \tag{9}$$

$R(\boldsymbol{\theta}_{nCal}^{LN})$ is the regularization for input calibration layers (i.e., the wrinkle and Hessian regularizer). The matrix $\boldsymbol{A}$ is represents the inequality constraints for $S(=2^K)$ parameters, and partial monotonicity is considered through the matrix $\boldsymbol{A}$. The matrix $\boldsymbol{B}$ and $\boldsymbol{C}$ play a similar role in implementing the partial monotonicity constraints in the input and output calibration layers, respectively. Details on the efficient optimization strategies for **Equation (9)** can be referred to Gupta et al. (2016).

This study adopts the individual conditional expectation (ICE)(Goldstein et al., 2015) as post-analysis tools for DNN and LN to explain the attribute-specific effect (i.e., utility function) at individual level. Readers can refer to more details of ICE and its pros and cons in the Molnar (2018).

3. RESULTS AND DISCUSSION

Simulation study

The data generating process (DGP) for binary choice (Han et al., 2022) are defined as follows. The three individual attributes – income (IN), full-time job (FUL), and flexible commuting (FLX) create systematic taste heterogeneity for the effects of two alternative-specific attributes – travel time (TT) and waiting time (WT). The IN is a categorical variable with 10 intervals, while the FUL and FLX are dummy variables. We also define the crowding (CR) and its interaction with TT, and inherent non-linear utility of TC. **Equation (10)** denotes the true systematic utility of individual n for alternative j.

$$\begin{aligned} V_{nj} = &-0.1 - \mathbf{8}\cdot\sqrt{\boldsymbol{TC_{nj}}} - \mathbf{2.0}\times\boldsymbol{CR_{nj}} + \\ &\begin{pmatrix} -0.1 - 0.5\times IN_n - 0.1\times FUL_n + 0.05\times FLX_n - \mathbf{0.02}\times\boldsymbol{CR_{nj}} \\ -0.2\times IN_n\times FUL_n + 0.05\times IN_n\times FLX_n + 0.1\times FUL_n\times FLX_n \end{pmatrix}\times TT_{nj} + \\ &\begin{pmatrix} -0.2 - 0.8\times IN_n - 0.3\times FUL_n + 0.1\times FLX_n \\ -0.3\times IN_n\times FUL_n + 0.08\times IN_n\times FLX_n + 0.3\times FUL_n\times FLX_n \end{pmatrix}\times WT_{nj} \end{aligned} \tag{10}$$

Benchmark models

The DCM-DNN and DCM-Linear models are used as benchmark models. The DCM-Linear considers the true first-order interactions between the alternative and individual attributes (e.g., $FLX_n \times TT_{nj}$, $FUL_n \times TT_{nj}$) while ignoring second-order interactions (e.g., $FUL_n \times FLX_n \times TT_{nj}$), interactions between alternative attributes (e.g., $TT_{nj} \times CR_{nj}$), and inherent non-linearity (e.g., $8\sqrt{TC_{nj}}$). For the DCM-DNN and DCM-LN, we do not provide any information for the

DGP and input all the alternative and individual attributes. The DCM-LN only uses the prior knowledge in which TT, WT, CR, and TC monotonically decrease the utility at individual level.

Evaluation metrics for interpretability and predictability

This study evaluates the interpretability by comparing the true and estimated WTP. We compare the true and estimated distribution of value-of-time (VOT) and value-of-waiting-time (VOWT) to evaluate the capability of capturing individual taste heterogeneity. We define 40 (=10×2×2) individual groups by IN, FUL, and FLX. For DCM-DNN and DCM-LN, VOT for each individual group is calculated by aggregating ICE along all levels of the attribute value and each individual group. We examine the distribution of estimated VOT and VOWT using five quantile values: 1%, 25%, 50% (median), 75%, and 99%. Then, the accuracy of the estimated VOT and VOWT for 40 individual groups is evaluated by the root mean squared error (RMSE) and mean absolute percentage error (MAPE). The predictability is evaluated by accuracy for binary choice.

Evaluation results

Table 1 summarizes the evaluation results for interpretability and predictability. All the estimates are obtained from 50 synthetic datasets, and their mean and standard deviations are calculated. We examine the VOT and VOWT distributions using five quantiles value. The evaluation results provide four interesting findings. First, the predictability of the DCM-Linear is significantly worse than the DCM-DNN and DCM-LN, and its interpretability is worse than the DCM-LN. These results clearly show how the misspecification of utility function dramatically reduces the predictive performance and the trustworthiness of behavioural interpretation. The main cause may be the large discrepancy between the linear and non-linear utility functions for TC, which dramatically impacts WTP estimates. Second**,** DCM-DNN shows the best predictability and the worst interpretability, indicating that the trade-off relationship still holds for more complex functions. The interpretability of DCM-DNN is even lower than those of DCM-Linear. These results imply that the overly complex function fitted by DCM-DNN does not consider trustworthiness during training. Third**,** DCM-LN highly outperforms the DCM-Linear and DCM-DNN in terms of interpretability. It shows the best performance for all distribution and individual group values. In terms of predictive performance, DCM-LN highly outperforms the DCM-Linear but it is slightly outperformed by the DCM-DNN. Considering the balanced interpretability and predictability performance, the DCM-LN is the best model. Forth**,** both DCM-Linear and DCM-DNN estimate the negative VOT and VOWT for some individual groups, which substantially decreases the trustworthiness of the model's interpretation. In comparison, the DCM-LN that ensures the individual-level monotonicity does not suffer such misidentification and provides slightly low but consistent WTP estimates.

Table 1. Interpretability and predictability evaluation.

Parameter		True		MNL		DCM-DNN		DCM-LN	
		Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
Interpretability: recovery of distribution	*VOT (Median)*	0.284	0.014	0.126	0.019	0.075	0.105	**0.188**	0.080
	VOT (1%)	0.142	0.010	-0.026	0.029	-0.012	0.281	**0.093**	0.063
	VOT (25%)	0.216	0.013	0.066	0.021	0.040	0.085	**0.135**	0.072

	VOT (75%)	0.404	0.024	0.200	0.013	0.123	0.172	**0.257**	0.089
	VOT (99%)	0.675	0.027	0.372	0.024	0.222	0.214	**0.456**	0.105
	VOWT (Median)	0.480	0.019	0.258	0.148	0.146	0.210	**0.322**	0.134
	VOWT (1%)	0.252	0.011	-0.068	0.124	-0.114	0.797	**0.153**	0.082
	VOWT (25%)	0.372	0.017	0.118	0.141	0.086	0.159	**0.244**	0.127
	VOWT (75%)	0.779	0.033	0.414	0.175	0.233	0.273	**0.472**	0.181
	VOWT (99%)	1.236	0.049	0.719	0.288	0.402	0.560	**0.892**	0.263
Interpretability: recovery of individual groups' value	*VOT (MAPE)*			0.630	0.044	0.802	0.329	**0.351**	0.103
	VOT (RMSE)			0.193	0.012	0.272	0.102	**0.129**	0.030
	VOWT (MAPE)			0.598	0.195	0.846	0.459	**0.359**	0.112
	VOWT (RMSE)			0.348	0.092	0.546	0.259	**0.243**	0.063
Predictability:	*Training accuracy*			0.552	0.006	**0.775**	0.010	0.741	0.018
	Test accuracy			0.546	0.013	**0.716**	0.014	0.697	0.016

Figures 4 and 5 support the findings derived from Table 1 and reveal some insightful patterns**.** First, all models underestimate the VOT and VOWT for most individual groups, except for some peak points caused by misidentification of the interaction effects, but the extent of the underestimation of DCM-LN is smaller than the other models. Second, Figure 5 shows that DCM-LN approximates the non-linear effect much better than DCM-DNN at both population and individual levels. One major issue of DCM-DNN is that it provides almost zero or positive marginal utility of TC for some levels of TC, which may lead to unreasonably high VOT or VOWT estimates, as in the peak of red lines in Figure 4. In contrast, DCM-LN could prevent such misspecification using its monotonicity constraints, also representing relatively stable WTP patterns in Figure 4**.** Third, the estimated alternative-specific utility functions in DCM-Linear are far from the true DGP. This result implies that there is a need to go beyond hand-crafted utility specifications if predictability is of interest.

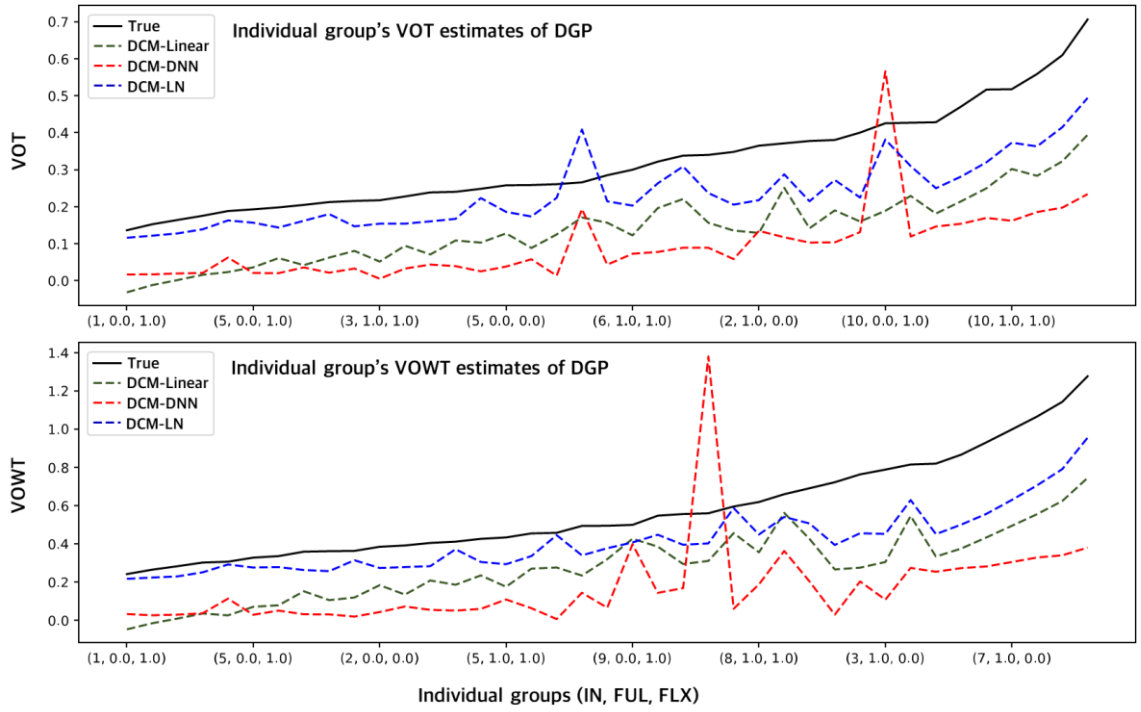

Figure 4. VOT and VOWT estimates for 40 individual groups in DGP.

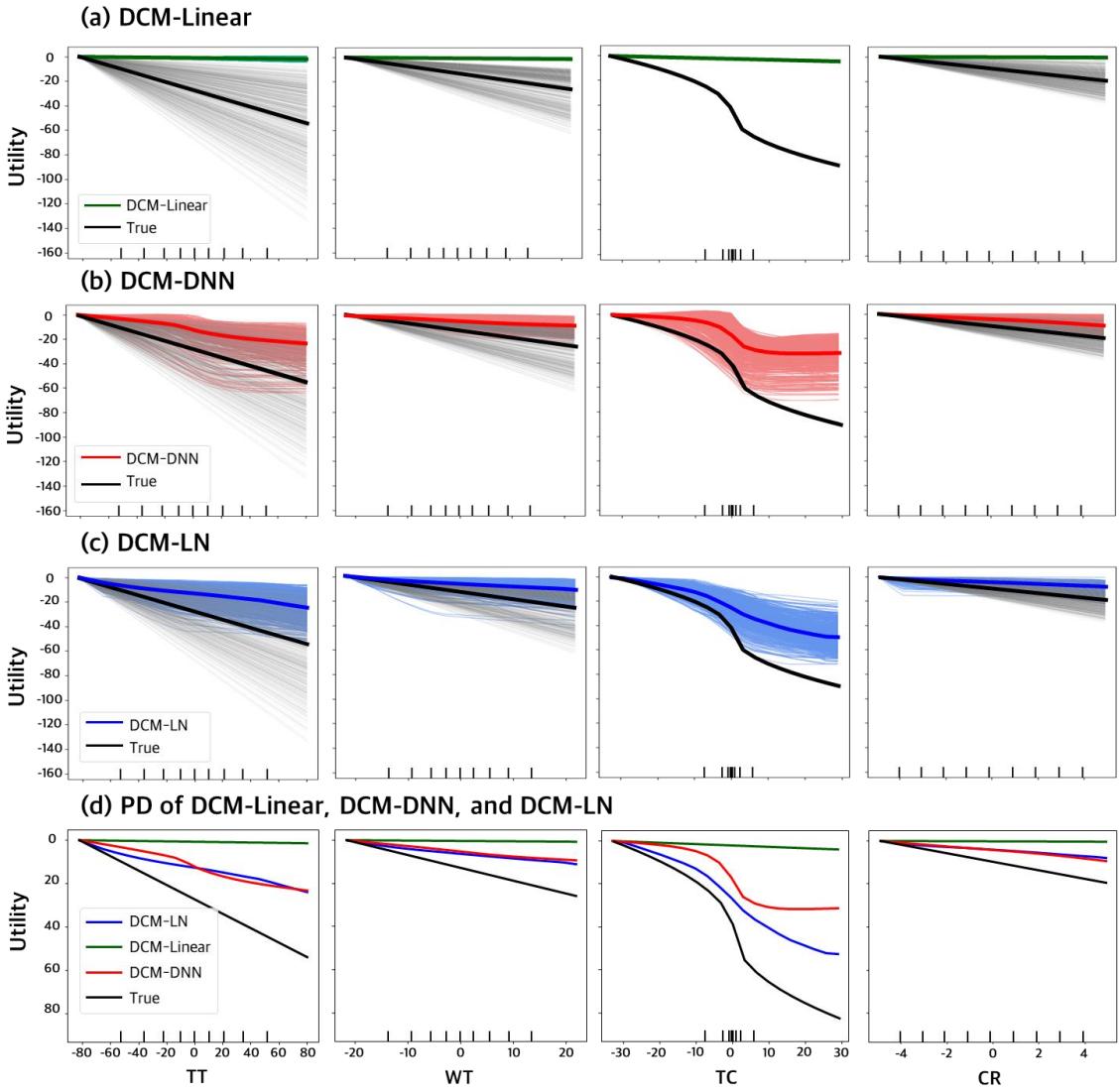

Figure 5. Attribute-specific utility functions estimated by (a) DCM-Linear, (b) DCM-DNN, and (c) DCM-LN at the individual and (d) population-level distribution (PD) of all three models.

4. CONCLUSIONS

In summary, we customize the lattice networks and introduce their first application to the DCMs to achieve a flexible utility specification while maintaining interpretability by imposing theory-driven constraints at an individual level. We benchmark the performance of DCM-LN against DCM-DNN and a parametric DCM (i.e., DCM-Linear) in a simulation study in terms of predictive accuracy and recovery of underlying marginal utility and individual-level WTP values across input space.

The evaluation results show that the DCM-LN highly outperforms the DCM-Linear and DCM-DNN in terms of interpretability, which is measured by the capability to recover the true utility and WTP of the simulation dataset. In contrast, the predictability of DCM-Linear is only slightly outperformed by the DCM-DNN, indicating that its capability to capture the complex

interactions in DGP remains intact after imposing monotonicity constraints. This balanced performance of DCM-LN is quite promising because it suggests that the DCM-LN approximates the true utility function during the training, rather than arbitrary functions that maximize the predictability as DCM-DNN does.

The work for evaluating the DCM-LN on real data is ongoing for further verification. Also, we are incorporating other behavioral mechanisms (i.e., soft constraints) into the DCM-LN, such as the non-compensatory decision rules (e.g., attribute cut-off and attribute non-attendance) and asymmetric marginal utility (e.g., prospect theory).

REFERENCES

Goldstein, A., Kapelner, A., Bleich, J., & Pitkin, E. (2015). Peeking Inside the Black Box: Visualizing Statistical Learning With Plots of Individual Conditional Expectation. *Journal of Computational and Graphical Statistics*, *24*(1), 44–65. https://doi.org/10.1080/10618600.2014.907095

Gupta, M., Cotter, A., Pfeifer, J., Voevodski, K., Canini, K., Mangylov, A., Moczydlowski, W., & van Esbroeck, A. (2016). Monotonic Calibrated Interpolated Look-Up Tables. *Journal of Machine Learning Research*, *17*, 1–47. https://doi.org/10.5555/2946645.3007062

Han, Y., Pereira, F. C., Ben-Akiva, M., & Zegras, C. (2022). A neural-embedded discrete choice model: Learning taste representation with strengthened interpretability. *Transportation Research Part B: Methodological*, *163*, 166–186. https://doi.org/10.1016/j.trb.2022.07.001

Molnar, C. (2018). *A guide for making black box models explainable*. Https://Christophm.Github.Io/Interpretable-Ml-Book.

van Cranenburgh, S., Wang, S., Vij, A., Pereira, F., & Walker, J. (2022). Choice modelling in the age of machine learning - Discussion paper. *Journal of Choice Modelling*, *42*(December 2021), 100340. https://doi.org/10.1016/j.jocm.2021.100340

Wang, S., Mo, B., & Zhao, J. (2020). Deep neural networks for choice analysis: Architecture design with alternative-specific utility functions. *Transportation Research Part C: Emerging Technologies*, *112*(February), 234–251. https://doi.org/10.1016/j.trc.2020.01.012

Wang, S., Wang, Q., & Zhao, J. (2020). Deep neural networks for choice analysis: Extracting complete economic information for interpretation. *Transportation Research Part C: Emerging Technologies*, *118*(February), 102701. https://doi.org/10.1016/j.trc.2020.102701