



Enhancing discrete choice models with representation learning

Brian Sifringer^a, Virginie Lurkin^{b,*}, Alexandre Alahi^a

^a Visual Intelligence for Transportation Laboratory (VITA), École Polytechnique Fédérale de Lausanne (EPFL), Switzerland

^b Department of Industrial Engineering & Innovation Sciences, Eindhoven University of Technology, Eindhoven 5600MB, The Netherlands



ARTICLE INFO

Article history:

Received 21 December 2018

Revised 17 June 2020

Accepted 24 August 2020

Available online 9 September 2020

Keywords:

Discrete choice models

Neural networks

Utility specification

Machine learning

Deep learning

ABSTRACT

In discrete choice modeling (DCM), model misspecifications may lead to limited predictability and biased parameter estimates. In this paper, we propose a new approach for estimating choice models in which we divide the systematic part of the utility specification into (i) a knowledge-driven part, and (ii) a data-driven one, which learns a new representation from available explanatory variables. Our formulation increases the predictive power of standard DCM without sacrificing their interpretability. We show the effectiveness of our formulation by augmenting the utility specification of the Multinomial Logit (MNL) and the Nested Logit (NL) models with a new non-linear representation arising from a Neural Network (NN), leading to new choice models referred to as the *Learning Multinomial Logit* (L-MNL) and *Learning Nested Logit* (L-NL) models. Using multiple publicly available datasets based on revealed and stated preferences, we show that our models outperform the traditional ones, both in terms of predictive performance and accuracy in parameter estimation. All source code of the models are shared to promote open science.

© 2020 Elsevier Ltd. All rights reserved.

1. Introduction

Discrete Choice Models (DCM) have emerged as a powerful theoretical framework for studying choices made by humans. The goal of these models is to *predict* the choice outcome among a given set of discrete alternatives (e.g., choice of walking as a transportation mode, rather than taking the car or the bus), while *understanding* the behavioral process that led to the specific choice. For many years, the Multinomial Logit model (MNL), based on a simple parametric utility specification, has provided the foundation for the analysis of discrete choice. Despite its oversimplified assumptions regarding the actual decision-making process, MNL is still commonly used in practice since it enables a high level of interpretability.

Interpretability is critical for researchers and practitioners. For instance, parametric specifications based on linear or logarithmic functions allow for straightforward derivation of the value-of-time (VOT), i.e., the marginal rate of substitution between time and cost, that constitutes a highly relevant measure in a wide range of public transport policy. However, interpretability gain from using an MNL model with such simple parametric specifications often leads to a sacrifice in the predictive power of the model. Indeed, MNL does not allow to incorporate individual heterogeneity explicitly in the empirical investigation, and the assumed simple parametric model cannot adequately capture the full underlying structure of the data. Several prior studies, within the discrete choice literature, have shown that incorrectly assuming a parametric util-

* Corresponding author.

E-mail address: v.j.c.lurkin@tue.nl (V. Lurkin).

ity specification can cause severe bias in parameter estimates and post-estimation indicators (e.g., Torres et al., 2011; Van Der Pol et al., 2014).

More advanced utility specifications, embedded into more complex models, would allow for a better fit of the data and, ultimately, a better prediction. In this regard, utility specifications based on Box-Cox transformation, piecewise-linear, exponential, or non-parametric functions have been proposed in the literature (e.g., Schindler et al., 2007; Kneib et al., 2007; Kim et al., 2016), as well as advanced DCM models, such as the Mixed Logit Model (MLM) or the Latent Class Model (LCM) (e.g., Shen, 2009; Xiong and Mannering, 2013; Vij et al., 2013; Kim et al., 2016). However, the main limitation of these choice models is that the standard procedure for estimating the parameters' values is to assume that the model specification is known *a priori*, while in practice determining the utility specification for a particular application remains a difficult task.

Within the machine learning community, Neural Networks (NN) masters the field of representation learning. They have become the go-to solutions, as their superior prediction performance has been repeatedly demonstrated, including in transportation applications (e.g., Omrani, 2015; Hagenauer and Helbich, 2017; Pekel and Soner Kara, 2017; Nam et al., 2017; Pirra and Diana, 2018; Chang et al., 2019; Zhou et al., 2019). Unlike discrete choice models, NN requires essentially no *a priori* beliefs about the nature of the true underlying relationships among variables. However, gaining in prediction accuracy often comes at the cost of losing interpretability, explaining why the discrete choice community often considers them as a black-box.

In this work, we propose a new approach to discrete choice modeling by integrating representation learning in the formulation. We propose to divide the systematic part of the utility specification into (i) a knowledge-driven (interpretable) part and (ii) a data-driven (representation learning) part, which aims at automatically discovering a good utility specification from available data. Our formulation partially replaces manual utility specification and allows for a better prediction performance without sacrificing interpretability. We demonstrate the effectiveness of our framework by augmenting the utility specification of the MNL and Nested logit models with a new non-linear representation, arising from a neural network. It leads to new choice models, referred to as the *Learning Multinomial Logit* (L-MNL) and *Learning Nested Logit* (L-NL) models, respectively. The source code is made publicly available to ease reproducibility and promote open science¹.

The remainder of the paper is organized as follows: Section 2 is a non-exhaustive overview of the recent attempts to conduct multi-disciplinary research, combining machine learning (ML) and discrete choice modeling. Section 3 gives a brief background and presents details on how we implement MNL with modern neural network libraries. Section 4 describes our proposed L-MNL and L-NL models. Section 5 demonstrates evidence that our learning choice models outperform both existing hybrid neural network and MNL models based on simple parametric utility specifications. Our results on real and synthetic data show that the research community can benefit from a better integration of representation learning techniques and discrete choice theory. In that sense, our research also contributes to bridging the gap between knowledge-driven and data-driven methods.

2. Related work

Discrete choice and machine learning researchers have typically different perspectives and research priorities. The related work in each field is too broad to be covered here exhaustively, and we, therefore, limit ourselves to the most relevant works at the intersection of the two communities.

More than 20 years ago, researchers were already interested in applying data-driven methods such as neural networks in different transportation applications (e.g., Faghri and Hua, 1992; Dougherty, 1995; Shmueli et al., 1996). Over the years, different types of NN, as well as decision trees, have been compared to MNL models (Agrawal and Schorling, 1996; Lee et al., 2018; Zhao et al., 2020), Nested Logit (NL) models (Mohammadian and Miller, 2002; Hensher and Ton, 2000) and more generally to random utility models (Sayed and Razavi, 2000; Cantarella and de Luca, 2005; Paredes et al., 2017) and statistical methods (West et al., 1997; Karlaftis and Vlahogianni, 2011; Iranitalab and Khattak, 2017; Golshani et al., 2018; Brathwaite et al., 2017). However, the literature has mainly focused on comparing the models in terms of prediction power, without discussing the issue of interpretability.

The machine learning community has also generated a tremendous amount of research aiming at predicting, with high accuracy, a variety of choices. Among the most recent works, Pekel and Soner Kara (2017) or Hillel et al. (2020) present comprehensive reviews of publications related to the application of NN or ML to predict travelers' choice in public transportation. Hagenauer and Helbich (2017) compare seven machine learning methods to classify travel mode choice. The methods investigated are MNL and NN, but also Naive Bayes (NB) (Rish et al., 2001), Gradient Boosting Machines (GBM) (Friedman, 2001), Bagging (BAG) (Breiman, 1996), Random Forests (RF) (Breiman, 2001), and Support Vector Machine (SVM) (Cortes and Vapnik, 1995). Pirra and Diana (2018) also propose to use SVM for travel mode choice while (Lhéritier et al., 2018) rely on RF and GBM to predict airline itinerary choice. Not surprisingly, the results exhibited in these papers show outstanding results in the ability of these models to fit the data but do not tackle the issue of their interpretability.

Some recent publications have gone beyond the comparison of the two fields and proposed innovative behavioral studies based on data-driven methods. Two notable examples are the work of Wong et al. (2018), who use a restricted Boltz-

¹ <https://github.com/BSiffringer/EnhancedDCM>.

mann Machine (BM) (Ackley et al., 1985) to represent latent behavior attributes, and the study of van Cranenburgh and Alwosheel (2019), who develop a novel NN based approach to investigate decision rule heterogeneity amongst travelers. Moreover, Wang et al. (2018c) have investigated the numerical extraction of behavioral information such as willingness to pay in a Dense Neural Network (DNN) using elasticity while (Wang et al., 2018b) have investigated the theoretical transition from DCM to DNN and subsequent trade between predictability and interpretability. Recently, the main authors have also developed a new structuring of neural networks, based on alternative specific utility architecture (Wang et al., 2020). Finally, Wong and Farooq (2020) have made use of a NN generative model to capture the joint distribution of multiple discrete-continuous travel data. However, these papers do not have the objective of finding a utility specification that allows high predictability while maintaining straightforward interpretability.

The closest studies relative to our work are in the brand choice literature. Bentz and Merunka (2000) introduce a hybrid model, afterward called the NN-MNL model. Their model involves a two-stage approach that starts with the estimation of a NN model that aims at discovering non-linearity effects in the utility function. Then, if the NN identifies some non-linearities, the specification of the MNL model is modified to include new variables, specially created to account for the discovered non-linear effects. On their dataset, the re-specified MNL model, which includes the discovered non-linear effects, slightly outperforms the basis MNL model. Their work, similar to ours in spirit, aims at achieving better predictive power together with a greater understanding of the influencing factors. However, their approach is sequential, and their neural network is only used beforehand as a diagnostic and specification tool. Again, in the context of brand choice, Hruschka et al. (2002, 2004) and Hruschka (2007) investigate further the potential of neural network based choice models. In Hruschka et al. (2002), the NN-MNL model is compared against homogeneous and heterogeneous versions of linear utility MNL models. Meanwhile, in Hruschka et al. (2004), the comparison is made against two non-linear models, the generalized additive (GAM-MNL) model of Abe (1999) and a flexible functional form based on Taylor series approximations. The authors show that, by being capable of finding non-linear relationships from available data, the neural network based choice models with non-linear specifications outperform the other models in terms of prediction performance. However, the authors are unable to draw any conclusion regarding the significance of the factors with this method. Also, the post estimation analysis is limited to likelihood and choice elasticities. Finally, Hruschka (2007) proposes a Multinomial Probit (MNP) model with a neural network extension to model brand choice. Their model combines the heterogeneity across households with a multi-layer perceptron. Linear and non-linear deterministic utility specifications are considered by specifying the same variables in linear and non-linear terms, hence losing interpretability. The probit assumption also prevents the closed form of the choice probabilities and the model can only be estimated using a Markov Chain Monte Carlo simulation technique.

Our work significantly departs from these previous studies by proposing a flexible and general framework for the specification of the deterministic utility in any DCM model. More specifically, our approach combines, in a single joint optimization, the estimation of a standard DCM model, based on a carefully thought-out utility specification, with a representation learning model that aims at increasing the overall predictive performance. To our knowledge, this is the first formulation that benefits from the predictive power of representation learning techniques while keeping some key parameters interpretable, which allows us to derive insightful post-estimation indicators. In Section 5, we show the effectiveness of our learning choice method by comparing its performance with benchmarking models, including the NN-MNL model of Hruschka et al. (2002, 2004).

3. Background and notations

3.1. Multinomial logit

Discrete choice modeling complies with the Random Utility Maximization (RUM) theory (McFadden, 1974), which postulates that an individual is a rational decision-maker, who aims at maximizing the utility relative to their choice. Utility is a latent construct assumed to be partitioned into two components: a systematic (or deterministic) utility, V_{in} , and a random component, ε_{in} , that captures the uncertainty coming from the impossibility for the modeler to fully capture the choice context. Formally, the utility that individual n associates with alternative i from their choice set C_n is given as²:

$$U_{in} = V_{in} + \varepsilon_{in}, \quad (1)$$

For convenience, we consider the systematic part of the utility to be linear-in-parameter, as it is generally assumed:

$$V_{in} = \sum_d \beta_d \cdot x_{din}, \quad (2)$$

where β are the preference parameters (or estimators) associated with the explanatory variables (or the input features) $\mathbf{x} \in \mathcal{X}$ that describe the observed attributes of the choice alternative (e.g., the price or travel time associated with the mode), and the individual's socio-demographic characteristics (e.g., the individual's level of income or age).

Under the standard MNL assumption that error terms are independently and identically distributed (i.i.d.) and follow an Extreme Value Type I distribution with location parameter zero and the scale parameter 1 (i.e. $\varepsilon_{in} \sim \text{i.i.d. EV}(0, 1)$), the

² Notation convention can be found in Appendix A.

probability for individual n to select choice alternative i is given by

$$P_n(i) = \frac{e^{V_{in}}}{\sum_{j \in C_n} e^{V_{jn}}}. \quad (3)$$

The preference parameters β are typically estimated by maximizing the log-likelihood function given by:

$$\mathcal{L} = \sum_{n=1}^N \sum_{i \in C_n} y_{in} \log [P_n(i)], \quad (4)$$

where y_{in} is the observed choice variable (or true label) and is equal to 1 if the choice of individual n is the alternative i , and to 0 otherwise.

3.2. Implementing MNL as a neural network

A neural network consists of a function mapping the input space x to an output of interest U through several intermediate representations commonly referred to as hidden layers $\mathbf{h}^{(j)}$:

$$\mathbf{U} = \mathbf{h}^{(L)}(\mathbf{q}^{(L-1)}), \quad (5)$$

$$\text{with } \mathbf{q}^{(j)} = \mathbf{h}^{(j)}(\mathbf{q}^{(j-1)}), \quad \forall j = 1, \dots, L, \quad (6)$$

where $\mathbf{q}^{(0)} = \mathbf{x}$ and L is the last representation layer.

We make use of a Convolutional Neural Network (CNN) to retrieve the MNL formulation³ A CNN has weights in the shape of a filter that connects a layer $\mathbf{h}^{(j)}$ to the next by applying a convolution⁴. Therefore, the value of a neuron i in the next layer $(j+1)$ can be written as:

$$h_i^{(j+1)} = g\left(\sum_{k=0}^d h_{(s+i+k)}^{(j)} \beta_k^{(j)} + \alpha_i^{(j)}\right), \quad (7)$$

where $\{\beta_1, \dots, \beta_d\} = \beta$ is the filter of size $(1 \times d)$, s the stride of the convolution, α_i a bias term and $g(\cdot)$ an activation function.

The MNL formulation is retrieved by using a single layer ($L=1$), setting the activation function to identity ($g(x) = x$) and the stride s to d . Doing so, we get the utility functions $\mathbf{V}_n = \{V_{1n}, \dots, V_{In}\}$ as defined in Eq. (2).

Then, the probabilities can be obtained by using a softmax activation layer (Bishop, 1995) defined as:

$$(\sigma(\mathbf{V}_n))_i = \frac{e^{V_{in}}}{\sum_{j \in C_n} e^{V_{jn}}}, \quad (8)$$

which can be identified as Eq. (3) for all probabilities. The output of the network goes through a loss function, in our case categorical cross-entropy (CE) (Shannon, 1948):

$$H_n(\sigma, \mathbf{y}_n) = - \sum_{i \in C_n} y_{in} \log [\sigma_i(\mathbf{V}_n)]. \quad (9)$$

Minimizing (9) is equivalent to maximizing Eq. (4) when summed over all individuals n .

An illustrative example is given in Fig. 1 with :

$$U_{in} = \beta_c \cdot x_{1i} + \beta_t \cdot x_{2i} + \varepsilon_{in}, \quad \forall i \in C, \quad (10)$$

where $x_1 = \text{cost}$ stands for travel cost, and $x_2 = \text{time}$ for travel time. The choice set C is assumed to be the same for all individuals and contains the following mode alternatives: $C = \{\text{Car}, \text{Train}, \text{Swissmetro (SM)}\}$. The filter is made of $\beta = (\beta_t, \beta_c)$, and a stride of $d = 2$ allows to recover each utility specification in the next layer from *cost* and *time* of every alternative.

³ Previous studies, such as Bentz and Merunka (2000), have successfully written MNL with other Neural Network architectures. We have used CNN for its weight-sharing architecture.

⁴ This method is often seen in image processing where a filter with a fix number of weights is applied to an equal amount of inputs by multiplying the terms together and then summing them over to obtain a single new value. A new image is obtained by sliding the filter over all inputs with a set step-size named stride. In our case, the operation is done without padding, i.e., without applying the filter out of the boundaries, effectively reducing the size from one layer (filtering) to the next. In image processing, the correlation formula is the same as a convolution up to a flip in the order of its entries. Without loss of generality, we write the correlation formula in Eq. (7) for readability.

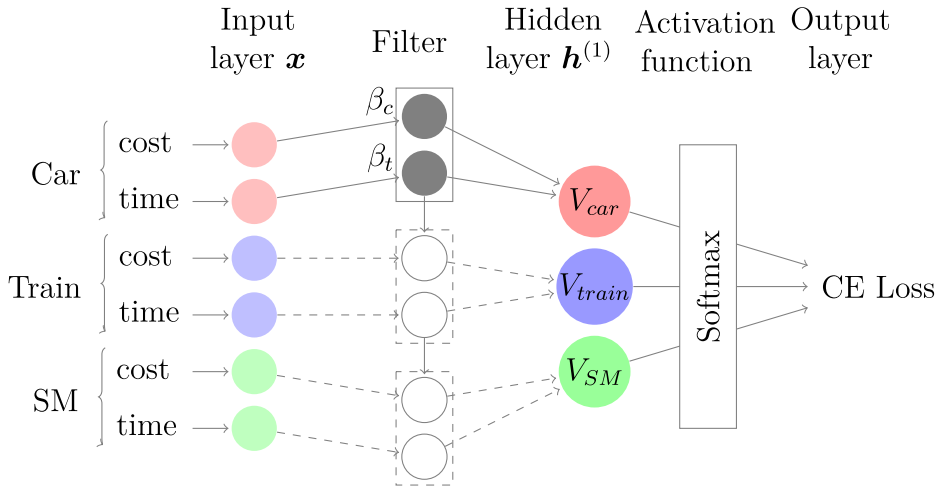


Fig. 1. By aligning inputs by class and convolving with a filter of equivalent shape and stride, we can retrieve linear utility specifications with a single CNN layer. By ending the network with a softmax activation layer and a cross-entropy (CE) loss, we retrieve the same formulation as for the MNL model.

4. Representation learning in discrete choice modeling

4.1. General formulation

As previously mentioned, the standard statistical procedure for estimating the parameter values of choice models is to assume that the true utility specification is known *a priori*. Typically, the most common representation is inspired from linear regression and involves the observed attributes of the choice alternative and the individual's socio-demographic characteristics. However, the influence of these explanatory variables on the utility of a choice alternative is unlikely to be known and to be precisely linear. The danger of an incorrect utility specification remains, therefore, highly present.

In this work, we propose a more flexible and data-driven approach that consists in expressing the systematic part of the utility function into two sub-parts, as follows:

$$V_{in} = f_i(\mathcal{X}_n; \boldsymbol{\beta}) + r_i(\mathcal{Q}_n, \mathbf{w}), \quad (11)$$

where

- $f(\mathcal{X}_n; \boldsymbol{\beta})$ is the knowledge-driven part, assumed interpretable. The function f is defined such that its unknown model parameters $\boldsymbol{\beta}$ are an interpretable combination of the explanatory variables (or input features) \mathcal{X}_n .
- $r(\mathcal{Q}_n, \mathbf{w})$ is the data-driven part, a representation that is learned from a set of explanatory variables (or input features) \mathcal{Q}_n , for which no *a priori* relationship is assumed.

Conditions between \mathcal{X} and \mathcal{Q} to keep f interpretable are described in Section 4.2.

Replacing the systematic utility component in Eq. (1) by its new expression given by Eq. (11), we obtain the following utility expression:

$$\mathbf{U}_n = \mathbf{f}(\mathcal{X}_n; \boldsymbol{\beta}) + \mathbf{r}(\mathcal{Q}_n; \mathbf{w}) + \boldsymbol{\epsilon}_n. \quad (12)$$

We may interpret this equation as a data-driven term finding the residual of a hand-modeled function. He et al. (2016) have shown the impact of residual network (ResNet) in the machine learning community, and its particular architecture has been widely used for increased interpretability by finding the residual of knowledge and equation based modeling (Liao et al., 2018; Wang et al., 2018a; Li et al., 2019), or choice modeling (Wong and Farooq, 2019). Our work differs with the use of two input sets \mathcal{X} and \mathcal{Q} necessary for keeping discrete choice interpretability as we define it in Section 4.2 and further explain in Section 5.

Finally, the above specification may be applied to any discrete choice modeling kernel. However, the following applications will limit themselves to models with Gumbel distributed random terms.

4.2. Modeling

When modeling a utility specification with the formulation in Eq. (12), one must understand the attributes of a parameter related to either \mathcal{X} in \mathbf{f} or \mathcal{Q} in \mathbf{r} . The first assumes to have all the benefits of expert modeling in discrete choice, which includes the availability of the $\boldsymbol{\beta}$ parameters' Hessian and thus their standard deviation approximation. The second produces a new representation of its associated input, which can span from discrete to continuous inputs such as signals and images,

while the hessian is generally not available due to computational complexity. Therefore, when one studies the impact, or elasticity, of a feature t_{in} on alternative i , this readily translates as:

$$\frac{\partial U_{in}}{\partial t_{in}} = \frac{\partial f_{in}}{\partial t_{in}} + \frac{\partial r_{in}}{\partial t_{in}} \quad (13)$$

Bentz and Merunka (2000) and Wang et al. (2018c) have shown that behavioral insight can be captured and studied through the elasticities of a neural network. However, to ensure the same straightforward interpretability of behavioral choice models for a parameter t , we explicitly stipulate that the standard deviation must be available, and the elasticity must not depend on the representation term. In other words, we impose that:

$$\frac{\partial r_{in}}{\partial t_{in}} = 0 \quad (14)$$

Eq. (14) constitutes a critical assumption which ensures that insightful post-estimation indicators can be retrieved. This definition, as well as the general formulation is what sets us apart from old and new literature alike. Indeed, to only cite a few, Hruschka (2007) or Wong et al. (2018) are specific cases of our formulation when both sets are the same, i.e., $\mathcal{X} = \mathcal{Q}$, while Bentz and Merunka (2000), Otsuka and Osogami (2016) and Wang et al. (2018c) are cases where $\mathcal{X} = \emptyset$. All of which do not follow our goal of fixing a biased expert modeling component by leveraging the strength of data-driven methods in an added and non-overlapping representation term.

Moreover, the global theoretical effect of this new term may be derived. When one considers Eq. (12) without the new term, then this value would be found in the random component such that:

$$\bar{\varepsilon}_{in} = r_i(\mathcal{Q}_n; \mathbf{w}) + \varepsilon_{in}. \quad (15)$$

To have unbiased parameter estimates in $\mathbf{f}(\mathcal{X}_n; \boldsymbol{\beta})$, one must avoid correlation between the specification and the random terms known as endogeneity, avoid correlation of the random terms between each alternative and overall avoid utility misspecification. Many methods have been developed to specifically fix for endogeneity, such as Guevara (2015), while others account for correlation between alternatives (see Section 4.4). Our model aims at correcting for underfit due to misspecification and omitted variable bias thanks to data-driven methods. In this sense, the new error term ε_{in} is more likely to fit the aforementioned criteria than $\bar{\varepsilon}_{in}$. The new representation term increases the probability of having unbiased parameters while increasing the prediction rate as it may easily find undiscovered specifications during the expert modeling process. Moreover, our general framework does not exclude the coupling with other methods that aim at fixing the random term, as we will observe in the following sections.

4.3. L-MNL Model formulation

In the specific case where we choose to add a representation term to a Multinomial Logit for Eq. (12), the standard MNL assumptions regarding the distribution of the error term ε_{in} (described in Section 3.1) have to be made.

The likelihood of selecting the choice alternative i for individual n , given the values of the model parameters ($\boldsymbol{\beta}$ and \mathbf{w}) and the influencing factors (\mathcal{X}_n and \mathcal{Q}_n), is therefore naturally expressed as:

$$P_n(i) = \frac{e^{f_i(\mathcal{X}_n; \boldsymbol{\beta}) + r_i(\mathcal{Q}_n; \mathbf{w})}}{\sum_{j \in C_n} e^{f_j(\mathcal{X}_n; \boldsymbol{\beta}) + r_j(\mathcal{Q}_n; \mathbf{w})}}. \quad (16)$$

For our L-MNL model, we choose to use a Dense Neural Network (DNN) as the learning method. More specifically, its representation term r_{in} is the resulting function of a DNN with L layers of H neurons and a single output per utility function:

$$r_{in} = \sum_{k=1}^H \mathbf{w}_{ik}^{(L)} g(\mathbf{q}_n^{(L-1)} \mathbf{w}_k^{(L-1)} + \alpha_k^{(L-1)}) + \alpha_i^{(L)}, \quad (17)$$

where $g(\cdot)$ is the rectifier linear units (ReLU) activation function and $\mathbf{q}_n^{(j)}$ is recurrently defined by:

$$[\mathbf{q}_n^{(j+1)}]_i = \sum_{k=1}^H \mathbf{w}_{ik}^{(j+1)} g(\mathbf{q}_n^{(j)} \mathbf{w}_k^{(j)} + \alpha_k^{(j)}) + \alpha_i^{(j+1)}, \quad (18)$$

with $\mathbf{q}_n^{(0)}$ being the vector of input features \mathcal{Q}_n .

The architecture of our L-MNL hybrid model is shown in Fig. 2, where we see both the linear component, written as a CNN (see Section 3), and the learned representation term, obtained through a DNN. Our hybrid approach is both general and flexible. Other functions of \mathbf{f} and other architectures for \mathbf{r} may be chosen. For more complex DCM frameworks, new assumptions must be made on the error terms, probabilities, and likelihood function. For instance, an extension of our work to the nested logit model is shown in the next section.

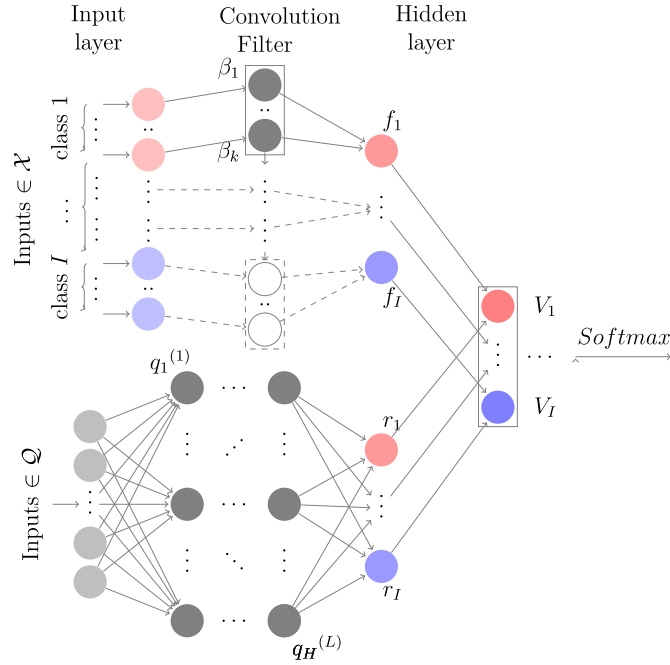


Fig. 2. L-MNL model architecture. On the top, we have the I class generalization of a linear-in-parameter MNL model, as depicted in Fig. 1. At the bottom, we have a deep neural network (i.e., multilayer and fully connected) that enables us to obtain the representation learning term r_i . The terms from each part are added together defining the new systematic function of Eq. (11).

4.4. L-MNL Nested generalization

The Nested Logit formulation has been first introduced by Williams (1977). It relaxes the *i.i.d.* assumption of MNL random terms in the case where alternatives may have correlation among each other in the decision making process. They are gathered together in nests which then affects their final probability such that:

$$P_n(i|C) = P_n(i|m) \cdot P_n(m|C) \\ = \frac{e^{\mu_m V_{in}}}{\sum_{j \in C_m} e^{\mu_m V_{jn}}} \cdot \frac{\exp\left(\frac{\mu}{\mu_m} \ln \sum_{p \in C_m} e^{V_{pm}}\right)}{\sum_{k=1}^M \exp\left(\frac{\mu}{\mu_k} \ln \sum_{p \in C_k} e^{\mu_k V_{pm}}\right)} \quad (19)$$

where M is the total number of nests, μ_l is the factor for nest l and C_l is the set of alternatives belonging to that nest. This is the probability of choosing an alternative within the nest m multiplied by the probability of choosing m among all nests.

To add a representation learning term to the Nested Logit, one must only change the final function of L-MNL seen in Eq. (8) and adapt it to Eq. (19). This is done by removing the commonly used softmax layer seen in Fig. 2 for a custom-built network with added trainable weights for the nest factors. The new architecture is depicted in Fig. E.1 of Appendix E. Further changes to the network would allow us to easily model Cross-Nested Logit (Vovsha, 1997) and Generalized Extreme Value models (McFadden, 1978) with an added representation term.

5. Experiments

Our experiments demonstrate how our L-MNL and L-NL models increase the *predictability* of the MNL and NL models respectively while keeping their *interpretability*. The former, *predictability*, is directly quantified in terms of likelihood using two real-world datasets. The latter *interpretability*, however, is more challenging to assess. To do so, we define interpretability as the ability of the model to have both a quantifiable uncertainty of its parameters as well as an understandable or meaningful association with its variables. This latter point usually translates as the ability to mathematically derive and interpret the impact, or elasticity, of a variable. We will use synthetic and semi-synthetic data for which the true parameters are known to assess the quality of the estimates and the interpretability of every model.

5.1. Benchmarking models

We want to compare our models (L-MNL and L-NL) against key previous works. As our approach is both general and flexible, we retrieve previous works with appropriate modeling decisions on the \mathcal{X} , \mathcal{Q} input sets and \mathbf{f} , \mathbf{r} functions of Eq. (12).

Essentially, we compare the following models:

- **Logit(\mathcal{X})**: The standard MNL model with linear-in-parameter specification, and no learning component. The variables $x \in \mathcal{X}$ enter the linear utility specification and $\mathcal{Q} = \emptyset$. The model has $|\mathcal{X}|$ parameters.
- **DNN(n, \mathcal{Q})**: A Dense Neural Network for every alternative with softmax loss. This is the NN-MNL model proposed by Hruschka et al. (2004) and based on Bentz and Merunka (2000) with n neurons in the hidden layer ($H = n$) and a single layer deep. The variables $x \in \mathcal{Q}$ are the input features of the neural network. There is no variable entering the linear utility specification ($\mathcal{X} = \emptyset$). The model has around $n(|\mathcal{Q}| + |\mathcal{C}| + 1)$ weights.
- **DNN_L($n, \mathcal{X} = \mathcal{Q}$)**: A modified version of the NN-MNL model proposed by Hruschka et al. (2004). The variables $x \in \mathcal{X} (= \mathcal{Q})$ enter both the linear utility specification and the neural network. The latter can be seen as finding the residual from the linear input of \mathcal{X} and the overall architecture is close to the concept of the more well known residual building block from ResNet (He et al., 2016). The model has in the order of $(n + 1)(|\mathcal{X}| + |\mathcal{C}|)$ weights.
- **L-MNL($n, \mathcal{X}, \mathcal{Q}$)**: Our learning logit model with n neurons in the hidden layer ($H = n$). Part of the variables $x \in \mathcal{X}$ enter the linear utility specification and the remaining variables $q \in \mathcal{Q}$ are given as input features to the neural network. This separation satisfies the interpretability condition (Eq. (14)) for all x . The function $\mathbf{r}(\mathcal{Q})$ is defined by Eq. (17) where we limit the complexity to a single layer ($L = 1$). Therefore, the model has a total of $|\mathcal{X}|$ interpretable parameters and $n(|\mathcal{Q}| + |\mathcal{C}|)$ weights. We remind the reader that this architecture for the representation term is for the sake of example and more complex networks or methods depend on the datasets at hand and are open for future research.

Unless specified otherwise, all models have run for 200 epochs with an Adam optimizer (Kingma and Ba, 2014) running on default parameters from Keras python deep learning library (Chollet et al., 2015). Every model with a Neural Network has a 20% dropout regularizer following the DNN layer. The parameters in every model are initialized by the default random initialization of the used library, and no seed is set. Moreover, in our current framework, no notable difference in learning time between models are observed.

5.2. Synthetic data

We now use synthetic data to better analyze the performance of our L-MNL model in terms of both prediction performance and estimates accuracy. We start by describing how we generated synthetic data. Then, we perform Monte Carlo experiments to compare all benchmarking models on parameters estimation. Then, a scan on increasing number of neurons is introduced to analyze the impact of the NN architecture better. Finally, we investigate the case of strongly correlated variables between input sets \mathcal{X} and \mathcal{Q} . Additional experiments to study the particular effects of complexly correlated variables, missing variables, or sequential *versus* joint optimization strategies can be found in Appendix B. To study the limits of the hybrid model, a special case study of very noisy and highly non-linear utility specifications using semi-synthetic data can be found in Appendix C. For all experiments, all benchmarking models remain the same as in the previous Section (5.1).

5.2.1. Data generation

Synthetic data provides a controlled environment for analyzing our L-MNL model. To generate our synthetic data, we consider a simple binary choice model where an individual n has to choose between two options (e.g., choosing to use public transportation to go to work as opposed to taking an alternative mode). Following the generation process of Guevara (2015), we obtain the following utility specifications for $i = 1, 2$:

$$U_{in} = V_{in} + \varepsilon_{in}, \quad (20)$$

with

$$V_{in} = \underbrace{\beta_p \cdot p_{in} + \beta_a \cdot a_{in} + \beta_b \cdot b_{in}}_{\text{known relation}} + \underbrace{\beta_{qc} \cdot q_{in} c_{in}}_{\text{unknown interactions}} \quad (21)$$

$$p_{in} = 5 + z_{in} + 0.03wz_{in} + \varepsilon_{pin} \quad (22)$$

$$q_{in} = 2h_{in} + k_{in} + \varepsilon_{qin} \quad (23)$$

$$k_{in} = h_{in} + \varepsilon_{kin} \quad \text{for } i = 1, 2 \quad (24)$$

where β_x stand for parameters, and all variables a_{in} , b_{in} , c_{in} , z_{in} , wz_{in} , h_{in} and error terms ε_{pin} , ε_{qin} , ε_{kin} follow a uniform distribution. For the following experiments we consider the non-linear interactions among variables to represent unknown and undiscovered causalities during the modeling phase. Our data generation process slightly differs from Guevara (2015) on two points: first of all, p_{in} is not correlated with q_{in} as this is kept for an experiment by itself in Section 5.2.4 and Appendix B.1. Secondly, we have added an interaction term, c_{in} to allow a simulated situation where the modeler would misspecify the utility due to an undiscovered interaction. To avoid too large difference in magnitude order among our variables as well as the interaction term, we generated all variables as:

$$x_{in} \sim \mathcal{U}([-1, 1]) \quad (25)$$

Table 1

Monte Carlo average log-likelihood (\bar{LL}) and standard deviation ($s.d.(LL)$) for the different models. Based on the test set value, we conclude that our L-MNL learns the best general representation. An MNL model with true utility specification is given as a reference. Average of accuracies.

Model	Train set		Test set		Accuracies [%]		ρ_{test}^2
	\bar{LL}	$s.d.(LL)$	\bar{LL}	$s.d.(LL)$	\bar{Acc}_{train}	\bar{Acc}_{test}	
Logit(\mathcal{X}_{true})	-459	15	-94	8	78	77	0.32
Logit(\mathcal{X}_1)	-604	11	-123	6	67	66	0.11
DNN(25, \mathcal{Q})	-363	21	-112	13	84	74	0.19
DNN_L(25, $\mathcal{X} = \mathcal{Q}$)	-367	18	-108	12	83	75	0.22
L-MNL(25, \mathcal{X}, \mathcal{Q})	-429	16	-97	8	80	76	0.30

Under the classical logit assumption that $\varepsilon_{in} \sim \text{i.i.d.} EV(0, 1)$ and given fixed values of parameter estimates β in the utility specification, the choice probabilities $P_{in}(\mathbf{x}_n)$ are known, and the synthetic choice y_{in} can then be seen as a Bernoulli random variable that takes the value 1 with probability $P_{in}(\mathbf{x}_n)$ and 0 with probability $1 - P_{in}(\mathbf{x}_n)$. Formally, we assume that

$$y_{1n} \sim \text{Bern}(P_{1n}(\mathbf{x}_n)), \quad (26)$$

$$y_{2n} = 1 - y_{1n} \quad \forall n = 1, \dots, N. \quad (27)$$

Using synthetic data allows us to compare our L-MNL model with standard choice models, not only in terms of prediction performance, but also in accuracy in parameter estimation since we know the true value of parameter estimates β , i.e., the true linear and non-linear dependencies between the variables.

5.2.2. Monte carlo experiment

We rely on Monte Carlo simulations to investigate how well each model performs in both prediction and parameter estimation of Eq. (21). To do so, we select a true utility specification with $\beta_p = -1$, $\beta_a = 0.5$, $\beta_b = 0.5$, $\beta_{qc} = 1$. For each experiment, we generate 1,000 synthetic individual observations for the training set, and 200 more for the testing set. The following results are obtained by performing 100 of such experiments.

Our new choice model is defined as

- L-MNL(25, \mathcal{X}, \mathcal{Q}), with $\mathcal{X} = \{\mathbf{p}, \mathbf{a}, \mathbf{b}\}$, and $\mathcal{Q} = \{\mathbf{q}, \mathbf{c}\}$,

and is compared with the following three benchmarking models and one reference model⁵:

- Logit(\mathcal{X}_1), with $\mathcal{X}_1 = \{\mathbf{p}, \mathbf{a}, \mathbf{b}, \mathbf{q}, \mathbf{c}\}$,
- DNN(25, \mathcal{Q}), with $\mathcal{Q} = \{\mathbf{p}, \mathbf{a}, \mathbf{b}, \mathbf{q}, \mathbf{c}\}$,
- DNN_L(25, $\mathcal{X} = \mathcal{Q}$), with $\mathcal{X} = \mathcal{Q} = \{\mathbf{p}, \mathbf{a}, \mathbf{b}, \mathbf{q}, \mathbf{c}\}$,
- *Logit(\mathcal{X}_{true}), with $\mathcal{X}_{true} = \{\mathbf{p}, \mathbf{a}, \mathbf{b}, \mathbf{qc}\}$.

The log-likelihood results from Monte Carlo experiments can be seen in Table 1. As expected, the best models in terms of predictive performance are the neural network-based choice models. Among them, despite overfitting on the train set, our L-MNL gives the best general representation of the data, achieving the best predictive performance in the test set (with a LL of -97 compared to -123 for the standard MNL model).

In order to evaluate the models in terms of accuracy in interpretable parameter estimation, we define the relative errors e_β and e_{β_i/β_j} as

$$e_\beta = \left| \frac{\beta - \hat{\beta}}{\beta} \right|, \quad (28)$$

$$e_{\beta_i/\beta_j} = \left| \frac{e_{\beta_i} - e_{\beta_j}}{1 - e_{\beta_j}} \right|. \quad (29)$$

The relative error results⁶ from Monte Carlo experiments can be seen in Table 2. As the beta parameters of the model have different values, we normalize the performance by reporting the average relative error of the estimates for every model, as well as the standard deviation of all relative errors. We see that our L-MNL greatly outperforms every model in the ability to recover the true parameter values with a relative error smaller than 1% away from the true model. It is worth noting that for the ratios of parameters, the MNL models perform second best, although they have high relative errors at parameters level. This phenomenon has been investigated in a few papers (e.g., Lee, 1982; Cramer, 2007) where it has been shown that even

⁵ The reference model is considered differently than the benchmarks as it breaks the assumption that the modeler does not know the ground truth.

⁶ Note that the DNN(25, \mathcal{Q}) benchmarking model does not appear in Table 2 since no variable enters the linear utility specification.

Table 2

Monte Carlo relative errors for the different models in [%] with \bar{e} the average relative error, s.d. its standard deviation and β_p, β_a are from Eq. (21).

Model	\bar{e}_{β_p}	s.d. (e_{β_p})	\bar{e}_{β_a}	s.d. (e_{β_a})	$\bar{e}_{\beta_p/\beta_a}$	s.d. (e_{β_p/β_a})
Logit(\mathcal{X}_{true})	6.4	± 4.9	14.4	± 10.7	10.8	± 9.7
Logit(\mathcal{X}_1)	26.7	± 6.2	26.7	± 14.7	15.5	± 12.4
DNN_L(25, $\mathcal{X} = \mathcal{Q}$)	60	± 32.4	74	± 54	460	± 166
L-MNL(25, \mathcal{X}, \mathcal{Q})	7.1	± 5.1	15.2	± 11.7	11.3	± 10.3

Table 3

Monte Carlo hypothesis testing for the different models, for β_p and β_a taken separately and for their ratio. Parameters β_p, β_a are from Eq. (21).

Model	% of experiments not rejecting H_0	
	β_p and β_a	β_p/β_a
Logit(\mathcal{X}_{true})	97.5	94
Logit(\mathcal{X}_1)	34	97
DNN_L(25, $\mathcal{X} = \mathcal{Q}$)	25.5	37
L-MNL(25, \mathcal{X}, \mathcal{Q})	95	94

when utility is misspecified, the MNL model is still able to retrieve good ratios between estimates. The hybrid models with $\mathcal{X} = \mathcal{Q}$, on the other hand, generate high errors in parameter estimates. It indicates that the NN component is also partially learning linear dependencies of \mathbf{p} or \mathbf{a} , and prevents the linear function f , or the β parameters, to reach a minimum as good as our L-MNL. This is a direct consequence from Eq. (13) as the impact of the variable is shared between both data-driven and knowledge-driven terms.

Finally, we conduct hypothesis testing to determine whether the estimated parameters are statistically different from the true ones. Formally, we consider the following null and alternative hypotheses: $H_0 : \hat{\beta} = \beta$, and $H_1 : \hat{\beta} \neq \beta$, where the null hypothesis is rejected for a t -test above 1.96, which corresponds to a 5% confidence interval. The standard deviation of the estimates in \mathcal{X} for the t -tests are obtained via Hessian approximation⁷.

The results are shown in Table 3. We see that the coefficients alone are almost always statistically different from the true ones for every model except L-MNL. The NN component of our model fixes the estimators by learning only on the data that are included in the linear component. The ratio of parameters is well retrieved with both MNL and L-MNL models. Concerning the residual block model, we see that its NN component compromises the ratios between parameters. Indeed, as seen in Eq. (14), part of the Neural Network has overlapped over the estimates of β_p and β_a , thus breaking both the estimates and their ratio.

To better illustrate the origin of ratio discrepancy of variables $x_{in} \in \mathcal{X}$ when $\frac{\partial r_{in}}{\partial x_{in}} \neq 0$ in hybrid models, we have shown in Section 5.2.4 that we retrieve the same biasing effect on our L-MNL when we make use of highly correlated data between both sets.

5.2.3. Choice of neural network architecture

Choosing the structure of an NN requires one to have enough capacity to capture the underlying pattern of the data without overfitting. In the following, we gradually increase the size of the NN component of our L-MNL, spanning from an underfit of the non-linear *truth* of Eq. (21) to an overfit of the generated data. To illustrate these effects, we study the values of β_p and β_a , as well as likelihoods of both train and test sets.

The results are depicted in Fig. 3, where we take L-MNL($n, \mathcal{X}, \mathcal{Q}$) with a single layer $L=1$, and scan from $n=0$ neurons in the hidden layer to $n=5000$. We see that from $n=0$ neurons (\equiv MNL) to about $n \sim 10$, the NN has not yet captured all the non-linearities of the original utility specification. This can be seen by the higher values in likelihood and how the parameter estimates have not yet reached their true minimum. From $n=10$ to $n=100$, the model reaches stable values where the linear terms are equivalent to their ground truth. The model performs almost as well as the true model, depicted by the boundary lines, leading us to believe that the NN component has successfully learned the non-linearities of the data. When we continue to increase the number of neurons, we see the effects of overfitting, namely a drop in train likelihood and an increase in test likelihood. We finally observe that the values of β_p and β_a also start to suffer from the overfitting, as the L-MNL is no longer a general representation of the data, but has become specific to the training set. This experiment also shows that for equal optimal likelihood values, the change in architecture and non-convex property of the neural network does not significantly affect the β estimates as it converges to the ground truth. We can see that the standard deviation spread of the estimates over 100 experiments match with those of the true model also reported on the figure. Although, is

⁷ Note that profile likelihood approaches such as Hessian approximation create confidence intervals that fail to have the desired frequentist coverage properties. See Berger et al. (1999) for further discussion on that topic.

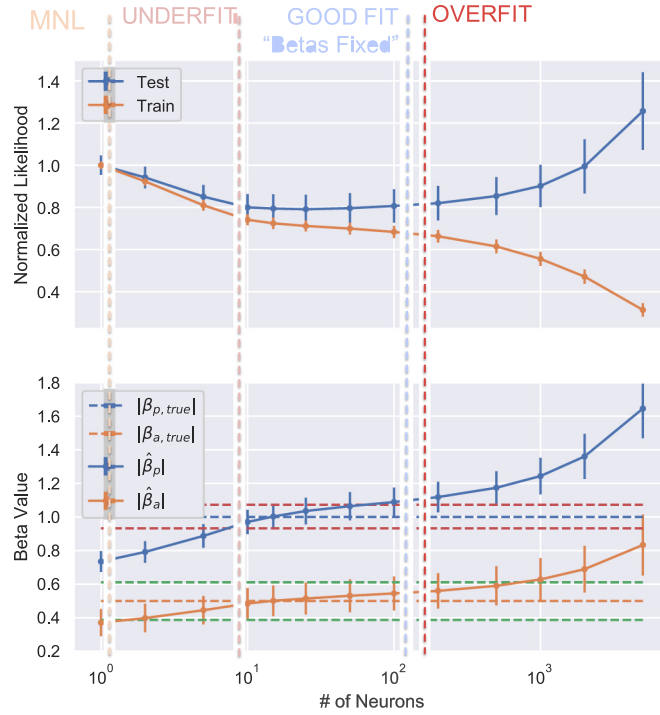


Fig. 3. Likelihood and values of parameter estimates for an increasing number of neurons in the hidden layer. Error bars show the standard deviation for 100 experiments. Red and Green lines show the standard deviation spread of the true model's parameter estimation. Best results are obtained with $n \in [10, 100]$.

has been shown in Draxler et al. (2018) and Garipov et al. (2018) that a neural network may have many different parameter estimates for similar valued optimum, it does not seem to be the case for our convex part of the model.

5.2.4. Impact of strongly correlated variables

Experimental results from Section 5.2.2 suggest that having identical variables (or features) in both sets lead to poor parameter estimates because of the neural network's better ability to discover correlation with the observations for all its variables. In this sense, Eq. (13) shows how a variable's impact will be shared between both knowledge-driven and data-driven terms if it belongs to both input sets.

Although our L-MNL requests the variables in the two sets to be different, strong correlation between the variables can remain an issue. To investigate the impact of correlated variables, we replace the original explanatory variable q_{in} with a new explanatory variable q'_{in} that is defined to be correlated to p_{in} . Formally, we define q'_{in} as

$$q'_{in} = s \cdot p_{in} + \sqrt{1 - s^2} \cdot q_{in}, \quad (30)$$

where $s \in [0, 1]$ is the correlation coefficient.

Our new choice model is defined as

- L-MNL(\mathcal{X}, \mathcal{Q}), with $\mathcal{X} = \{\mathbf{p}, \mathbf{a}, \mathbf{b}\}$, and $\mathcal{Q} = \{\mathbf{q}', \mathbf{c}\}$,

and is compared with the following two benchmarking models:

- Logit(\mathcal{X}_1), with $\mathcal{X}_1 = \{\mathbf{p}, \mathbf{a}, \mathbf{b}, \mathbf{q}', \mathbf{c}\}$,
- Logit(\mathcal{X}_2), with $\mathcal{X}_2 = \{\mathbf{p}, \mathbf{a}, \mathbf{b}\}$.

The Monte Carlo mean relative errors for different levels of correlation are shown in Fig. 4. Overall we can see two effects: bias with high variance due to the correlated variables (Logit(\mathcal{X}_2), L-MNL), and bias due to misspecification (Logit(\mathcal{X}_1), Logit(\mathcal{X}_2)). For a correlation coefficient below $s \leq 0.8$ we can see the L-MNL has much better estimates with respect to the true model than the Logit models. For $s \leq 0.4$ the L-MNL is practically as efficient as the true model. This is important as it suggests that in order to get insightful results, the modeler has to carefully check that the variables that enter as input features in the neural network are not the same or not too strongly correlated with the variables in the linear utility specification.

Finally, we also show in Appendix B.1, that our new choice model does not suffer from complexly correlated variables between both sets \mathcal{X} and \mathcal{Q} .

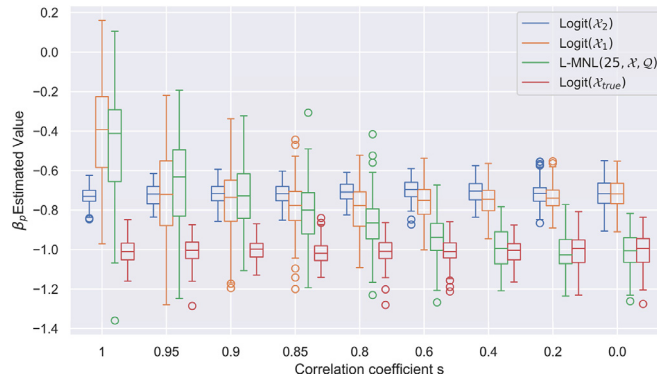


Fig. 4. Impact of correlated variables on parameter estimates, where $\mathbf{p} \in \mathcal{X}$ is correlated to $\mathbf{q}' \in \mathcal{Q}$ (see Eq. (30)). We see in this case that the L-MNL correlation bias is smaller than the bias due to underfit for all $s \leq 0.8$. For each coefficient, we performed 100 experiments.

Table 4

Swissmetro benchmark utility function, from Bierlaire et al. (2001) .

Variables		Alternative		
		Car	Train	Swissmetro
ASC	Constant	Car-Const		SM-Const
TT	Travel Time	B-Time	B-Time	B-Time
Cost	Travel Cost	B-Cost	B-Cost	B-Cost
Freq	Frequency		B-Freq	B-Freq
GA	Annual Pass		B-GA	B-GA
Age	Age in classes		B-Age	
Luggage	Pieces of luggage	B-Luggage		
Seats	Airline seating			B-Seats

Table 5

Variables in the Swissmetro dataset used for modeled component of the utility specification.

Variable	Description
TT	Door-to-door travel time in [minutes], scaled by 1/100.
Cost	Travel cost in [CHF], scaled by 1/100.
Freq	Transportation headway in [minutes]
GA	Binary variable indicating annual pass holders (=1).
Age	Integer variable scaled with the traveler's age.
Luggage	Integer variable scaled with amount of luggage during travel.
Seats	Binary variable for special seats configuration in Swissmetro (=1).

5.3. A real case study: The swissmetro dataset

For the following experiments, we use the openly available dataset Swissmetro (Bierlaire et al., 2001). The Swissmetro dataset consists of survey data collected in Switzerland during March 1998. The respondents provided information to analyze the impact of a new innovative transportation mode, represented by the Swissmetro, a revolutionary mag-lev underground system. Each individual reported the chosen transportation mode choice for various trips, including the car, the train, or the Swissmetro. The original dataset contains 10,728 observations. We removed observations for which the information regarding the chosen alternative was missing (9 observations). For convenience, we also decided to discard the observations for which the three alternatives car, train, and Swissmetro were not all available (1,683 observations). Our dataset contains 9036 observations. We split this initial dataset into a training set of 7234 observations and a test set of 1802 observations.

5.3.1. Models comparison

In Table 7, we present a detailed comparison of all the benchmarking models with respect to log-likelihood in the training and testing sets as well as the parameters estimates. Since the true values of parameter estimates are unknown, we compare the values obtained with our L-MNL to the ones obtained using the MNL model described in Bierlaire et al. (2001),

Table 6
Unused variables in the Swissmetro dataset.

Variable	Description
Purpose:	Integer variable indicating the trip purpose (business, leisure, etc.)
First :	Binary variable indicating if first class (=1) or not (=0)
Ticket:	Integer variable indicating the ticket type (one-way, half-day, etc.)
Who:	Integer variable indicating who is paying the ticket (self, employer, etc.)
Male:	Binary variable indicating the traveler's gender (0 = female, 1 = male)
Income:	Integer variable indicating the traveler's income per year.
Origin:	Integer variable indicating the canton in which the travel begins.
Dest:	Integer variable indicating the canton in which the travel ends.

Table 7

Comparison of log-likelihood and parameters estimates for different models with utility specification of [Bierlaire et al. \(2001\)](#). Number of observations = 7234.

Model	Parameters	Estimates	Std errors	t-stat	p-value
MNL $\rho_{test}^2 = 0.28$ $\mathcal{L}(\hat{\beta}) = -5764$ $\mathcal{L}_{test}(\hat{\beta}) = -1433$	ASC_{Car}	1.08	0.162	6.67	0.00
	ASC_{SM}	1.05	0.153	6.84	0.00
	β_{age}	0.146	0.436	3.35	0.00
	β_{cost}	-0.695	0.0423	-16.42	0.00
	β_{freq}	-0.733	0.1132	-6.47	0.00
	β_{GA}	1.54	0.167	9.24	0.00
	$\beta_{luggage}$	-0.114	0.0488	-2.338	0.02
	β_{seats}	0.432	0.115	3.76	0.00
	β_{time}	-1.34	0.051	-26.18	0.00
L-MNL(100, \mathcal{X}_1 , \mathcal{Q}_1) $\rho_{test}^2 = 0.41$ $\mathcal{L}(\hat{\beta}) = -4511$ $\mathcal{L}_{test}(\hat{\beta}) = -1181$	ASC_{Car}	0.106	0.174	0.61	0.54
	ASC_{SM}	0.454	0.163	2.80	0.01
	β_{age}	0.390	0.045	8.63	0.00
	β_{cost}	-1.378	0.048	-28.45	0.00
	β_{freq}	-0.860	0.127	-6.77	0.00
	β_{GA}	0.214	0.194	1.10	0.27
	$\beta_{luggage}$	0.116	0.0529	2.19	0.03
	β_{seats}	0.104	0.109	0.95	0.34
	β_{time}	-1.563	0.056	-27.97	0.00
DNN_L(100, $\mathcal{X}_1 = \mathcal{Q}$) $\rho_{test}^2 = 0.37$ $\mathcal{L}(\hat{\beta}) = -4964$ $\mathcal{L}_{test}(\hat{\beta}) = -1257$	ASC_{Car}	0.365	0.165	3.61	0.00
	ASC_{SM}	0.549	0.162	2.22	0.03
	β_{age}	0.087	0.0423	2.07	0.04
	β_{cost}	-0.897	0.046	-19.46	0.00
	β_{freq}	-0.639	0.123	-5.20	0.00
	β_{GA}	1.40	0.172	8.15	0.10
	$\beta_{luggage}$	0.186	0.0523	3.52	0.00
	β_{seats}	0.233	0.102	2.29	0.02
	β_{time}	-1.146	0.049	-23.32	0.00
Logit(\mathcal{X}_{dum}) (all 41 inputs) $\rho_{test}^2 = 0.33$ $\mathcal{L}(\hat{\beta}) = -5451$ $\mathcal{L}_{test}(\hat{\beta}) = -1322$	β_{cost}	-1.062	0.059	18	0.00
	β_{freq}	-0.79	0.118	6.69	0.00
	β_{time}	-1.326	0.053	25.02	0.00

L-MNL(100, \mathcal{X}_2 , \mathcal{Q}_2) $\rho_{test}^2 = 0.44$ $\mathcal{L}(\hat{\beta}) = -3895$ $\mathcal{L}_{test}(\hat{\beta}) = -1108$	β_{cost}	-1.671	0.0523	-31.94	0.00
	β_{freq}	-0.865	0.0765	-11.30	0.00
	β_{time}	-1.769	0.0389	-45.4	0.00

whose linear-in-parameter utility specification is described in [Table 4](#) and variables described in [Table 5](#). The results of parameter optimization for this MNL model are shown in [Table 7](#)⁸

We next estimate our new choice model L-MNL(100, \mathcal{X}_1 , \mathcal{Q}_1), where \mathcal{X}_1 contains the same variables as the ones included in the utility specification of [Bierlaire et al. \(2001\)](#), and \mathcal{Q}_1 contains all the unused variables in the dataset. These variables are described in [Table 6](#). The results of this model are depicted in [Table 7](#). We see that adding the representation learning component in the utility specification significantly increases the log-likelihood, suggesting that these variables contain information that helps to explain travelers' choice. However, we also observe that several estimates are not statistically different from zero (p-value > 0.05). We have concluded in [Section 5.2.2](#) and further developed in [Section 5.2.4](#), that only strong correlation between variables can lead to biased parameter estimates. However, the correlation among the variables has

⁸ It is worth noting that the values of estimates that we obtain slightly differ from the ones depicted in the original paper. This is normal since the original Swissmetro dataset is different from the training dataset that we used.

Table 8
Parameter ratio comparison.

Model	Value of Time	Value of Frequency	Train Log-Likelihood	Test Log-Likelihood
Logit(\mathcal{X}_1)	0.52	0.95	-5764	-1433
Logit(\mathcal{X}_{dum})	0.80	1.34	-5451	-1322
DNN_L(\mathcal{X}_1)	0.78	1.40	-4964	-1257
L-MNL(\mathcal{X}_1)	0.88	1.60	-4511	-1181
L-MNL(\mathcal{X}_2)	0.94	1.93	-3895	-1108
CNL(\mathcal{X}_1)	0.59	1.52	-5711	-1415
TPM(\mathcal{X}_1)	0.72	1.59	-4752	-1350

been investigated and does not seem to be the issue here (see Table G.1 in Appendix for the correlation among variables). Moreover, the DNN_L(100, $\mathcal{X}_1 = \mathcal{Q}$) model (see Table 7) has one-to-one correlation among variables in both sets and does not lose significance in its parameters. One explanation could be that the coefficients in L-MNL have lost their significance due to the neural network's ability to learn better which variables and interactions are most correlated to the data. Therefore, the significance of the same parameters in the initial MNL model can originate from a bias due to the model's underfit. Also, with many explanatory variables being omitted in the initial MNL model, it is worth noting that the model is more likely to be subject to endogeneity, *i.e.*, correlation among the dependent variables and the error term. Endogeneity is a well-known cause of bias in parameter estimates and can cause bias in the parameter estimates of the initial MNL model.

Finally, in Table 7, we estimate the new choice model L-MNL(100, $\mathcal{X}_2, \mathcal{Q}_2$), where \mathcal{X}_2 contains only the variables needed to compute the Value of Time (VOT) and Value of Frequency (VOF), *i.e.*, the variables *time*, *cost*, and *frequency*. All other variables constitute the set \mathcal{Q}_2 and are therefore given to the neural network. We observe that all parameter estimates are significant, while the log-likelihood has greatly increased compared to the standard MNL model. In other words, the representation term was able to get information from the previously non-significant variables in the linear specification.

Furthermore, as L-MNL has a bigger feature space than the aforementioned models, we have designed a dummy coded Multinomial Logit as another benchmark. The parameter space contains homogeneous β_x for the variables in \mathcal{X}_2 and alternative specific β_{iq} for all variables contained in \mathcal{Q} . The full feature space is written as $\mathcal{X}_{dum} = \mathcal{X}_2 \cup \mathcal{Q}_2$ and the results can be seen in Table 7. The big difference in final likelihoods with our L-MNL models further demonstrate that this dataset indeed contains complex functions and interactions among its variables which may be difficult to capture by the modeler.

We show a comparison of VOT and VOF for the different models in Table 8. We have also added a Cross-Nested Logit (CNL), and a Triangular Panel Mixture (TPM) model to the results for comparison purposes, whose detailed parameter values can be found in Appendix F and for which the implementation was taken from Biogeme examples⁹. The difference in log-likelihood values between the L-MNL and the Logit(\mathcal{X}_1) suggests that the MNL model suffers from underfitting that leads to significant ratio discrepancy. A similar conclusion was derived only under strong non-linearities in the dataset, suggested in Appendix C. The linear specification of the MNL model does not allow to capture the Swissmetro's non-linear dependencies among variables. Moreover, we may observe for both added DCM methods, that their ratio values move towards those of L-MNL as their likelihood increases. This trend implies a better specification in their utility would allow them to reach even more similar values in ratios and likelihood with our new choice model.

In support of these conclusions, we incrementally increase the size of the NN component. We observe in Fig. 5 the parameter values with respect to the number of neurons in our neural network. As for the synthetic data experiments in Section 5.2.3, we see that stable ratios for VOT and VOF are obtained for a neural network having from 10 to 200 neurons. We see the MNL ratios for $n=0$ neuron highlights the underfit of the MNL model. An overfit effect is observed starting from 500 neurons, where the test likelihood no longer improves.

5.3.2. Fixing artifact correlation between alternatives

In this experiment, we show results of a nested logit based on the baseline specification Bierlaire et al. (2011) as well as the nested generalization of the L-MNL models seen in the previous experiment. Moreover, we have added a model containing \mathcal{X}_1 and a smaller set $\mathcal{Q}_0 = \{\text{Purpose, First, Male, Income}\}$ for illustrative purposes. We may observe in Table 9 the Log-Likelihoods for the standard models and their nested variants with the following nests: (1) *known*: {Car, Train}, and (2) *new*: {Swissmetro}.

The experiment shows that as we allow the representation term to find a data-driven result over an increasing amount of input, *i.e.*, from none to \mathcal{Q}_2 , the nest value reduces itself to 1. If we relate this to the nested theory reminded in Section 4.4, this means that the correlation between the choices in a nest –car together with train– disappears as we better fit with a data-driven term. This suggests that the random terms were initially correlated between alternatives due to misspecification, but have become independent with the added representation term. This interpretation is in line with the mathematical implications of Eq. (15).

In further experiments, it would be possible to identify which variables are most active in the correlation between alternatives, thanks to ablation and elasticity studies. An example of such experiments can be seen in Appendix D.1.

⁹ <https://biogeme.epfl.ch/examples.html>

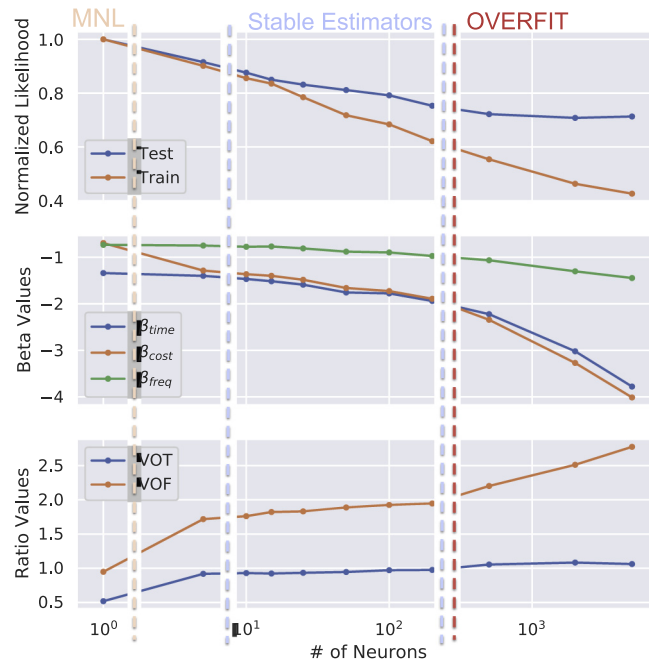


Fig. 5. Scan of Likelihood and Beta values over number of neurons n in the densely connected layer for L-MNL($100, \mathcal{X}_2, Q_2$). For $n \in [10, 200]$ we have $VOT \approx 1$ and $VOF \approx 1.9$. Over $n = 200$, we have signs of overfit.

Table 9

Performance of Swissmetro models with added nest structure. Increase in fit lowers the need of nests.

	Standard Models		Nest Factor	Nested Models	
	Train LL	Test LL		Train LL	Test LL
Logit(\mathcal{X}_1)	-5764	-1433	1.43	-5734	-1413
L-MNL(\mathcal{X}_1, Q_0)	-5324	-1348	1.37	-5305	-1340
L-MNL(\mathcal{X}_1, Q_1)	-4268	-1173	1.01	-4260	-1173
L-MNL(\mathcal{X}_2, Q_2)	-3895	-1108	1	-3894	-1108

5.4. Revealed preference study: Optima dataset

In contrast with the previous Swissmetro study, we investigate in this section a case much more difficult for representation learning due to the size limitation of gathered data. The project named Optima is a small survey collected in Switzerland between 2009 and 2010 where respondents filled up extensive information in the topic of mode choice, including time and cost of performed trips, socio-economic characteristics as well as opinions on statements and answers to "semi-open" questions as defined in Glerum et al. (2014). More details on this particular dataset can be read in Bierlaire et al. (2011) and Glerum et al. (2014). To benchmark our results, we take the latest results discovered on this dataset, notably a well-specified base multinomial logit and one of its suggested extension, the Integrated Choice and Latent Variable (ICLV) model, from Fernández-Antolín et al. (2016).

5.4.1. Models description

The multinomial logit specification from Fernández-Antolín et al. (2016) can be seen in Table 10. We then compare two L-MNL models and two pure Neural Networks. The input sets for our first model is defined such that \mathcal{X}_1 contains the variables from the baseline specification, while Q_1 are those shown in Table 11. In this case it is interesting to note, that the straightforward interpretability condition for *cost* in Section 4.2 is still satisfied, even though it interacts with *income* $\in Q$. Correlation between inputs of both sets can be found in Appendix H. Based on experiments seen in Section 5.2.4 and Appendix B.1, this specification will not suffer bias from correlated variables. The second L-MNL model has an input set \mathcal{X}_2 , which contains only time, cost, and distance attributes, while the removed features were added into the second set to create Q_2 . The Neural Networks have all features from both tables as inputs, but differ in size, the first having a single layer of 100 neurons, just like the L-MNL models, and the second has a single layer of 30 neurons.

We preprocess the data by removing any entry which does not have all choices available or unanswered variables we make use of. This gives us a total of 1376 answers, where 1089 are kept for the training set and 287 for the testing set.

Table 10
Base model specification from Fernández-Antolín et al. (2016).

Variables		Alternative		
		Public Transportation	Car	Slow modes
ASC	Constant	PT-Const	CAR-Const	
TT	Travel Time [min]	B-Time-PT	B-Time-CAR	
MCost	Marginal Cost Income	B-MCost-PT	B-MCost-CAR	
Distance	Trip distance [km]			B-Dist
Work	Work related Trip		B-Work	
French	French Speaking area		B-French	
Student	Occupation is student	B-Student		
Urban	Urban area	B-Urban		
NbChild	Number of Children		B-NbChild	
NbCar	Number of Cars		B-NbCar	
NbBicy	Number of Bicycles			B-NbBicy

Table 11
Added variables for representation learning term in Optima dataset. All variables gave -1 for missing values.

Variable	Description
Age:	Age of the respondent (in years)
HouseType :	1 is individual house (or terraced house), 2 is apartment, 3 is independent room
Gender:	1 is man, 2 is woman. -1 for missing value.
Education:	Highest education achieved ^a . Categories from 1 to 8.
FamilSitu:	Family situation ^a . Categories from 1 to 7.
ScaledIncome:	Integer variable indicating the traveler's income per year.
OwnHouse:	Do you own the place where you are living? 1 is yes, 2 is no
MotherTongue:	1 for german or swiss german, 2 for french, 3 for other,
SocioProfCat:	Socio-professional ^a categories from 1 to 8.

^a More details at <https://biogeme.epfl.ch/data.html>.

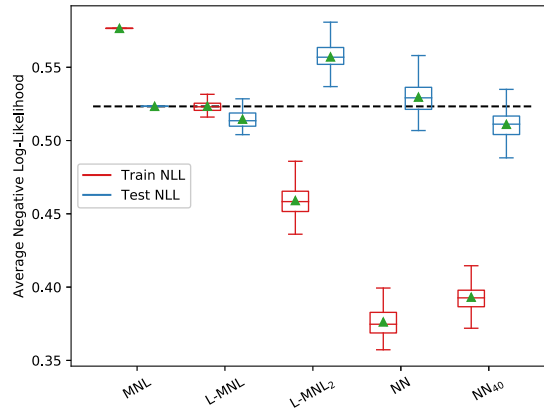


Fig. 6. Performance of multiple models on a small revealed preference dataset (Optima) for 100 minimization iterations. The first two models have expert modeling of the utility specification and generalize well. The last three have small or no modeling of the utility and show clear signs of overfitting as well as high variance in performance results.

This is considered a tiny dataset for neural networks, which makes it a difficult task to avoid overfitting. Therefore, we have added a 30% dropout layer and an l_2 regularizer of weight $\lambda = 0.5$ (Krogh and Hertz, 1992).

5.4.2. Expert modeling as a regularizer

The performances of the five models described above are shown in Fig. 6, where the same minimization code was run 100 times for every model. The observed variation comes from the change in starting values, which ultimately brings to a slightly different optimum given the strong regularizers and fixed number of training cycles, i.e., 80 epochs.

We may observe that both $\text{MNL}(\mathcal{X}_1)$ and $\text{L-MNL}(\mathcal{X}_1, \mathcal{Q}_1)$ generalize well with an increase in performance for the second model. As opposed to the case seen in Swissmetro Section 5.3.1, simplifying the \mathcal{X}_1 to the minimum of interest \mathcal{X}_2 allows the model to overfit and perform poorly. The same can be seen for a full neural network of 100 neurons. If we try to reach better performances with a smaller neural network, including the high regularizers and low training cycles, we may observe the result of the last model, containing 40 neurons. Although we still overfit on the training set, we have better

Table 12
Models accuracy on training and testing sets of Optima.

Model	Logit(\mathcal{X}_1)	L-MNL($\mathcal{X}_1, \mathcal{Q}_1$)	NN ₄₀ ($\mathcal{X} \cup \mathcal{Q}$)	ICLV(\mathcal{X}_1)
Accuracy Train [%]	76.8	80.4	86.1	80.0
Accuracy Test [%]	76.7	79.2	81.3	77.7

generalization to the test set. However, the optimum has high variance, and parameter interpretation may only be recovered through elasticity plot studies, missing the straightforward interpretation of discrete choice model parameters.

5.4.3. Benchmarking with ICLV

In Fernández-Antolín et al. (2016), the base logit is also compared to an ICLV model, where the latent variable CarLoving is added to the model. This latent variable is estimated with two Likert indicators (Likert, 1932):

- I_1 : It is difficult to take the public transport when I travel with my children.
- I_2 : With my car I can go wherever and whenever.

and the following specifications:

$$t_{car} \cdot I_i = \alpha_i + \lambda_i \cdot \text{CarLoving} \cdot t_{car} + \omega_i \quad (31)$$

where $\omega_i \sim \mathcal{N}(0, \sigma_i)$ is the random term, α_i , λ_i , σ_i are parameters to be estimated and:

$$\text{CarLoving} = \eta_{car} + \omega \quad (32)$$

with η_{car} and σ being the estimated parameters and $\omega \sim \mathcal{N}(0, \sigma)$ the random term.

Further details are described in Fernández-Antolín et al. (2016). Given that the full loss function from an ICLV model differs from that of the base logit or MNL, we may only compare models by observing their accuracy on a test set. The implementation of the new model was done in biogeme (Bierlaire, 2003) and the results can be seen in Table 12. We may observe that the ICLV model performs better than MNL while having the same initial feature space. A stronger increase in accuracy could be expected with added latent variables or more complex structural equations. For the models with a neural network, we have reported the average accuracy of all iterations. As seen from the loss in Fig. 6, the Neural Network ($n = 40$) performs better than the L-MNL on average while slightly overfitting the training set. However, the obtained model does not contain straightforward interpretable parameters, has higher variance in performance, and required more efforts in fine-tuning for optimal performance. In the DCM sense, the most useful models would be L-MNL with the full expert specification and the ICLV model. Adding a representation term does not limit itself to MNL or its nested generalizations. We believe ICLV architectures may also benefit from a data-driven counterpart and we consider this for future research.

In conclusion, when working with small datasets, Discrete Choice Modeling may effectively generalize very well when a good specification is found. The effect of the learning term may show how well our model is specified, i.e., a low increase in performance would signify a well-specified utility. Lastly, for the machine learning community, this may also encourage to use expert modeling of a utility specification as an efficient regularizer for neural networks when working with small datasets.

6. Future directions

Our proposed framework paves the way to several avenues of research. One main direction is the new possibility to investigate many more types of datasets for discrete choice modeling. Indeed, representation learning methods exist for all types of inputs, including continuous signals, images, time series, and more. Consequently, we can explore new neural network architectures (e.g., other variants of the residual blocks). Although research has already been done in this direction, e.g., Otsuka and Osogami (2016) who used images for choice classification, our model is the first to propose the coexistence of these new inputs with standard discrete choice modeling variables. Furthermore, we suggest that the representation learning architecture, together with the modeled specification architecture, complement each other to reach both a better optimum when they are jointly optimized.

Secondly, we believe that other discrete choice models such as more advanced GEV models, Mixed Logit or Latent Class models can also benefit from an added data-driven term while keeping high degree of interpretability. Integrating them in a unified framework/library is a big direction for future research. Indeed, as of now, by making use of the NN structure we have been able to implement a new learnable term to the Multinomial logit, and thanks to the implementation of multiple custom loss layers in a deep learning library, we have implemented its first nested generalization, the Nested logit. Both models have benefited from a data-driven term, and we hope to bring them to more advanced models in the near future.

Moreover, the proposed architecture of our model may also help in the task of modeling the knowledge-driven term of the utility specification via feature selection. Indeed, understanding what a data-driven method has learned is an active field of research and may overall greatly benefit the field of discrete choice modeling. In our specific case, by capturing insights about the representation term, we would be able to reduce the number of parameters in \mathcal{Q} incrementally, and add their interactions or non-linear functions in \mathcal{X} when discovered, taking advantage of the joint optimization benefits of our

model. To this end, we may also interest ourselves to recent methods which move towards explainable A.I., notably those on network visualization and interpretation (Ribeiro et al., 2016; Samek et al., 2016; Shrikumar et al., 2017; Murdoch et al., 2019) and compare with recent works on interpretable or knowledge-injected machine learning (Borghesi et al., 2020; von Rueden et al., 2020; Murdoch et al., 2019).

Yet another avenue of research concerns the structure and role of the representation term in the utility function. Indeed, for example, one could have multiple terms per utility, such that each chosen input set Q_i , belonging to their own respective network, would create a meaningful embedding in the utility. In the same spirit, one could inspire ourselves of Wang et al. (2020) to have an added utility specific network per alternative. On another hand, the data-driven term may simply be used to discover usually complex specifications in the utility. This has been done in Han et al. (2020), a direct extension of our work, which make use of neural networks to automatically discover the true taste heterogeneity function of a parameter.

Finally, while we have shown the benefits of our data-driven term in the DCM field, the machine learning community can also benefit from our formulation to tackle small datasets. We have shown that our architecture may perform as a regularization tool for common deep learning methods when applied to small datasets. This may encourage the machine learning community to reach out for DCM practices and make use of a priori specification to help the performance of their models.

7. Conclusions

In this paper, we introduced a novel general and flexible theoretical framework that integrates a representation learning technique into the utility specification of a discrete choice model to automatically discover good utility specification from available data. This data-driven term may account for many forms of misspecifications and greatly improves the overall predictability of the model. Also, unlike the existing hybrid models in the literature, our framework is carefully designed to keep the interpretability of key parameters, which is critical to allow researchers and practitioners to get insights into the complex human decision-making process.

Using synthetic and real world data, we demonstrated the effectiveness of our framework by augmenting the utility specification of the Multinomial Logit, as well as the Nested Logit, with a new non-linear representation arising from a neural network, leading to new choice models referred to as the *Learning Multinomial Logit* (L-MNL) and *Learning Multinomial Nested Logit* (L-NL) models. Our experiments showed that our models outperformed the traditional choice models and existing hybrid models, both in terms of predictive performance and accuracy in parameter estimation.

There is a growing interest within the transportation community to exploit multidisciplinary methods to solve the ever more challenging problems its members face. By making our code openly available, we hope that we successfully contributed to bridge the gap between theory-driven and data-driven methods and that it will encourage researchers to combine the strengths of choice modeling and machine learning methods.

Authorship Statement

All persons who meet authorship criteria are listed as authors, and all authors certify that they have participated sufficiently in the work to take public responsibility for the content, including participation in the concept, design, analysis, writing, or revision of the manuscript. Furthermore, each author certifies that this material or similar material has not been and will not be submitted to or published in any other publication before its appearance in the *Transportation Research Part B*.

Category 1

Conception and design of study: Brian Siffringer, Virginie Lurkin, Alexandre Alahi. acquisition of data: Brian Siffringer. analysis and/or interpretation of data: Brian Siffringer, Virginie Lurkin, Alexandre Alahi.

Category 2

Drafting the manuscript: Brian Siffringer, Virginie Lurkin, Alexandre Alahi. revising the manuscript critically for important intellectual content: Brian Siffringer, Virginie Lurkin, Alexandre Alahi.

Category 3

Approval of the version of the manuscript to be published (the names of all authors must be listed): Brian Siffringer, Virginie Lurkin, Alexandre Alahi.

Appendix A. Notation

In this section, we give a very short explanation on the main notation rule used. Variables written as plain text with or without subscripts are single elements, while their vectorized form will be in **bold** and have the relevant subscript disappear.

Here are two examples:

$$u_{in} \sim (1 \times 1) \quad (A.1)$$

$$\mathbf{u}_n \sim (I \times 1) \quad \text{or} \quad V_i \sim (1 \times 1) \quad (A.2)$$

$$\mathbf{u}_i \sim (1 \times N) \quad \mathbf{V} \sim (I \times 1) \quad (A.3)$$

$$\mathbf{u} \sim (I \times N) \quad (A.4)$$

Appendix B. Synthetic Annex Experiments

B1. Complex correlation patterns

In this section, we make use of the same data generation process defined in Guevara (2015). This allows us to take away one main observation: we show that a complexly correlated variable in \mathcal{Q} with a variable in \mathcal{X} does not negatively affect the estimated parameters. In other words, the endogeneity created by the omitted variable can be fixed when including it in the representation term without creating biasing effects. Therefore, together with other experiments of Section 5.2, we show our model fixes for omitted variable and function misspecification bias.

The data generation process defines two utility functions where:

$$U_{in} = -2p_{in} + a_{in} + b_{in} + q_{in} + \varepsilon_{in} \quad \text{for } i = 1, 2 \quad (B.1)$$

with p_{in} , a_{in} , b_{in} , and q_{in} being the generated variables and ε_{in} is the random term. The complex correlation is between q_{in} and p_{in} such that:

$$p_{in} = 5 + q_{in} + z_{in} + 0.03wz_{in} + \varepsilon_{pin} \quad \text{for } i = 1, 2 \quad (B.2)$$

where all generated variables are sampled in a uniform distribution $\mathcal{U}([-2, 2])$ such that we have exactly the same data generation from the original paper.

As done in Guevara (2015), we perform a Monte Carlo simulation by generating 100 repetitions of 1000 observations and by estimating the following models:

- MNL_{True} : The true logit model where $\mathcal{X} = \{\mathbf{a}, \mathbf{b}, \mathbf{q}\}$
- $\text{L-MNL}_{\text{True}}$: Complex correlated model between both sets where $\mathcal{X} = \{\mathbf{a}, \mathbf{b}\}$ and $\mathcal{Q} = \{\mathbf{q}\}$. The representation term arises from a single DNN layer ($L = 1$) of 100 neurons ($H = 100$).
- MNL_{endo} : The endogenous logit model where $\mathcal{X} = \{\mathbf{a}, \mathbf{b}\}$

All models have reach a stable optimum after 50 epochs, with standard Adam optimizer and 20% dropout regularizer. Running time in this framework have no noticeable difference between models.

The results of the experiment can be seen on Fig. B.1, where we have Box-Plots for the ratios of the estimators $\hat{\beta}_p/\hat{\beta}_a$. This value is studied since correction of endogeneity may produce a change in scale of the estimators, and we wish to remain comparable to the original paper and their multiple models. Indeed, we obtain the same distribution as in Guevara (2015) for the true model as well as the endogenous one. For the L-MNL models, we may observe that a complex correlated term in the neural network for $\text{L-MNL}_{\text{true}}$ does not affect the parameter estimates, and the overall architecture is able to estimate the true function.

B2. Impact of unobserved variables

We take advantage of synthetic data to better analyze the impact of unobserved variables on parameter estimates. To do so, we assume that we know the following true utility specification for $i = 1, 2$:

$$V'_{in} = V_{in} + \beta_u \cdot u_{in} \quad (B.3)$$

where V_{in} is given by Eq. (21).

We assume that variables $u_{1n}, u_{2n} \sim \mathcal{U}([-1, 1])$ are unobserved by the modeler, i.e., that $\mathbf{u} \notin (\mathcal{X} \cup \mathcal{Q})$. We also choose $\beta_u = 1$ and perform 100 experiments with this new utility specification. For the sampling scheme, we generate 1,000 synthetic observations for the training set and 200 more for the testing set.

Our new choice model is defined as

- $\text{L-MNL}(100, \mathcal{X}, \mathcal{Q})$, with $\mathcal{X} = \{\mathbf{p}, \mathbf{a}, \mathbf{b}\}$, and $\mathcal{Q} = \{\mathbf{q}, \mathbf{c}\}$,

and is compared with the following benchmarking model:

- $\text{Logit}(\mathcal{X}_1)$, with $\mathcal{X}_1 = \{\mathbf{p}, \mathbf{a}, \mathbf{b}, \mathbf{q}, \mathbf{c}\}$.

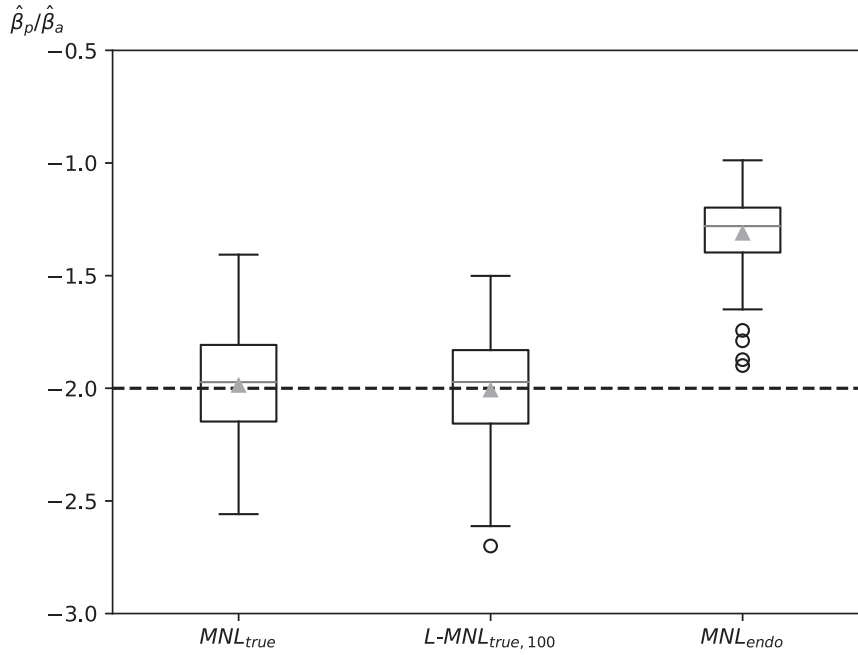


Fig. B.1. Monte Carlo experiment for complex correlation analysis, Box-Plots for 100 parameter estimates repetitions of 1000 observations each. True parameter ratio is $\beta_p/\beta_a = -2$.

Table B.1

Impact of unseen variables on parameter estimates and log-likelihood. \bar{e} is the average relative error, s.d. its standard deviation and β_p, β_a are from Eq. (21).

Model	\bar{e}_{β_p}	s.d.(e_{β_p})	\bar{e}_{β_a}	s.d.(e_{β_a})	$\bar{e}_{\beta_p/\beta_a}$	s.d.(e_{β_p/β_a})
L-MNL(25, \mathcal{X}, \mathcal{Q})	11.2	± 7.4	21.1	± 13.6	16.2	± 12.5
Logit(\mathcal{X}_1)	34.2	± 6.3	38.7	± 16.2	20.9	± 15.5
	\bar{LL}_{train}	s.d.(LL)	\bar{LL}_{test}	s.d.(LL)		
L-MNL(25, \mathcal{X}, \mathcal{Q})	-460	± 16	-101	± 7		
Logit(\mathcal{X}_1)	-620	± 11	-124	± 5		

The Monte Carlo mean relative errors for the two models are shown in Table B.1.

We see that, compared to the results depicted in Table 2 where all variables in the *true* utility specification had been included in the models, the unseen variables lead, for both models, to a decrease in the fit and an increase in relative errors in parameter estimates. Nevertheless, one important observation is that our L-MNL retrieves consistent ratio estimates, similarly to the MNL model, while having better global performances. The unseen variables do not provoke unexpected behavior to our new model.

B3. Optimization strategy

As depicted in Fig. 2, our L-MNL architecture contains two parts: the top of the figure represents the *a priori* defined component in the utility specification, while the bottom depicts the learning component. The number of parameters to be estimated in each part can vary significantly. While a limited number of parameters are generally included in linear specification, a neural network can easily contain thousands of parameters. Given this architecture, one could be tempted to estimate the model sequentially. To show the importance of jointly estimating the parameters, we consider the three following strategies for estimating our L-MNL(100, \mathcal{X}, \mathcal{Q}):

- 1) First optimizing the standard discrete choice component of the specification (i.e., the β parameters) and then learning the representation term after fixing the previously found linear-in-parameter estimates.
- 2) First optimizing the representation term (i.e., the \mathbf{w} weights) and then learning the modeled specification after fixing the previously estimated weights.
- 3) Optimizing jointly both components.

Results are shown in Table B.2 for $\beta_p = -2$, $\beta_a = 1$, $\beta_b = 0.5$ and $\beta_{qc} = 1$. We see that the joint optimization allows for the best minima in both likelihood and parameter estimation. Starting with the modeled specification gives the same pa-

Table B.2

Values of parameter estimates and likelihoods based on optimization strategy for $\beta_p = -2$ and $\beta_a = 1$. Best results are obtained with joint optimization.

Strategy	$\hat{\beta}_p$	$\hat{\beta}_a$	$\mathcal{L}(\hat{\beta})$	$\mathcal{L}_{test}(\hat{\beta})$
(1) $\hat{\beta}$ then \mathbf{w}	-1.41	0.75	-4927	-942
(2) \mathbf{w} then $\hat{\beta}$	-1.59	0.82	-3924	-758
(3) $\hat{\beta}$ and \mathbf{w}	-1.95	1.0	-3678	-721

rameters as if it were an MNL, but ends up with a better likelihood as the NN component then only increases predictability. The second strategy, on the other hand, reaches a sub-optimal minima when learning the representation. It is only with joint optimization that all important explanatory variables are expressed. The two components complete each other to achieve the best prediction performance with the correct parameter values.

Appendix C. Semi-Synthetic Data

In this section, we challenge our L-MNL model by analyzing its effectiveness outside of fully synthetic data and in the presence of strong non-linearities and noisy data. To generate our semi-synthetic data, we follow the same procedure as in Section 5.2, but instead of using normally distributed explanatory variables, we randomly select them from a real world dataset, the Swissmetro dataset (Bierlaire et al., 2001).

We define the following utility specifications:

$$\begin{aligned} V_{Train} &= l_{Train} & +1 \cdot DEST^3 \cdot AGE & -1 \cdot AGE^{0.5} \cdot ORIGIN \\ V_{SM} &= l_{SM} & +1 \cdot DEST \cdot AGE & +3 \cdot INCOME^5 \cdot PURPOSE^2 \\ V_{Car} &= l_{Car} & +5 \cdot AGE \cdot INCOME^5 & +2 \cdot ORIGIN^2 \cdot INCOME^5 \end{aligned}$$

with

$$l_i = -1 \cdot TT_i - 2 \cdot TC_i \quad \forall i \in \mathcal{C}, \quad (\text{C.2})$$

The coefficients are chosen in order to have a balanced dataset, and the interacting variables are the categorical features of Swissmetro data (Bierlaire et al., 2001) and are described in Table 5. Note that we have chosen to use power series to complexify the non-linear terms.

The models under study are:

- Logit(\mathcal{X}_a) with linear-in parameters utility specification based on the following features:
 - $\mathcal{X}_{Train,a} = \{1, TT_{Train}, TC_{Train}, AGE, DEST, ORIGIN\}$
 - $\mathcal{X}_{SM,a} = \{1, TT_{SM}, TC_{SM}, AGE, DEST, INCOME, PURPOSE\}$
 - $\mathcal{X}_{Car,a} = \{1, TT_{Car}, TC_{Car}, AGE, ORIGIN, INCOME\}$
- Logit(\mathcal{X}_b) with only travel time and cost for each utility, i.e.:
 - $\mathcal{X}_{ib} = \{1, TT_i, TC_i\}$ for all $i \in \mathcal{C}$.
- L-MNL(\mathcal{X}, \mathcal{Q}) with
 - $\mathcal{X}_i = \{TT_i, TC_i\}$,
 - $\mathcal{Q} = \{AGE, DEST, ORIGIN, INCOME, PURPOSE\}$

The results can be seen in Table C.1. We see that standard MNL models are unable to retrieve the correct parameter estimates for utility specifications with important non-linearities. Both MNL models exhibit large relative errors (about 40% for the Logit(\mathcal{X}_b) and at least 25% for the Logit(\mathcal{X}_a)). The relative errors are also large for the ratio of parameters, which would lead to wrong postestimation indicator, the VOT in this case. Unlike the Logit models, our L-MNL model recovers the true estimates in both parameters and ratio, while achieving a much better fit. We therefore conclude that the representation term was able to learn the complex non-linearities and that ignoring these non-linearities lead to models that greatly suffer from underfit.

Table C.1

Values of parameter estimates and likelihoods for different models based on Eq. (C.1). Ground truth is $\beta_{TT} = -1$ and $\beta_{TC} = -2$. Only L-MNL is able to estimate correctly the parameters.

Models	$\hat{\beta}_{TT}$	$\hat{\beta}_{TC}$	$\hat{\beta}_{TC}/\hat{\beta}_{TT}$	LL_{train}	LL_{test}
Logit(\mathcal{X}_a)	-0.65	-1.63	2.50	-5412	-1354
Logit(\mathcal{X}_b)	-0.25	-0.70	2.81	-7722	-1925
L-MNL(\mathcal{X}, \mathcal{Q})	-1.01	-1.99	1.96	-2516	-809

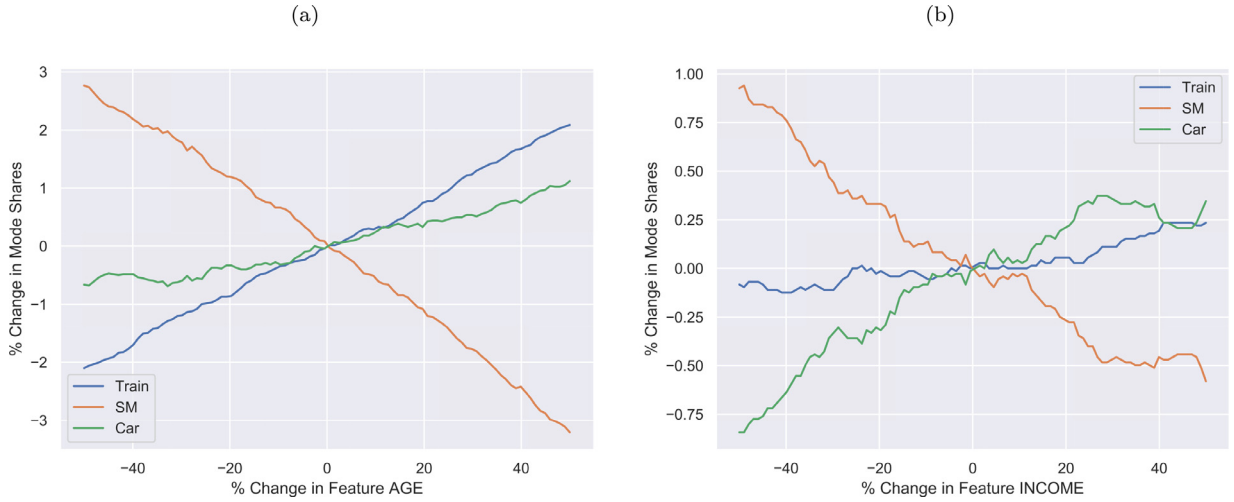


Fig. D.1. Percentage change in total mode shares against percentage change in L-MNL features belonging to \mathcal{Q}_2 . These general trends appear by fixing all other values as constant.

Appendix D. Swissmetro supplementary experiment

D1. Feature impact and sensitivity analysis

As previously done by [Bentz and Merunka \(2000\)](#), we finish our experiments by investigating what the neural network component has learned through the study of a sensitivity analysis. To do so, we vary the value of a feature in \mathcal{Q} while keeping the others variables constant, and we analyze its impact on the utilities and market shares of the alternatives. We do the analysis on the L-MNL(100, \mathcal{X}_2 , \mathcal{Q}_2) for which only *cost*, *time*, and *frequency* are included in the linear specification, while the other 14 variables are given to the neural network. [Fig. D.1](#) presents a sensitivity analysis for two variables in \mathcal{Q}_2 : *AGE* and *INCOME*¹⁰. We observe that the *AGE* variable has almost linear relations to the utilities, which has also been seen in [Bierlaire et al. \(2001\)](#)'s benchmark. Changing *INCOME* however, seems to present non-linearities and an overall weaker impact on the change in mode share. We recognize that this is only the average behavior of the feature in the population when keeping all other variables constant. Further investigation of non-linear interactions can be done by separating our sensitivity analysis based on the values of the other variables as seen in [Bentz and Merunka \(2000\)](#).

To have some insights on the impact of each feature in \mathcal{Q} on the utility function, we get inspiration from saliency maps ([Simonyan et al., 2013](#)). A saliency map is obtained through back-propagation of an observation's prediction score, as opposed to its output loss which is performed during training. We then read the results at the nodes of the input layer. The retrieved values are considered to be the gradient estimation of a prediction with respect to a given input. For our case, we do this by changing the loss function of our pre-trained model:

$$\text{loss}(\mathbf{V}_n, \mathbf{y}_n) = \sum_{i \in \mathcal{C}_n} y_{in} \cdot V_{in}, \quad (\text{D.1})$$

where y_{in} is 1 when individual n is predicted to choose alternative i and V_{in} is the output for utility i as seen in [Eq. \(11\)](#).

As opposed to an image, which is a 2D set of pixels, the position of our input has always the same meaning for each individual. In other words, the gradient read on the first position will always be *PURPOSE* and the last always *SM_SEATS*. This allows us to measure the average impact of a feature on each class. Indeed, when summing over all individuals the absolute gradients at the input layer for each feature, we get the [Fig. D.2](#). The sum has been separated for each utility and normalized by the count of the chosen alternative of all individuals.

As we can see, all variables are being used by the NN. Some, such as *GA*, *AGE* or *LUGGAGE* seem to have an overall bigger impact than *SM_SEATS*. This supports the conclusion that the MNL benchmark of [Bierlaire et al. \(2001\)](#) misses potential useful information by ignoring many variables.

¹⁰ These two variables were chosen for elasticity study as they are the only integer variables which represent a discrete scale of intensity.

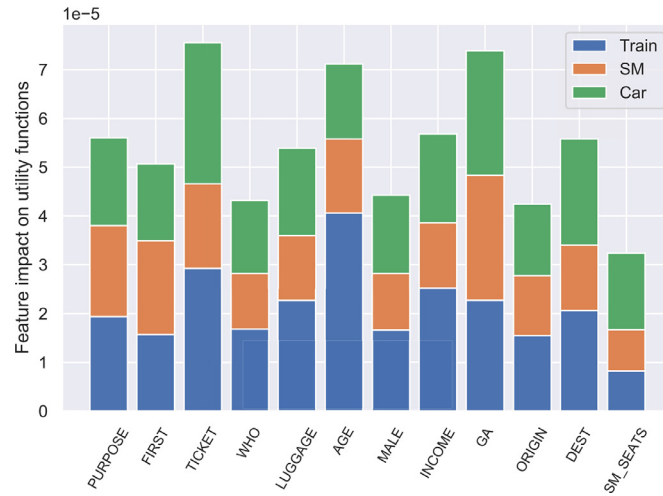


Fig. D.2. Mean feature contributions to each utility obtained with loss of Eq. (D.1) and gradient evaluation on the input layer.

Appendix E. Learning Nested-Logit: network-loss architecture

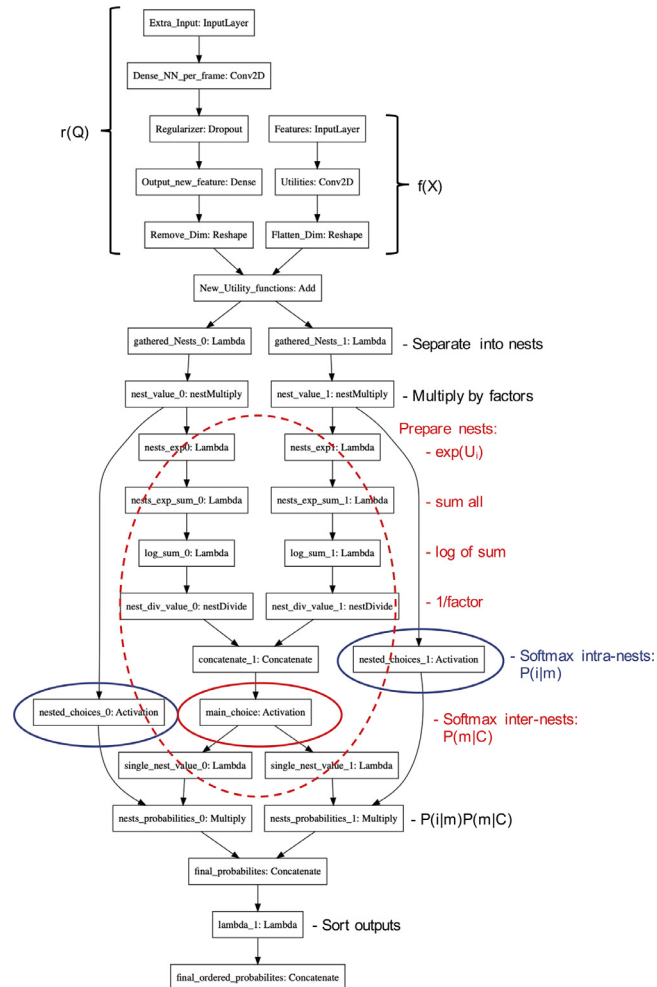


Fig. E.1. For 2 nests. As the nested loss needs trainable weights, this was solved using custom layers. "nestMultiply" and "nestDivide" share the same trainable parameter μ and has constraint $\mu \geq 1$.

[illegible]

Appendix H. Correlation coefficients among variables in Optima dataset

Table H.1

Correlation coefficients among variables between sets \mathcal{X} and \mathcal{Q} included in Optima for our training dataset.

	age	HouseType	Gender	Income	Educ.	SocioCat	House	Famil.
TimePT	-0.07	0.04	-0.03	0.07	0.09	-0.01	0.09	0.08
TimeCar	-0.00	0.00	-0.06	0.04	0.07	-0.03	0.07	0.08
CostPT	-0.02	0.02	-0.06	0.04	0.06	-0.03	0.08	0.06
distance	0.01 -0.05	-0.01	-0.25	0.03	0.06	-0.04	0.06	0.07
CostCarCHF	0.01	-0.01	-0.05	0.03	0.07	-0.04	0.06	0.07
TripPurpose	0.23	-0.02	0.08	-0.11	0.03	0.06	-0.04	-0.09
NbBicy	-0.38	-0.18	0.03	0.24	0.05	-0.01	-0.09	0.35
NbCar	-0.16	-0.17	-0.04	0.24	-0.04	-0.04	-0.08	0.20
NbChild	-0.41	-0.07	0.07	0.13	0.09	0.01	-0.02	0.17
GenAbST	0.06	0.06	0.04	-0.00	-0.06	0.02	-0.01	-0.14

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:[10.1016/j.trb.2020.08.006](https://doi.org/10.1016/j.trb.2020.08.006).

References

- Abe, M., 1999. A generalized additive model for discrete-choice data. *J. Business. Econ. Stat.* 17 (3), 271–284.
- Ackley, D.H., Hinton, G.E., Sejnowski, T.J., 1985. A learning algorithm for boltzmann machines. *Cogn. Sci.* 9 (1), 147–169.
- Agrawal, D., Schorling, C., 1996. Market share forecasting: an empirical comparison of artificial neural networks and multinomial logit model. *J. Retail.* 72 (4), 383–407.
- Bentz, Y., Merunka, D., 2000. Neural networks and the multinomial logit for brand choice modelling: a hybrid approach. *J. Forecast.* 19 (3), 177–200.
- Berger, J.O., Liseo, B., Wolpert, R.L., 1999. Integrated likelihood methods for eliminating nuisance parameters. *Stat. Sci.* 14 (1), 1–22.
- Bierlaire, M., 2003. Biogeme: a free package for the estimation of discrete choice models. In: *Swiss Transport Research Conference*.
- Bierlaire, M., Axhausen, K., Abay, G., 2001. The acceptance of modal innovation: The case of swissmetro (TRANSP-OR-CONF-2006-055).
- Bierlaire, M., Curchod, A., Danalet, A., Doyen, E., Faure, P., Glerum, A., Kaufmann, V., Tabaka, K., Schuler, M., 2011. *Projet de recherche sur la mobilité combinée, Rapport définitif de l'enquête de préférences révélées*. Technical Report.
- Bishop, C.M., 1995. *Neural Networks for Pattern Recognition*. Oxford university press.
- Borghesi, A., Baldo, F., Milano, M., 2020. Improving deep learning models via constraint-based domain knowledge: a brief survey. *arXiv preprint arXiv:2005.10691*.
- Brathwaite, T., Vij, A., Walker, J.L., 2017. Machine learning meets microeconomics: the case of decision trees and discrete choice. Working paper arXiv preprint arXiv:1711.04826.
- Breiman, L., 1996. Bagging predictors. *Mach. Learn.* 24 (2), 123–140.
- Breiman, L., 2001. Random forests. *Mach. Learn.* 45 (1), 5–32.
- Cantarella, G.E., de Luca, S., 2005. Multilayer feedforward networks for transportation mode choice analysis: an analysis and a comparison with random utility models. *Transp. Res. Part C* 13 (2), 121–155.
- Chang, X., Wu, J., Liu, H., Yan, X., Sun, H., Qu, Y., 2019. Travel mode choice: a data fusion model using machine learning methods and evidence from travel diary survey data. *Transportmetrica A* 15 (2), 1587–1612.
- Chollet, F., et al., 2015. Keras. <https://github.com/fchollet/keras>.
- Cortes, C., Vapnik, V., 1995. Support-vector networks. *Mach. Learn.* 20 (3), 273–297.
- Cramer, J.S., 2007. Robustness of logit analysis: unobserved heterogeneity and mis-specified disturbances. *Oxf. Bull. Econ. Stat.* 69 (4), 545–555.
- van Cranenburgh, S., Alwosheel, A., 2019. An artificial neural network based approach to investigate travellers' decision rules. *Transp. Res. Part C* 98, 152–166.
- Dougherty, M., 1995. A review of neural networks applied to transport. *Transp. Res. Part C* 3 (4), 247–260.
- Draxler, F., Veschini, K., Salmhofer, M., Hamprecht, F.A., 2018. Essentially no barriers in neural network energy landscape. *arXiv preprint arXiv:1803.00885*.
- Faghri, A., Hua, J., 1992. Evaluation of artificial neural network applications in transportation engineering. *Transp. Res. Rec.* 1358, 71.
- Fernández-Antolín, A., Guevara, C.A., De Lapparent, M., Bierlaire, M., 2016. Correcting for endogeneity due to omitted attitudes: empirical assessment of a modified mis method using rp mode choice data. *J. Choice Modell.* 20, 1–15.
- Friedman, J.H., 2001. Greedy function approximation: a gradient boosting machine. *Ann. Stat.* 1189–1232.
- Garipov, T., Izmailov, P., Podoprikin, D., Vetrov, D.P., Wilson, A.G., 2018. Loss surfaces, mode connectivity, and fast ensembling of dnns. In: *Advances in Neural Information Processing Systems*, pp. 8789–8798.
- Glerum, A., Atasoy, B., Bierlaire, M., 2014. Using semi-open questions to integrate perceptions in choice models. *J. Choice Modell.* 10, 11–33.
- Golshani, N., Shabanpour, R., Mahmoudifard, S.M., Derrible, S., Mohammadian, A., 2018. Modeling travel mode and timing decisions: comparison of artificial neural networks and copula-based joint model. *Travel Behav. Soc.* 10, 21–32.
- Guevara, C.A., 2015. Critical assessment of five methods to correct for endogeneity in discrete-choice models. *Transp. Res. Part A* 82, 240–254.
- Hagenauer, J., Helbich, M., 2017. A comparative study of machine learning classifiers for modeling travel mode choice. *Expert. Syst. Appl.* 78, 273–282.
- Han, Y., Zegras, C., Pereira, F.C., Ben-Akiva, M., 2020. A neural-embedded choice model: tastenet-mnl modeling taste heterogeneity with flexibility and interpretability. *arXiv preprint arXiv:2002.00922*.
- He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.
- Hensher, D.A., Ton, T.T., 2000. A comparison of the predictive potential of artificial neural networks and nested logit models for commuter mode choice. *Transp. Res. Part E* 36 (3), 155–172.
- Hruschka, H., 2007. Using a heterogeneous multinomial probit model with a neural net extension to model brand choice. *J. Forecast.* 26 (2), 113–127.
- Hruschka, H., Fettes, W., Probst, M., 2004. An empirical comparison of the validity of a neural net based multinomial logit choice model to alternative model specifications. *Eur. J. Oper. Res.* 159 (1), 166–180.

- Hruschka, H., Fettes, W., Probst, M., Mies, C., 2002. A flexible brand choice model based on neural net methodology a comparison to the linear utility multinomial logit model and its latent class extension. *OR Spectrum* 24 (2), 127–143.
- Iranitalab, A., Khattak, A., 2017. Comparison of four statistical and machine learning methods for crash severity prediction. *Accid. Anal. Prevent.* 108, 27–36.
- Hillel, T., Bierlaire, M., Elshafie, M., Jin, Y., 2020. A systematic review of machine learning classification methodologies for modelling passenger mode choice. *J. Choice Modell.*
- Karlaftis, M.G., Vlahogianni, E.I., 2011. Statistical methods versus neural networks in transportation research: differences, similarities and some insights. *Transp. Res. Part C* 19 (3), 387–399.
- Kim, J., Rasouli, S., Timmermans, H., 2016. A hybrid choice model with a nonlinear utility function and bounded distribution for latent variables: application to purchase intention decisions of electric cars. *Transp. A* 12 (10), 909–932.
- Kingma, D.P., Ba, J., 2014. Adam: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Kneib, T., Baumgartner, B., Steiner, W.J., 2007. Semiparametric multinomial logit models for analysing consumer choice behaviour. *ASTA Adv. Stat. Anal.* 91 (3), 225–244.
- Krogh, A., Hertz, J.A., 1992. A simple weight decay can improve generalization. In: *Advances in neural information processing systems*, pp. 950–957.
- Lee, D., Derrible, S., Pereira, F.C., 2018. Comparison of four types of artificial neural network and a multinomial logit model for travel mode choice modeling. *Transportation Research Record*, 0361198118796971.
- Lee, L.-F., 1982. Specification error in multinomial logit models: analysis of the omitted variable bias. *J. Econom.* 20 (2), 197–209.
- Lhéritier, A., Bocamazo, M., Delahaye, T., Acuna-Agost, R., 2018. Airline itinerary choice modeling using machine learning. *J. Choice Modell.*
- Li, Q., Lin, T., Shen, Z., 2019. Deep learning via dynamical systems: an approximation perspective. *arXiv preprint arXiv:1912.10382*.
- Liao, B., Zhang, J., Cai, M., Tang, S., Gao, Y., Wu, C., Yang, S., Zhu, W., Guo, Y., Wu, F., 2018. Dest-resnet: a deep spatiotemporal residual network for hotspot traffic speed prediction. In: *Proceedings of the 26th ACM International Conference on Multimedia*, pp. 1883–1891.
- Likert, R., 1932. A technique for the measurement of attitudes. *Arch. Psychol.*
- McFadden, D., 1974. The measurement of urban travel demand. *J. Public Econ.* 3 (4), 303–328.
- McFadden, D., 1978. Modeling the choice of residential location. *Transp. Res. Rec.* (673).
- Mohammadian, A., Miller, E., 2002. Nested logit models and artificial neural networks for predicting household automobile choices: comparison of performance. *Transp. Res. Record* (1807) 92–100.
- Murdoch, W.J., Singh, C., Kumbier, K., Abbasi-Asl, R., Yu, B., 2019. Definitions, methods, and applications in interpretable machine learning. *Proc. Natl. Acad. Sci.* 116 (44), 22071–22080.
- Nam, D., Kim, H., Cho, J., Jayakrishnan, R., 2017. A model based on deep learning for predicting travel mode choice. In: *Proceedings of the Transportation Research Board 96th Annual Meeting Transportation Research Board*, Washington, DC, USA, pp. 8–12.
- Omrani, H., 2015. Predicting travel mode of individuals by machine learning. *Transp. Res. Procedia* 10, 840–849.
- Otsuka, M., Osogami, T., 2016. A deep choice model. In: *AAAI*, pp. 850–856.
- Paredes, M., Hemberg, E., O'Reilly, U.-M., Zegras, C., 2017. Machine learning or discrete choice models for car ownership demand estimation and prediction? In: *Models and Technologies for Intelligent Transportation Systems (MT-ITS), 2017 5th IEEE International Conference on*. IEEE, pp. 780–785.
- Pekel, E., Soner Kara, S., 2017. A comprehensive review for artificial neural network application to public transportation. *Sigma: Journal of Engineering & Natural Sciences/Mühendislik ve Fen Bilimleri Dergisi* 35 (1).
- Pirra, M., Diana, M., 2018. A study of tour-based mode choice based on a support vector machine classifier. *Transp. Plann. Technol.* 1–14.
- Ribeiro, M.T., Singh, S., Guestrin, C., 2016. "why should i trust you?" explaining the predictions of any classifier. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144.
- Rish, I., et al., 2001. An empirical study of the naive bayes classifier. In: *IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence*, 3. IBM New York, pp. 41–46.
- von Rueden, L., Mayer, S., Beckh, K., Georgiev, B., Giesselbach, S., Heese, R., Kirsch, B., Pfrommer, J., Pick, A., Ramamurthy, R., et al., 2020. Informed machine learning—a taxonomy and survey of integrating knowledge into learning systems. *arXiv preprint arXiv:1903.12394*.
- Samek, W., Binder, A., Montavon, G., Lapuschkin, S., Müller, K.-R., 2016. Evaluating the visualization of what a deep neural network has learned. *IEEE Trans. Neural Netw. Learn. Syst.* 28 (11), 2660–2673.
- Sayed, T., Razavi, A., 2000. Comparison of neural and conventional approaches to mode choice analysis. *J. Comput. Civil Eng.* 14 (1), 23–30.
- Schindler, M., Baumgartner, B., Hruschka, H., 2007. Nonlinear effects in brand choice models: comparing heterogeneous latent class to homogeneous non-linear models. *Schmalenbach Bus. Rev.* 59 (2), 118–137.
- Shannon, C.E., 1948. A mathematical theory of communication. *Bell Syst. Tech. J.* 27 (3), 379–423.
- Shen, J., 2009. Latent class model or mixed logit model? a comparison by transport mode choice data. *Appl. Econ.* 41 (22), 2915–2924.
- Shmueli, D., Salomon, I., Shefer, D., 1996. Neural network analysis of travel behavior: evaluating tools for prediction. *Transp. Res. Part C* 4 (3), 151–166.
- Shrikumar, A., Greenside, P., Kundaje, A., 2017. Learning important features through propagating activation differences. 1704.02685.
- Simonyan, K., Vedaldi, A., Zisserman, A., 2013. Deep inside convolutional networks: visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.
- Torres, C., Hanley, N., Riera, A., 2011. How wrong can you be? Implications of incorrect utility function specification for welfare measurement in choice experiments. *J. Environ. Econ. Manage.* 62 (1), 111–121.
- Van Der Pol, M., Currie, G., Kromm, S., Ryan, M., 2014. Specification of the utility function in discrete choice experiments. *Value Health* 17 (2), 297–301.
- Vij, A., Carrel, A., Walker, J.L., 2013. Incorporating the influence of latent modal preferences on travel mode choice behavior. *Transp. Res. Part A* 54, 164–178.
- Vovsha, P., 1997. Application of cross-nested logit model to mode choice in tel aviv, israel, metropolitan area. *Transp. Res. Rec.* 1607 (1), 6–15.
- Wang, B., Yuan, B., Shi, Z., Osher, S.J., 2018. Enresnet: resnet ensemble via the feynman-kac formalism. *arXiv preprint arXiv:1811.10745*.
- Wang, S., Mo, B., Zhao, J., 2020. Deep neural networks for choice analysis: architecture design with alternative-specific utility functions. *Transp. Res. Part C* 112, 234–251.
- Wang, S., Wang, Q., Bailey, N., Zhao, J., 2018b. Deep neural networks for choice analysis: a statistical learning theory perspective. 1810.10465.
- Wang, S., Wang, Q., Zhao, J., 2018c. Deep neural networks for choice analysis: extracting complete economic information for interpretation. 1812.04528.
- West, P.M., Brockett, P.L., Golden, L.L., 1997. A comparative analysis of neural networks and statistical methods for predicting consumer choice. *Market. Sci.* 370–391.
- Williams, H.C., 1977. On the formation of travel demand models and economic evaluation measures of user benefit. *Environ. Plann. A* 9 (3), 285–344.
- Wong, M., Farooq, B., 2019. Reslogit: a residual neural network logit model. *arXiv preprint arXiv:1912.10058*.
- Wong, M., Farooq, B., 2020. A bi-partite generative model framework for analyzing and simulating large scale multiple discrete-continuous travel behaviour data. *Transp. Res. Part C* 110, 247–268.
- Wong, M., Farooq, B., Bilodeau, G.-A., 2018. Discriminative conditional restricted boltzmann machine for discrete choice and latent variable modelling. *J. Choice Modell.* 29, 152–168.
- Xiong, Y., Mannering, F.L., 2013. The heterogeneous effects of guardian supervision on adolescent driver-injury severities: a finite-mixture random-parameters approach. *Transp. Res. Part B* 49, 39–54.
- Zhao, X., Yan, X., Yu, A., Van Hentenryck, P., 2020. Prediction and behavioral analysis of travel mode choice: a comparison of machine learning and logit models. *Travel Behav. Soc.* 20, 22–35.
- Zhou, X., Wang, M., Li, D., 2019. Bike-sharing or taxi? modeling the choices of travel mode in chicago using machine learning. *J. Transp. Geogr.* 79, 102479.