# Highlights

**Functional effects models: Accounting for preference heterogeneity in panel data with machine learning**

Nicolas Salvadé, Tim Hillel

- Introduces the functional effects model, using Machine Learning methodologies to account for inter-individual heterogeneity.

- Functional Effects Models learn individual-specific parameters/preferences from socio-demographic characteristics.

- Verifies that true functional effects can be recovered on a synthetic experiment.

- Provides a thorough comparison between Functional Effects Models and machine learning and statistical models.

- Case study on the easySHARE, Swissmetro, and LPMC datasets.

# Functional effects models: Accounting for preference heterogeneity in panel data with machine learning

Nicolas Salvadé[a], Tim Hillel[a,*]

[a]*Department of Civil, Environmental and Geomatic Engineering, University College London, Gower Street, London, WC1E 6BT, United Kingdom*

## Abstract

In this paper, we present a general specification for Functional Effects Models, which use Machine Learning (ML) methodologies to learn individual-specific preference parameters from socio-demographic characteristics, therefore accounting for inter-individual heterogeneity in panel choice data. This approach exploits the generalisation power of gradient-based ML regression techniques to account for inter-individual heterogeneity in sequential choices. We identify three specific advantages of the Functional Effects Model over traditional fixed, and random/mixed effects models: (i) by mapping individual-specific effects as a function of socio-demographic variables, we can account for these effects when forecasting choices of previously unobserved individuals (ii) the (approximate) maximum-likelihood estimation of functional effects avoids the incidental parameters problem of the fixed effects model, even when the number of observed choices per individual is small; and (iii) we do not rely on the strong distributional assumptions of the random effects model, which may not match reality. We learn functional intercept and functional slopes with powerful non-linear machine learning regressors for tabular data, namely gradient boosting decision trees and deep neural networks. We validate our proposed methodology on a synthetic experiment and three real-world panel case studies, demonstrating that the Functional Effects Model: (i) can identify the true values of individual-specific effects when the data generation process is known; (ii) outperforms both state-of-the-art ML choice modelling techniques that omit individual heterogeneity in terms of predictive performance, as well as traditional static panel choice models in terms of learning inter-individual heterogeneity. The results indicate that the FI-RUMBoost model, which combines the individual-specific constants of the Functional Effects Model with the complex, non-linear utilities of RUMBoost, performs marginally best on large-scale revealed preference panel data.

*Keywords:* Panel Data, Machine Learning, Choice Modelling, Mode Choice, Ordinal Regression

---

*Corresponding author

*Email addresses:* `nicolas.salvade.22@ucl.ac.uk` (Nicolas Salvadé), `tim.hillel@ucl.ac.uk` (Tim Hillel)

## 1. Introduction

Human choices are immensely complex and are inherently linked to observed variables, such as the cost of an alternative, and unobserved variables, such as life experiences and preferences. Consecutive choices made by the same individual share these unobserved variables and will be inevitably correlated. This violates the assumption of independence between observations, a common assumption in Machine Learning (ML) models and basic statistical models such as the Multinomial Logit (MNL) model. Therefore, these types of choice experiments, referred to as *panel* or *longitudinal* studies, require specific methodologies to address the correlation between observations. Common problems dealing with panel data include, for example, longitudinal health studies, household income variation over time studies, stated preference experiments where respondents are being asked to make several choices consecutively, or even time series analysis.

Traditionally, practitioners model preference parameters with two types of choice models based on the random utility theory: (i) static models; and (ii) dynamic models. For a detailed discussion, see Greene (2015). These models differ in how they account for the serial correlation between the error terms of the same individual. The first type models preference parameters with either fixed effects, i.e., individual-specific constants (or intercepts), estimated from maximum likelihood of observed choices, or random, or mixed, effects, i.e., random parameters distributed across the population[1]. However, the fixed effects model suffer from the incidental parameters problem, where the parameters are inconsistent when the number of observations per individual is small, and is typically only used in regression/ordinal tasks (Greene, 2015). Therefore, in this work, we focus on the mixed effects model. Two special cases of mixed effects models are the random intercept model, with the intercept, or constant, assumed to be a random variable, and the random slopes model, where the coefficients related to a variable are assumed to be randomly distributed. This method has been first developed to identify the true values of estimated parameters, removing bias from not accounting for individual heterogeneity. However, in recent years, with the ever-increasing use of ML models, predictive tasks have become predominantly important, and the mixed effects model is limited when predicting preferences of unknown individuals. More specifically, the random effects fitted during training have to be averaged (i.e., use the population-level mean with the mean of the random effects distribution) (Krueger et al., 2021), reducing predictive power. In addition, the random effects model rely on strong distributional assumptions that need to be specified by the modeller.

The second type of models is Markov chain models, where the last observed choice made by an individual is used to account for panel effects. When handling the first observed choice of an individual, dynamic models can be thought of as static models, since there are no previous choices to use as exogenous variables. This is known in the literature as the *initial conditions problem* of dynamic models (Train, 2009). Therefore, dynamic models are

---

[1]Note, that the mixed effects model refers to each parameter being the combination of a fixed "population-level" parameter with an individual-specific random intercept (i.e., $\beta_m + u_{mn}$ where $u_{mn}$ is a random intercept), whereas the fixed effects model refers to individual-specific preference parameters (i.e., $\alpha_{in}$ which is learnt for each individual using maximum likelihood estimation). While mixed effects is a general term, in the context of panel data we distinguish between the random intercepts and random slopes models.

also not suitable to predict preferences of unknown individuals, and there is a crucial need to develop specific methodologies to make static models suitable for forecasting. Note that we emphasise the parallels between dynamic models and recommender systems, especially in the *cold start problem* (Panda and Ray, 2022), where static models could be used to generate recommendations for individuals without prior choice knowledge. Hereafter, since we are interested in panel data predictions for unobserved individuals, we focus on static models for panel data. Table 1 provides a glossary of all the terminology used in this paper.

Table 1: Glossary of inputs, model types, and parameters.

| Term | Definition |
|---|---|
| *1. Inputs / Data* | |
| Panel / longitudinal data | Data containing repeated choices from individuals. |
| Target variable $(y_i)$ | The variable to be modelled or predicted. |
| Explanatory variables/features $(x_{imnt})$ | Observed variables indexed by $i$ (alternative), $m$ (variable), $n$ (individual), and $t$ (time). |
| Socio-demographic characteristics $(s_n)$ | Characteristics of individual $n$, such as demographic or economic background. |
| | |
| *2. Model Types* | |
| Static models | Models accounting preferences without prior choices knowledge. |
| Dynamic models | Models including prior choices as variables to model preferences. |
| Fixed effects model | Model preferences with individual-specific intercepts. |
| Mixed / random effects model | Model preferences with random variables drawn from a distribution. |
| Random intercept model | A mixed effects model where only the intercept varies randomly across individuals. |
| Random slope model | A mixed effects model where slopes of features vary randomly across individuals. |
| Random intercept and slope model | A mixed effects model with both random intercepts and random slopes. |
| Functional effect model | Model where preferences are learnt from $s_n$. |
| Functional intercept model with linear coefficients | Functional intercept depending on $s_n$ while slopes remain constant (linear). |
| Functional intercept model with non-linear coefficients | Functional intercept depending on $s_n$ while slopes are a non-linear function of the features. |
| Functional slopes model | Functional slopes depending on $s_n$ while intercepts remain constant. |
| Functional intercept and slopes model | Both intercept and slopes depend on $s_n$. |
| | |
| *3. Parameters* | |
| Intercepts $(\alpha_{in})$ | Value at which a function intersects the y-axis. |
| Linear coefficients $(\beta_{im})$ | Constant marginal effects of features. |
| Non-linear coefficients $(f_{im}(x_{imnt}))$ | Non-linear functions of features learnt with a gradient-based ML regressor. |
| Functional intercept $(g_{i,0}(s_n))$ | Intercept learnt from $s_n$ with a gradient-based ML regressor. |
| Functional slopes $(g_{im}(s_n))$ | Slope of variable $m$ learnt from $s_n$ with a gradient-based ML regressor. |

There have been several attempts to adapt static models, such as the mixed effects model, with Machine Learning (ML) methodologies. The main idea is to learn the population-level mean with an out-of-the-box ML regressor and estimate random effects as in a linear mixed effects model. For example, this has been done for regression trees (Hajjem et al.,

2017; Sela and Simonoff, 2012), linear trees (Fokkema et al., 2018), Random Forests (RF) (Hajjem et al., 2014), Gradient Boosting Decision Trees (GBDT) (Sigrist, 2023), Neural Networks (NN) (Mandel et al., 2023), and Convolutional Neural Networks (CNN) (Xiong et al., 2019). All these papers use some sort of Expectation-Maximisation (EM) algorithm to iteratively estimate the population-level mean and random effects. This framework has also been generalised by Ngufor et al. (2019) and Kilian et al. (2023). However, these models face two main problems when applied in choice modelling with panel data: (i) the random effects are not suited for predictions, which means that they need to be averaged, dropped, or re-estimated at inference time at the cost of an expensive computational procedure; and (ii) they mostly rely on out-of-the-box ML models, which are not interpretable. Note that in Xiong et al. (2019), the authors decompose the output of the model as fixed and random effects, where both effects depend on input features. However, by doing so, the random effects of their model are *effectively* fixed effects and do not have any random component, falling back to an out-of-the-box CNN. Finally, it is worth mentioning that there is very little existing research into dynamic ML models in a choice modelling context.

In this paper, we use existing gradient-based ML methodologies to learn individual-specific parameters as a function of socio-demographic characteristics. We learn two types of functional effects: (i) functional intercept, effectively emulating the random intercept model; and (ii) functional slopes, effectively emulating the random slopes model. At a high level, we are using unrestricted ML regressors to impute an individual-specific constant from the socio-demographic characteristics. This constant mimics the random effect in traditional mixed effects models, but since it is a function of the socio-demographic characteristics, it can be easily used for inference. We do so with the two most popular ML regressors for tabular data, Gradient Boosting Decision Trees (GBDTs) and Deep Neural Networks (DNNs). It is worth noting that, whilst the primary contribution of the paper is a general framework for capturing individual effects for panel data, this framework incorporates and builds on several existing works in the literature, specifically: L-MNL (Sifringer et al., 2020), TasteNet-MNL (Han et al., 2022), and RUMBoost (Salvadé and Hillel, 2025). We further note that all of the models presented in this paper could similarly be used to account for individual heterogeneity in cross-sectional data, though this is not the focus of this paper. We systematically evaluate our methodology on a synthetic experiment and three real-world panel case studies.

The main strengths of our proposed methodology are:

1. incorporates individual-specific preferences in powerful ML models, resulting in improved real-world predictive performance compared to existing state-of-the-art ML-based choice models that assume homogenous preferences;

2. accounts for individual-specific effects when forecasting choices of previously unobserved individuals, which is essential for counterfactual analysis; and

3. compared to traditional static choice models, avoids the incidental parameters problem of the fixed effects model and the strong distributional assumptions of the random effects models.

All the code used in this paper, including the implementation of a generalised functional effects model learnt with GBDTs and DNNs for different datasets and choice situations, is

open source and freely available on Github[2].

The rest of the paper is structured as follows: Section 2 provides the theoretical background, and Section 3 introduces the methods used in the paper. We validate the methodology with a synthetic experiment in Section 4 and provide a thorough benchmark with three real-world case studies in Section 5. Finally, Section 6 concludes the paper.

## 2. Theoretical background

### 2.1. Choice models with homogenous preferences

Choice models based on the random utility theory assume that choice makers are rational and will choose the alternative that maximises their utility, a latent representation of their preferences. The utility function is typically composed of a deterministic part and a random part, i.e.:

$$U_{in} = V_{in} + \epsilon_{in} \tag{1}$$

where:

- $U_{in}$ is the utility function for an individual $n$, and alternative $i$;

- $V_{in}$ is the deterministic utility for an individual $n$, and alternative $i$; and

- $\epsilon_{in}$ is the error term, capturing unobserved informations.

The deterministic utility is widely represented as a *linear-in-parameter* function of the variables, i.e.;

$$V_{in} = \alpha_i + \sum_{m=1}^{M_i} \beta_{im} x_{inm} \tag{2}$$

where:

- $M_i$ is the number of variables for alternative $i$;

- $x_{inm}$ is the variable $m$ for alternative $i$ and observation $n$;

- $\beta_{im}$ are homogenous parameters to be estimated from the data $\forall i, m$; and

- $\alpha_i$ is the intercept, or constant, for alternative $i$.

Interpretable non-linear ML utility models (e.g. RUMBoost (Salvadé and Hillel, 2025)) extends the deterministic utility function specification of Eq. 2 by replacing $\beta_{im} x_{inm}$ with the output of a gradient-based ML regressor:

$$V_{in} = \alpha_i + \sum_{m=1}^{M_i} f_{im}(x_{inm}) \tag{3}$$

where:

---

[2]https://github.com/big-ucl/functional-effects-model

- $f_{im}$ is a non-parametric non-linear function learnt from a gradient-based ML regressor $\forall i, m$

Note that some researchers have attempted to interpret the output of an ML regressor as the deterministic utility:

$$V_{in} = f_i(\mathbf{x}_{in}) \tag{4}$$

where:

- $f_i$ is a non-parametric non-linear function learnt from a gradient-based ML regressor $\forall i$, where all variables can interact without restrictions.

However, $f_i$ is usually not consistent with the random utility theory and is not interpretable.

In this work, we assume the error term to be *independent and identically distributed* (*i.i.d.*), such that the probability $P_{in}$ of an individual $n$ to choose alternative $i$ with $J$ alternatives can be derived as in an MNL model, that is:

$$P_{in} = \frac{e^{V_{in}}}{\sum_{j=1}^{J} e^{V_{jn}}} \tag{5}$$

The parameters are chosen to minimise the negative Cross-Entropy Loss (CEL) function, akin to Maximum Log-Likelihood Estimation (MLE):

$$\mathcal{L} = \sum_{n=1}^{N} \sum_{j=1}^{J-1} \mathbb{1}(j = y_n) \ln(P_{jn}) \tag{6}$$

where:

- $\mathbb{1}(j = y_n)$ is 1 if $j$ equals to the observed chosen alternative $y_n$, 0 otherwise.

*2.2. Static models for panel data*

Panel data have more than one observation per individual. Therefore, we extend Equation 1 as follows:

$$U_{int} = V_{int} + \epsilon_{int} \tag{7}$$

where $t$ represents the $t^{th}$ choice of an individual $n$. Inevitably, the error term $\epsilon_{int}$ is not *i.i.d.* across $t$ and specific methodologies are required to deal with this intrinsic correlation, i.e., dynamic models. However, we omit the dynamic correlations here, as we do not address these in the current methodology, instead leaving this to further work. In order to reduce the parameter bias by accounting for inter-individual heterogeneity, the fixed effects model extends Eq. 2 as follows:

$$V_{int} = \alpha_{in} + \sum_{m=1}^{M_i} \beta_{inm} x_{inmt} \tag{8}$$

where the model parameters $\alpha_{in}$ and $\beta_{inm}$ are now individual-specific. A fixed intercept model is a model with only $\alpha_{in}$ being individual-specific, and a fixed slopes model is a model with only the slopes, or coefficients, $\beta_{inm}$ being individual-specific.

On the other hand, the mixed/random effects model incorporates inter-individual heterogeneity with random variables, i.e.:

$$V_{int} = \alpha_i + u_{in0} + \sum_{m=1}^{M_i} (\beta_{im} + u_{inm}) x_{inmt} \tag{9}$$

where $\alpha_i$ and $\beta_{im}$ are population-level parameters and $u_{inm}$ is a random variable (typically normally distributed) with 0 mean and standard deviation to be estimated from the data. A mixed/random intercept model is a model with only $u_{in0}$ being randomly distributed, and a mixed/random slopes model is a model with only the slopes, or coefficients, $u_{inm}$ being individual-specific.

## 3. Methodology

### 3.1. Functional effects models

In this study, we assume that the correlation of the error term across $t$ can be captured by individual-specific parameters learnt from the socio-demographic characteristics, such that the error term $\epsilon_{int}$ can be assumed to be *i.i.d.* and the methodology described in Section 2.1 is still applicable.
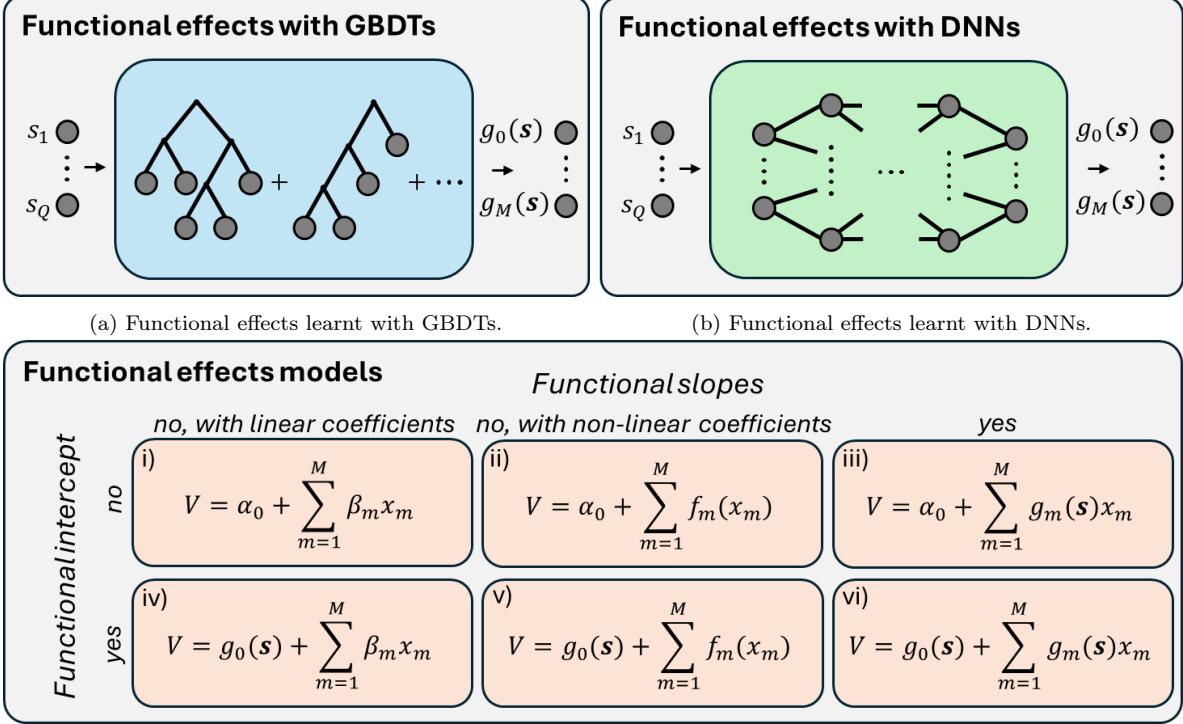
More formally, given a set of $M$ variables $\mathbf{x}_{int} \in \mathbb{R}^M$ and a set of $Q$ socio-demographic characteristics $\mathbf{s}_n \in \mathbb{R}^Q$ for individual $n$ and alternative $i$, we define the deterministic utility function as follows:

$$V_{int} = g_{i0}(\mathbf{s}_n) + \sum_{m=1}^{M_i} g_{im}(\mathbf{s}_n) x_{intm} \tag{10}$$

where:

- $g_{im}(\mathbf{s}_n)$ is the output of a gradient-based ML regressor using the socio-demographic characteristics as input data for all $M_i$ variables, $J$ classes, and individuals $n$ that does not depend on the choice dimension $t$.

- $g_{i0}(\mathbf{s}_n)$ is the output of an ML regressor using the socio-demographic characteristics as input data for the intercept for alternative $i$ and individual $n$.

We define more precisely the *Functional Intercept* (FI) model, where only $g_{i0}(\mathbf{s}_n)$ is the output of a gradient-based ML regressor, and the *Functional Slopes* (FS) model, where only $g_{im}(\mathbf{s}_n), \forall m, i$ are the outputs of a gradient-based ML regressor. Finally, the *Functional Intercept and Slopes* (FIS) model combines both functional intercept and slopes. If not learnt from the socio-demographic characteristics, the parameters can be either estimated linearly as in Eq. 2 or non-linearly imputed from a gradient-based ML regressor as in 3. Figure 1 provides an overview of all possible *functional effects* models. We note that keeping the same *linear-in-parameter* utility function as in a traditional MNL model allows for the functional effects model to maintain significant interpretability.

(a) Functional effects learnt with GBDTs.

(b) Functional effects learnt with DNNs.

(c) Types of functional effects models. They can be with or without functional intercept, slopes, and with linear or non-linear coefficients. Model i) is equivalent to an MNL.

Figure 1: Overview of the methodology.

### 3.2. ML Regressors

The functional intercept or slopes are the output of a gradient-based ML regressor. Note that this approach requires the models to be fit to the gradients of the loss function with respect to the individual-specific parameters, and so can be adapted for any gradient-based ML algorithm.

In this paper, we present a comparison between the two most popular ML regressors for tabular data: GBDTs and DNNs.

### 3.2.1. Functional effects with GBDTs

The first ML regressor used in this paper is GBDT (Friedman, 2001; Chen and Guestrin, 2016), which uses ensembles of regression trees to learn functional effects. Salvadé and Hillel (2025) extend this concept to a parametric utility context with RUMBoost, where, for a model with $R$ iterations, the GBDT predictive function is an additive function of the form:

$$f_{im}(x_{inm}) = \sum_{r=1}^{R} w_{inmr} \tag{11}$$

where $w_{inmr}$ is the leaf value (a constant) of a tree $r$, where the individual $n$ has been partitioned to. At each iteration, the leaf values of a new tree are chosen to directly minimise the second-order Taylor expansion of the loss function, such that:

$$w_{inmr} = -\frac{\left(\sum_{n \in l} g_{inm}\right)}{\left(\sum_{n \in l} h_{inm}\right)} \tag{12}$$

where:

- $g_{inm} = \partial \mathcal{L} / \partial V_{in}$ the first derivative of the loss function with respect to $V_{in}$;

- and $h_{inm} = \partial^2 \mathcal{L} / \partial^2 V_{in}$ is the second derivative of the loss function with respect to $V_{in}$; and

- $l$ is the subset of observations that has been partitioned to the corresponding leaf value.

In this paper, we extend this concept to functional effects being a function of socio-demographic characteristics $\mathbf{s}_n$. We have:

$$g_{im}(\mathbf{s}_n) = \sum_{r=1}^{R} w_{inmr} \tag{13}$$

where the leaf values are computed as in Eq. 12, but in the *parameter space*, that is $g_{inm} = \partial \mathcal{L} / \partial \beta_{inm} = \partial \mathcal{L} / \partial V_{in} \cdot \partial V_{in} / \partial \beta_{inm}$ the first derivative of the loss function with respect to $\beta_{inm}$ and $h_{inm} = \partial^2 \mathcal{L} / \partial^2 \beta_{inm} = \partial^2 \mathcal{L} / \partial^2 V_{in} \cdot (\partial V_{in} / \partial \beta_{inm})^2$ is the second derivative of the loss function with respect to $\beta_{inm}$. Since the utility function is linear-in-parameter, the first-order derivatives are scaled by $x_{inmt}$ and the second-order derivatives are scaled by $x_{inmt}^2$ compared to a RUMBoost model.

Note that, for the FI model, we combine RUMBoost to find generic non-linear coefficients (i.e., $f_{im}(x_{inm})$) with individual-specific intercept (i.e., $g_{i0}(\mathbf{s}_n)$) to capture preferences.

### 3.2.2. Functional effects with DNNs

The second ML regressor that we consider to learn functional effects is *feed-forward* DNNs (Goodfellow et al., 2016). More formally, the feed-forward DNN predictive function takes the following form:

$$g_{im}(\mathbf{s}_n) = h_O \circ \cdots \circ h_1(\mathbf{s}_n) \tag{14}$$

where:

- $h_o = a(w_o x + b_o) \quad \forall o = 1, \cdots, O$;

- $w_o$ and $b_o$ are the weight and biases of the hidden layer $o$;

- $a(x)$ is an activation function (e.g., ReLU, sigmoid, leaky ReLU, tanh, softplus, ...); and

- $O$ is the number of hidden layers in the DNN.

For training stability, it is common to split the datasets into batches of data. After each batch, the parameters of the models are updated with some variant of the gradient descent with first-order derivatives of the loss function (see e.g. Kingma and Ba (2014)), with

the gradients being efficiently computed through back-propagation (an algorithm efficiently applying the chain rule).

Note that the FI model falls back to the L-MNL proposed in Sifringer et al. (2020) and the FIS model falls back to the TasteNet-MNL proposed in Han et al. (2022). We refer the reader to the original papers for complete explanations of the underlying methodology.

### 3.2.3. Monotonicity constraints

For both models, we can constrain the sign of the functional effects with a Rectified Linear Unit (ReLU) activation function:

$$g_{im}^c(\mathbf{s}_n) = c \cdot \mathrm{ReLU}(cg_{im}(\mathbf{s}_n)) = c \cdot \max(0, cg_{im}(\mathbf{s}_n)) \tag{15}$$

where $c = 1$ for a positive monotonic constraint and $c = -1$ for a negative monotonic constraint.

### 3.3. Benchmarked models and relationship with prior work

Table 2 enumerates all 11 potential functional effects models, with the corresponding model numbers from Figure 1.

Table 2: All potential functional effects model. The model number is linked to the one from Figure 1. FI stands for functional intercept, FS stands for functional slopes, and FIS stands for functional intercept and slopes. FI-DNN is equivalent to the L-MNL model, and FIS-DNN is equivalent to the TasteNet-MNL model.

| Model | Intercept | Slopes | Model names | |
| | | | GBDT-based | DNN-based |
| --- | --- | --- | --- | --- |
| i) | $\alpha_0$ | $\beta_m x_m$ | (MNL/Ordinal Logit) | |
| ii) | $\alpha_0$ | $f_m(x_m)$ | RUMBoost | N/A |
| iii) | $g_0(\mathbf{s})$ | $\beta_m x_m$ | N/A | FI-DNN |
| iv) | $g_0(\mathbf{s})$ | $f_m(x_m)$ | FI-RUMBoost | N/A |
| v) | $\alpha_0$ | $g_m(\mathbf{s})x_m$ | FS-GBDT | FS-DNN |
| vi) | $g_0(\mathbf{s})$ | $g_m(\mathbf{s})x_m$ | FIS-GBDT | FIS-DNN |

Note that, whilst this paper presents a unified modelling framework for capturing functional effects in panel data, all models in Table 2 can also be applied to cross-sectional data. The proposed framework incorporates several existing models in the literature:

- the FI-DNN model is equivalent to the L-MNL model proposed by Sifringer et al. (2020),

- the FIS-DNN model is equivalent to the TasteNet-MNL model proposed by Han et al. (2022), and

- the FI-RUMBoost model was first proposed as an extension to the RUMBoost model by Salvadé and Hillel (2025).

10

The FS-GBDT, FIS-GBDT, and FS-DNN models are all new to this paper.

Note we use the names FI-DNN and FIS-DNN as: (i) they clearly indicate how each model relates to the others in the framework, and (ii) these models have been re-implemented within the Functional Effects framework[3], and may therefore be different from the original implementation. Note further that our naming convention distinguishes between FI-RUMBoost and FS-/FIS-GBDT, where the former uses the functional non linear coefficients $f_m(x_m)$ from Salvadé and Hillel (2025) whilst the latter use linear coefficients. Both FI-RUMBoost and FIS-GBDT use a single GBDT ensemble for the functional intercept.

We further highlight that there are three model combinations in the framework that are not evaluated in this paper. To the best of our knowledge, there is no existing methodology that enables the estimation of interpretable non-linear functional coefficients $f_m(x_m)$ with neural networks, hence we not present a DNN equivalent to the RUMBoost and FI-RUMBoost models. Similarly, due to the complexity of estimating generic parameters within gradient boosting, we do not evaluate the FI-GBDT model with linear coefficients.

### 3.4. Overview of experiments

Table 3 provides an overview of the datasets used in the experiments. For all experiments, we tune the hyperparameters with the Python library Optuna (Akiba et al., 2019), using the Tree-structured Parzen Estimator (TPE) algorithm with 100 trials. Table C.1 summarises the hyperparameters tuned and the search space. We keep the hyperparameters that have optimal values on the validation set. Note that for all functional effects models with GBDTs, the learning rate is fixed and calculated as $\min(0.1, 1/\min(\mathbf{M}))$ where $\min(\mathbf{M})$ is the smallest number of variables in a utility function.

Table 3: Overview of the datasets used in the experiments. SP means Stated Preference and RP means Revealed Preference

|                         | Synthetic   | Swissmetro  | LPMC        | easySHARE |
| ----------------------- | ----------- | ----------- | ----------- | --------- |
| Data type               | Synthetic   | SP          | RP          | RP        |
| Target type             | Multinomial | Multinomial | Multinomial | Ordinal   |
| $N_{\text{classes}}$    | 4           | 3           | 4           | 13        |
| $N_{\text{individuals}}$| 10000       | 1192        | 31954       | 130620    |
| $N_{\text{observations}}$| 100000     | 10692       | 81086       | 281975    |
| Avg. obs. per individual | 10         | 8.97        | 2.54        | 2.15      |

## 4. Synthetic experiment

The only true way of assessing the behavioural quality of the functional effects models is through a synthetic experiment. Indeed, in a control setting, we can generate known true functional effects and verify the ability of the functional effects models to recover them. We do so for FI-RUMBoost, FI-DNN and a Randnom Intercept model (i.e. a mixed logit model

---

[3]https://github.com/big-ucl/functional-effects-model

with distributed alternative specific constants) on a fully synthetic dataset of 100000 observations from 10000 individuals. We simulate a multinomial discrete choice problem with 4 alternatives. We generate 4 socio-demographic characteristics ($x_1$-$x_4$) and 4 alternative-specific variables ($x_5$-$x_8$), drawn from a continuous uniform distribution between 0 and 1 such that $x_k \sim \mathcal{U}_{[0,1]}, \forall k = 1, \cdots, 8$. We draw 10000 unique individuals as independent draws of $x_1$-$x_4$. For each individual, we then draw 10 different choice scenarios, as separate independent draws of $x_5$-$x_8$, repeating the socio-demographic characteristics for each scenario. The complete model specification is the following:

$$V_1 = e^{x_1+x_2+x_3+x_4} - 1 \cdot x_5 \tag{16}$$
$$V_2 = (x_1 + x_2 + x_3 + x_4)^2 - 1 \cdot x_6 \tag{17}$$
$$V_3 = -\ln(x_1 x_2 x_3 x_4) - 1 \cdot x_7 \tag{18}$$
$$V_4 = -1 \cdot x_8 \tag{19}$$

We assume a Type-1 Extreme Value i.i.d. error term, and so obtain choice probabilities from the logit/softmax function. Discrete choices are then sampled from the probabilities using a Monte-Carlo simulation. The functional effects are normalised to 0 in the third utility function, as the functional effects would not be recoverable because of the overspecification of the logit/softmax function. We repeat the process to obtain a test set of 20000 observations from 2000 previously unobserved individuals. Table C.2 of Appendix C summarises the optimal hyperparameter values from the hyperparameter search. We implement the Random Intercept model with Biogeme (Bierlaire, 2023), making the assumption that the random intercepts are normally distributed, and using the mean (i.e. without Monte-Carlo Simulation) for forecasting on the test set. We follow Krueger et al. (2021) to simulate the maximum likelihood estimation with 500 draws generated from the Modified Latin Hypercube sampling approach (Hess et al., 2006).

Table 4 shows the Mean Absolute Error (MAE) between recovered and true functional intercepts, the negative Cross-Entropy Loss (CEL) on both train and test set and the computational time to train the models. We observe that FI-DNN marginally outperforms FI-RUMBoost on the MAE and CEL, while FI-RUMBoost is about 4 times faster. The good performance of the two models with functional effects contrasts with the MAE of the Random Intercept model, which is bound to a normal distribution on the train set, and can only use the mean of this distribution on the test set.

Table 4: MAE between recovered and true functional intercepts, negative cross-entropy loss on train and test set, and computational time. The MAE is averaged over the three class intercepts.

| | MAE | | CEL | | |
| | Train | Test | Train | Test | Comput. time [s] |
|---|---|---|---|---|---|
| **FI-RUMBoost** | 0.044 | 0.043 | 1.343 | 1.344 | 3.100 |
| **FI-DNN** | 0.038 | 0.037 | 1.346 | 1.343 | 11.820 |
| **Random Intercept** | 0.289 | 0.112 | 1.35 | 1.349 | 47.640 |

Figure 2 shows the recovered functional effects and parameters on the train set. The distributions of the functional effects are overall well-recovered by FI-RUMBoost and FI-

DNN. We note that the distributions of the functional effects learnt with DNN are more concentrated around the mean for alternative 1 and 2. However, the functional effects learnt with GBDT seem to match more closely the true distribution. The Random Intercept model, on the other hand, fails to recover the true distribution, because of the *a priori* assumption of normality.

Figure 3 displays the recovered functional intercepts on the holdout test set. FI-RUMBoost and FI-DNN are able to recover the functional effects even on unseen data. These results indicate that the functional effects models are suitable for inference, and contrast with the limitations of the Random Intercept model, which can only output the mean of the random effects estimated during training.

Finally, we also observe in Figure 4 that all models are able to recover the true linear parameters for all variables. This special linear case highlights the proficiency of *linear-in-parameters* models to recover linear functions, but it is reassuring to notice that FI-RUMBoost, developed to discover non-linear coefficients, is still able to recover the linear functions.



(a) Intercept for alternative 1 - train set

(b) Intercept for alternative 2 - train set

(c) Functional intercept for alternative 3 - train set

Figure 2: Distribution of intercepts for the ground truth, FI-RUMBoost, FI-DNN, and Random Intercept model on synthetic train dataset. Note the plot for the Random Intercept model is truncated to [0,1] for comparison.

(a) Intercept for alternative 1 - test set

(b) Intercept for alternative 2 - test set

(c) Intercept for alternative 3 - test set

Figure 3: Distribution of intercepts for the ground truth, FI-RUMBoost, FI-DNN, and Random Intercept model on synthetic test dataset. Note the Random Intercept uses the mean intercept values for forecasting and is truncated for comparison.
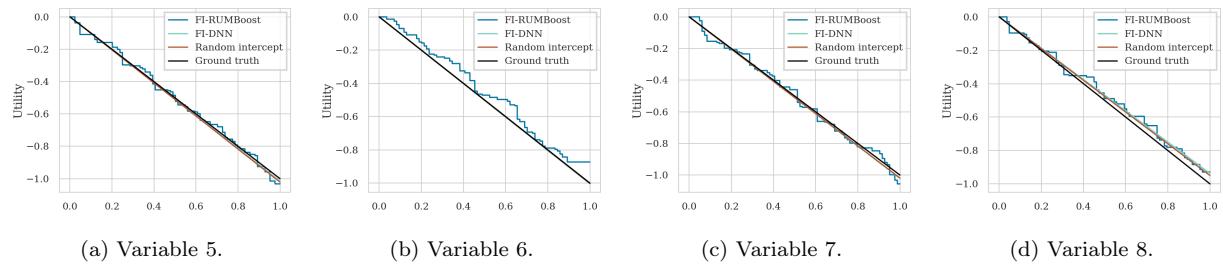


(a) Variable 5.    (b) Variable 6.    (c) Variable 7.    (d) Variable 8.

Figure 4: Coefficients for the ground-truth, FI-RUMBoost, FI-DNN, and Random Intercept model. Note FI-RUMBoost uses non-linear coefficients $f_m(x_m)$, where the other models use linear coefficients $\beta_m x_m$.

14

## 5. Real-world case studies

### 5.1. Mode choice datasets

#### 5.1.1. Datasets and model specifications

We use the open-source Swissmetro (Bierlaire et al., 2001) and London Passenger Mode Choice (LPMC) (Hillel et al., 2018) datasets for the benchmarks. The first dataset is a stated preference datasets where respondents are asked to choose between Swissmetro (a new and hypothetical underground mag-lev transportation mode in Switzerland), car, or train. We follow the same data preprocessing and model specification as in Han et al. (2022). We keep observations where the choice is known. After this preprocessing, we have 10692 observations from 1192 individuals (about 9 observations per individual) that we split into 70 % for training, 15 % for validation, and 15 % for testing.

The second dataset is a revealed preference dataset obtained from the London Travel Demand Survey (LTDS) travel diary dataset, enriched with alternative-specific travel times obtained from the Google Directions API and a bespoke cost model. Possible alternatives are walking, cycling, public transport, and driving. We follow the same data preprocessing and model specification as in Salvadé and Hillel (2025). The dataset contains 81086 trips made by 31954 individuals (about 2.5 trips per individual). The first two years of observation are used as a training set that we split at the household level into 80% for training and 20% for validation.

Table 5 summarises which variables are included in the models for the Swissmetro and LPMC datasets, as well as monotonicity constraints. Note that we assume that all alternatives are available for every individual. Table C.3 and C.4 in Appendix C summarise optimal hyperparameter values obtained from the hyperparameter search for all models.

#### 5.1.2. Predictive performance

We compare the different models with their negative cross-entropy loss on a holdout test set. We include in the benchmarks two out-of-the-box ML classifiers where the utility is computed as in Eq. 4: GBDT, using the LightGBM python library (Ke et al., 2017); and DNN, using the PyTorch python library (Ansel et al., 2024). These models provide state-of-the-art predictive performance without accounting for inter-heterogeneity, therefore being useful to measure the impact of accounting for preference heterogeneity. The results are shown in Table 6. Overall, all functional effects models learnt with GBDTs exhibit better predictive performance than the functional effects models learnt with DNNs on the Swissmetro dataset. FIS-GBDT performs the best, while the interpretable baselines perform the least well. We note that GBDT, a blackbox baseline, is the second best performing model, but the model has full trip feature interaction, as opposed to the functional effects models and interpretable baselines, which have no trip feature interaction. Looking at the computational time, FS-DNN and FIS-DNN are faster than the FS-GBDT and FIS-GBDT, whereas FI-RUMBoost and RUMBoost are slightly faster than FI-DNN and MNL. On the LPMC dataset, FI-RUMBoost performs best. On this dataset, FS-DNN and FIS-DNN perform better than their GBDT counterpart, whereas the baseline models perform the least well. We deduce from these results that non-linear coefficients and inter-individual heterogeneity are of greater importance in this dataset. Finally, the functional effects models

Table 5: Variables used in Swissmetro and LPMC datasets. A negative monotonic constraint indicates that an increase in the variable must decrease the utility function.

| Variable | Swissmetro | LPMC | Monotonic constr. |
|---|---|---|---|
| *Alternatives* | Swissmetro, | Walking, | – |
| | Train, | Cycling, | – |
| | Driving | PT, | – |
| | | Driving | – |
| | | | |
| *Socio-demographics (inputs for functional effects)* | | | |
| Age | Yes | Yes | – |
| Income | Yes | – | – |
| Gender | Yes | Yes | – |
| Purpose of trip | Yes | Yes | – |
| Has luggage | Yes | – | – |
| Pays for trip | Yes | – | – |
| Swiss seasonal ticket | Yes | – | – |
| First class | Yes | – | – |
| Driving license | – | Yes | – |
| N. of cars in household | – | Yes | – |
| Fuel type | – | Yes | – |
| | | | |
| *Trip variables (alternative-specific variables)* | | | |
| Travel times | Yes | Yes | Negative |
| Costs | Yes | PT and Driving | Negative |
| Headway | Train and Swissmetro | – | Negative |
| Seat type | Swissmetro only | – | – |
| Trip day | – | Yes | – |
| Start time | – | Yes | – |
| Distance | – | Yes | Negative |
| Degree of congestion | – | Driving only | Negative |

learnt with GBDTs are faster than the ones learnt with DNNs, and all models are faster proportionally to the number of functional effects in the model.

Table 6: Benchmarks on the holdout test set. The models are evaluated on the negative cross-entropy loss (lower the better). We also report the training time with optimal hyperparameters in seconds. Best results are highlighted in bold. FE means functional effects. Baseline models are trained without socio-demographic characteristics.

|  | Model | Swissmetro | | LPMC | |
| --- | --- | --- | --- | --- | --- |
|  |  | CEL | Time [s] | CEL | Time [s] |
| **GBDT-based FE** | FIS-GBDT | **0.614** | 9.02 | 0.699 | 15.40 |
|  | FS-GBDT | 0.679 | 7.32 | 0.724 | 12.87 |
|  | FI-RUMBoost | 0.630 | 1.76 | **0.673** | 3.40 |
| **DNN-based FE** | FIS-DNN | 0.670 | 6.04 | 0.691 | 69.19 |
|  | FS-DNN | 0.720 | 3.04 | 0.714 | 66.27 |
|  | FI-DNN | 0.779 | 12.18 | 0.705 | 52.21 |
| **Baselines - Interpretable** | RUMBoost* | 0.786 | 1.54 | 0.824 | 3.36 |
|  | MNL* | 0.854 | 9.74 | 0.841 | 29.00 |
| **Baselines - Blackbox** | GBDT* | 0.622 | 18.92 | 0.805 | 15.30 |
|  | DNN* | 0.746 | 1.76 | 0.840 | 29.51 |

*Baseline models are trained without socio-demographic characteristics.

### 5.1.3. Model parameters

Due to the large number of parameters and model combinations across multiple alternatives for the Swissmetro and LPMC, we do not present a structured overview of the model parameters for each model in this paper. We direct the reader to Salvadé and Hillel (2025) for an in-depth discussion of the FI-RUMBoost parameters on the LPMC data. For the remaining models and DNN-based models, all results are available on GitHub[4].

We summarise key findings for the model coefficients and functional effects as follows:

- The models with non-linear coefficients show the biggest differences with models with linear coefficients on continuous variables such as travel time or trip starting time;

- The functional effects are more scattered on the LPMC dataset compared to the Swissmetro dataset;

- The functional effects learnt with GBDT are more likely to experience extreme negative values; and

- The monotonic constraints push some functional slopes towards zero for most individuals.

---

[4]https://github.com/big-ucl/functional-effects-model

## 5.2. EasySHARE: modelling the mental health of elder people in Europe

### 5.2.1. Dataset

For this case study, we use data from the Survey of Health, Ageing and Retirement in Europe (SHARE), a longitudinal study of older people's health across 28 European countries. We use the simplified easySHARE dataset (SHARE-ERIC, 2024), which has been preprocessed to combine the data from the 9 waves of the study in a long table format. For this study, we follow Mendorf et al. (2023) where we model the EURO-D measure of depressive symptoms, a scale composed of 12 depressive symptoms. A measurement of 0 indicates that the patient has no depressive symptoms and a score of 12 means that the patient exhibits all 12 depressive symptoms. This is an ordinal target variable, meaning that we need to adapt the generic multiclass methodology described in 2.1. This is done using the CORAL methodology (Shi et al., 2023), with complete derivations in Appendix A.

We preprocess the dataset to remove missing values for the target variable and drop variables with more than 10% missing values. We further remove observations that would still have missing values. We drop wave 3 and wave 7 observations as the survey differs from other waves. We encode all categorical variables with dummy variables, where we normalise one category to 0. After preprocessing, we have 281975 observations from 130620 individuals (about 2.15 observations per individual). We split the data at the individual level to avoid data leakage, keeping 20% for the hold-out test set and 80% for training. We further split (also at the individual level) the training set into 80% for training and 20% for validation, to perform a hyperparameter search for both functional effects models with GBDTs and DNNs. Table C.5 in Appendix C summarises the optimal hyperparameter values from the hyperparameter search. Table 7 summarises and provides a brief description of all variables used in the model.

### 5.2.2. Predictive performance

We report the performance of the 8 different models described in 3.3 with respect to the MAE, EMAE, and MCEL on the holdout test set (see Appendix A for complete derivations of the metrics). Table 8 shows all these results.

On average, all models predict the number of depressive symptoms between 1.368 to 1.421 away from the observed measurement. We observe that all models accounting for preferences perform better than models without any inter-individual heterogeneity. FI-RUMBoost has the best MAE, whereas FI-RUMBoost, FS-GBDT, FIS-GBDT, and FS-DNN exhibit the best MCEL and EMAE. We remark that the metric difference between models with functional effects is minimal. Since we perform only a single train–validate–test split for computational reasons, it is difficult to conclude that one model is better than another. In terms of computational time, learning functional effects with DNNs is about 3 to 16 times faster than with GBDTs. Finally, we also compare the estimation of the ordinal thresholds for all models in Appendix D. It is reassuring to observe that all values are in the same range, with minor differences.

### 5.2.3. Model parameters

The primary advantage of using ML regressors in a constrained setting, such as functional effects models, compared to out-of-the-box ML classification models, is that they are

Table 7: easySHARE dataset variable descriptions and types

| Variable | Description | Type |
|---|---|---|
| *Target* | | |
| eurod | Depression scale (0–12) | Ordinal |
| *Socio-demographics coefficients (inputs for functional effects):* | | |
| age | Age in years at interview | Continuous |
| female | Gender (1 - female, 0 - male) | Binary |
| country | Country of residence | Nominal |
| mar_stat | Marital status | Nominal |
| dn004_mod | Respondent born in interview country (yes/no) | Binary |
| isced1997_r | Education level (1-6) | Ordinal |
| thinc_m | Household net income | Continuous |
| ch001_ | Number of children | Discrete |
| hhsize | Household size | Discrete |
| partnerinhh | Partner lives in the same household (yes/no) | Binary |
| mother_alive | Mother alive (yes/no) | Binary |
| father_alive | Father alive (yes/no) | Binary |
| sp002_mod | Receives help from outside household (yes/no) | Binary |
| smoking | Respondent smokes (yes/no) | Binary |
| ever_smoked | Respondent ever smoked (yes/no) | Binary |
| br015_ | Frequency of vigorous activities | Ordinal |
| ep005_ | Job situation | Nominal |
| co007_ | Household meets end | Ordinal |
| has_citizenship | Respondent has citizenship (yes/no) | Binary |
| *Situational variables:* | | |
| *Health indices & conditions* | | |
| bmi | Body mass index | Continuous |
| sphus | Self-perceived health (1 - excellent to 5 - poor) | Ordinal |
| chronic_mod | Number of chronic diseases | Discrete |
| maxgrip | Maximum grip strength (kg) | Continuous |
| *Functional & mobility scores* | | |
| adla | Sum of difficulty in daily tasks (0–5) | Ordinal |
| iadlza | Instrumental activities difficulties (0–5) | Ordinal |
| mobilityind | Mobility difficulties (0–4) | Ordinal |
| lgmuscle | Large muscle functioning difficulties (0–4) | Ordinal |
| grossmotor | Gross motor skills difficulties (0–4) | Ordinal |
| finemotor | Fine motor skills difficulties (0–3) | Ordinal |
| *Cognitive measures* | | |
| recall_1 | Immediate word recall score (0–10) | Discrete |
| recall_2 | Delayed word recall score (0–10) | Discrete |
| orienti | Orientation score (0–4) | Discrete |
| numeracy_1 | Numeracy test score | Discrete |
| *Healthcare usage* | | |
| hc002_ | Number of doctor visits last 12 months | Discrete |
| hc012_ | Hospital stay in last 12 months (yes/no) | Binary |
| hc029_ | Nursing home in last 12 months (perm./temp./no) | Ordinal |

Table 8: Benchmarks on the holdout test set. The models are evaluated on the mean absolute error, expected mean absolute error, and multi-label cross entropy loss. All metrics are lower the better. We also report the training time with optimal hyperparameters in seconds. Best results are highlighted in bold. FE means functional effects. Baseline models are trained without socio-demographic characteristics.
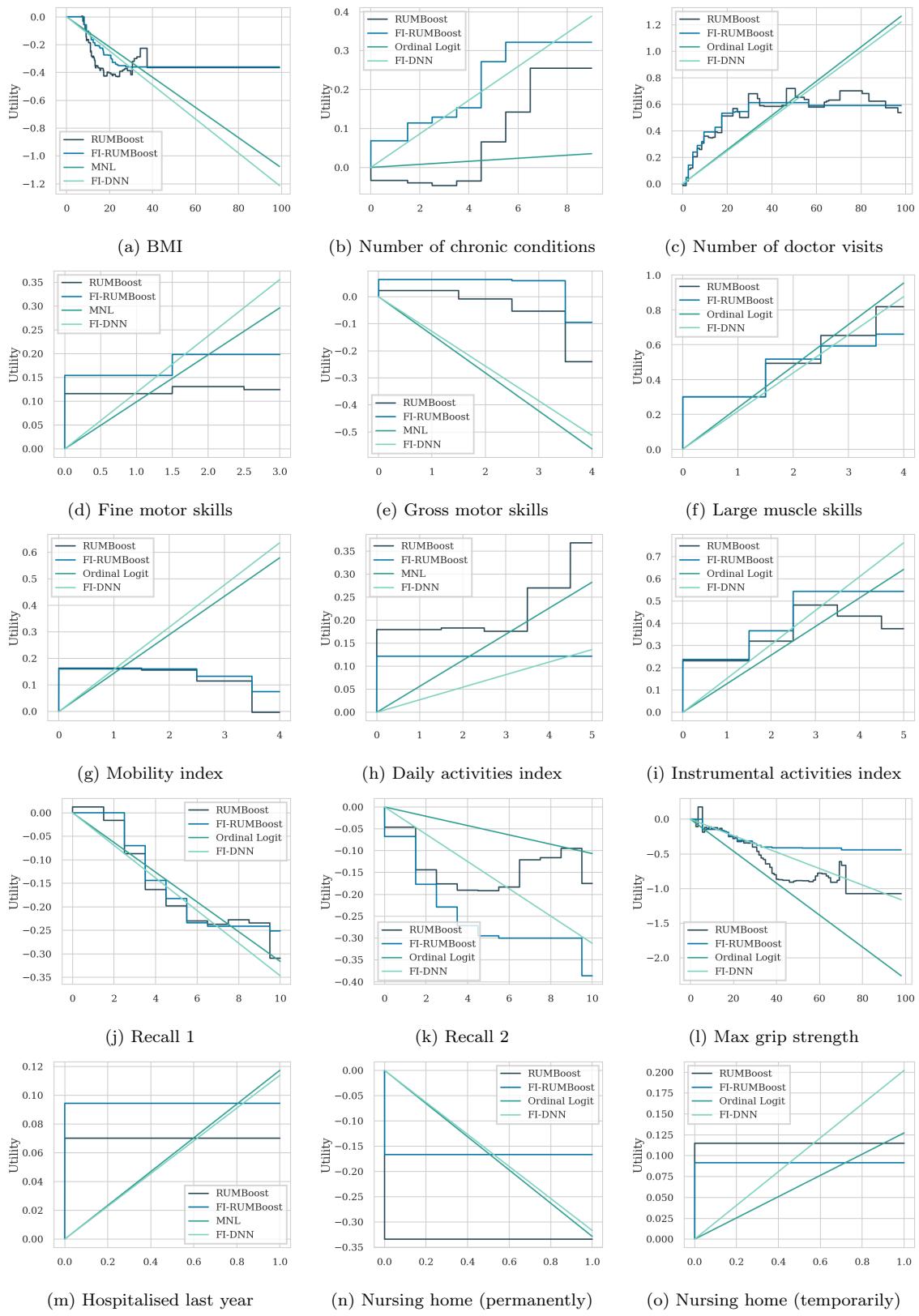
| | Model | MAE | EMAE | MCEL | Time [s] |
|---|---|---|---|---|---|
| **GBDT-based FE** | FIS-GBDT | 1.369 | **0.146** | **0.251** | 169 |
| | FS-GBDT | 1.37 | **0.146** | **0.251** | 178 |
| | FI-RUMBoost | **1.368** | **0.146** | **0.251** | 241 |
| **DNN-based FE** | FIS-DNN | 1.373 | 0.148 | 0.252 | 61 |
| | FS-DNN | 1.371 | **0.146** | **0.251** | 65 |
| | FI-DNN | 1.38 | 0.148 | 0.253 | 44 |
| **Baselines** | RUMBoost* | 1.414 | 0.151 | 0.26 | 807 |
| | Ordinal Logit* | 1.421 | 0.152 | 0.261 | 50 |

*Baseline models are trained without socio-demographic characteristics.

interpretable. When trained without functional slopes, we can observe the traditional parameters as in an Ordinal Logit from a functional effects model with linear coefficients and the non-linear utility function from functional effects models with non-linear coefficients. We can also observe the distribution of the individual-specific functional intercept and functional slope values as histograms.

Figure 5 shows the linear and non-linear effects from the four models without functional slopes. Overall, we observe that the linear and non-linear utility output from the functional effects models exhibit the same trends. An increase in the number of conditions, number of doctor visits, fine motor difficulties, large muscle difficulties, mobility difficulties, daily activity difficulties, instrumental activity difficulties, if the respondent has been hospitalised or temporarily in a nursing home, and if the self-perceived health is not excellent (reference category) increases the likelihood of having more depressive symptoms. On the other hand, a higher BMI, better gross motor difficulties, being able to recall more words, having a better max grip strength, and living permanently in a nursing home decrease the likelihood of having more depressive symptoms. We also observe that the non-linearity provides benefits for: the number of doctor visits, which is logarithmic; the BMI, which exhibits a plateau; and the mobility difficulties, which is a parabola.

We can also visualise histograms of the functional intercept and slopes of the six models having functional effects. We show the functional intercept in Figure 6, and we select the functional slopes of the max grip strength variable to analyse in Figure 7. Other figures are in Appendix B. We observe that all functional intercepts learnt with GBDT are unimodal, mostly normally distributed, whereas the functional intercepts learnt with DNN seem bimodal. It is interesting to see how the functional intercept learnt with DNN has a slightly bigger standard deviation than the one learnt with GBDT when the model is also trained with functional slopes. We also observe that all distributions have a positive mean. For the max grip strength functional slopes distribution, we mostly observe an extreme value distribution with a negative mean for all models. For most individuals, an increase in max grip strength is negatively correlated with a higher number of depressive symptoms. It is similar
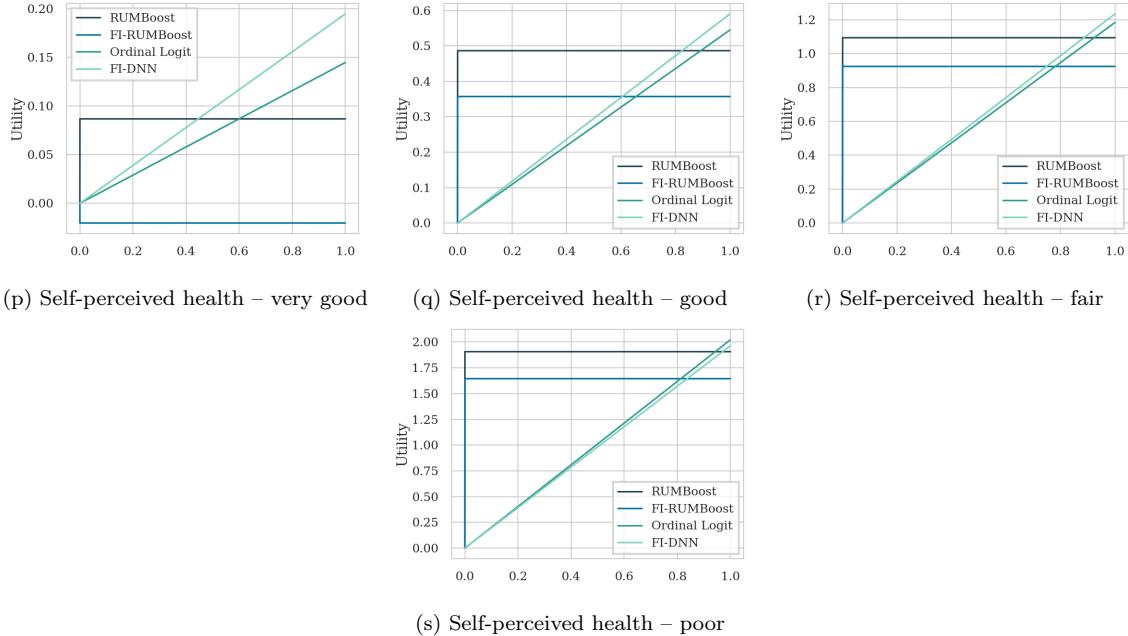
(a) BMI

(b) Number of chronic conditions

(c) Number of doctor visits

(d) Fine motor skills

(e) Gross motor skills

(f) Large muscle skills

(g) Mobility index

(h) Daily activities index

(i) Instrumental activities index

(j) Recall 1

(k) Recall 2

(l) Max grip strength

(m) Hospitalised last year

(n) Nursing home (permanently)

(o) Nursing home (temporarily)

Figure 5: Non-linear and linear coefficients of health-related variables on the EURO D depression scale measurement.

to what was observed for the models without functional slopes, but with more nuances, since we can observe that some individuals have a positive slope, meaning that an increase in the max grip strength is positively correlated with more depressive symptoms.

## 6. Conclusion

In this paper, we have adapted and applied existing methodologies to panel data, effectively learning functional effects from socio-demographic characteristics. This allows for modelling inter-heterogeneity in an interpretable ML framework, outperforming out-of-the-box and interpretable ML choice models that would not account for preferences. In addition, we have extensively benchmarked 8 out of the 11 possible functional effects models, showing how they can account for preferences of unobserved individuals, a limitation of traditional choice models such as the fixed and mixed effects models. By doing so, we make the implicit assumption that the choices and tastes of individuals are inherently linked with their socio-demographic characteristics. We note that it is important to review the socio-demographic characteristics included in the model to stay on ethical grounds and be sure that no specific strata of the population are being discriminated against. Interestingly, our approach captures the relationship of individuals at a given point in time. This means that socio-demographic characteristics can change, also adapting the functional effects that these individuals are experiencing. This is likely to be the case for real-life situations, where, for example, having a higher monthly income can change the life experience and, therefore, unobserved factors influencing the decision process. In addition, we have applied the methodology to an ordinal choice problem and multiclass classification problems, but the methodology is not problem-specific and can easily be applied to regression tasks and more complex choice models. Further work includes applying eXplainable Artificial Intelligence (XAI) techniques to

(a) Functional intercept

(b) Functional intercept - models with functional slopes

Figure 6: Functional intercepts learnt with GBDT and DNN for FI-RUMBoost and FI-DNN (a) and FIS-GBDT and FIS-DNN (b).



(a) Max grip strength

(b) Max grip strength - models with functional intercept

Figure 7: Functional slopes learnt with GBDT and DNN for the max grip strength variable for FS-GBDT and FS-DNN (a) and FIS-GBDT and FIS-DNN (b).

23

the functional effects distributions to map heterogeneity to socio-demographics, e.g., certain demographic groups having preferences to alternatives or being more time sensitive, and extending the parametric ML approach to dynamic effects for panel data.

## CRediT authorship contribution statement

**Nicolas Salvadé**: Writing – original draft, Visualization, Software, Methodology, Conceptualization, Formal analysis, Investigation. **Tim Hillel**: Writing – review & editing, Supervision, Methodology, Conceptualization, Project administration.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

## Appendix A. Ordinal data - CORAL

The CORAL methodology, introduced in Shi et al. (2023), has been developed to model ordinal target variables. It is a popular methodology in the ML community and exhibits close similarities with the Ordinal Logit model. CORAL can be derived from a latent regression model, where we have:

$$U_n = \hat{y}_n = V_n + \epsilon_n \tag{A.1}$$

where:

- $\hat{y}_n$ is the unobserved latent response for an individual $n$ interpreted as the utility function $U_n$. Note that this is a simple regression value, being the same for all alternatives;

- $V_n$ is the deterministic utility defined in Equation 2; and

- $\epsilon_n$ is the error term, capturing unobserved information.

The continuous space of the deterministic utility can be separated into $J$ categories with $J-1$ thresholds $\tau_j$ such that the predicted dependent variable $y'_n$ is:

$$y'_n = \begin{cases} 1, & \text{if } \hat{y}_n \leq \tau_1, \\ 2, & \text{if } \tau_1 < \hat{y}_n \leq \tau_2, \\ \vdots \\ J, & \text{if } \hat{y}_n > \tau_{J-1} \end{cases} \tag{A.2}$$

If we assume that the error term follows a standard logistic distribution, the probability that the latent scalar output of the model is greater than a threshold $\tau_j$ is the sigmoid function:

$$P_{jn} = P(\hat{y}_n > \tau_j) = \sigma(V_n - \tau_j) = \frac{1}{1 + \exp^{-(V_n - \tau_j)}} \quad \forall j = 1, \ldots, J-1 \tag{A.3}$$

Then, the probability of choosing a class can be reconstructed from the binary decomposition, i.e.:

$$P(\hat{y}_n = 1) = 1 - P(\hat{y}_n > \tau_1) \tag{A.4}$$

$$P(\hat{y}_n = j) = P(\hat{y}_n > \tau_{j-1}) - P(\hat{y}_n > \tau_j) \quad \forall j = 2, \ldots, J-1 \tag{A.5}$$

$$P(\hat{y}_n = J) = P(\hat{y}_n > \tau_{J-1}) \tag{A.6}$$

In CORAL, and this is where the methodology differs from a traditional Ordinal Logit model, the thresholds and parameters are chosen to maximise the Multi-label Cross-Entropy Loss (MCEL) function of the binary sub-problems (as opposed to the negative CEL function of the probabilities in the Ordinal Logit model):

$$\mathcal{L} = \sum_{n=1}^{N} \sum_{j=1}^{J-1} \mathbb{1}(j > y_n) \ln(P(\hat{y}_n > \tau_j)) + (1 - \mathbb{1}(j > y_n)) \ln(1 - P(\hat{y}_n > \tau_j)) \tag{A.7}$$

where:

$$\mathbb{1}(j > y_n) = \begin{cases} 1, & \text{if } j > y_n \text{ the chosen class,} \\ 0, & \text{otherwise} \end{cases} \tag{A.8}$$

This equation has the advantage of including all thresholds in the loss function, therefore also penalising thresholds that are not directly precedent or subsequent to the chosen class. For model evaluation, in addition to the MCEL, we also report two additional metrics:

1. the *mean absolute error* (MAE) a discrete metric; and

2. and the *expected mean absolute error* (EMAE), a probabilistic metric which considers distances between classes.

The MAE is defined as:

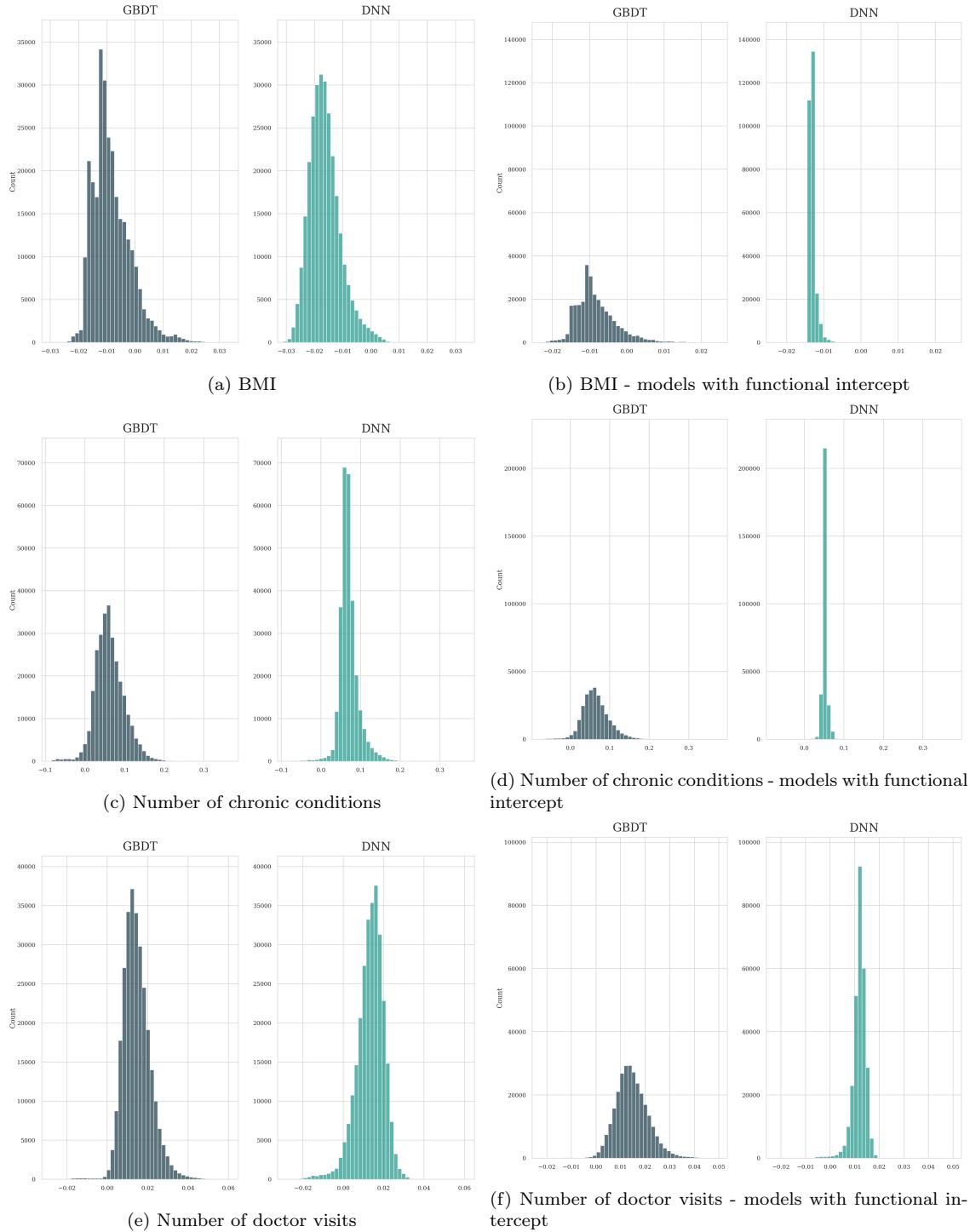$$MAE = \frac{1}{N} \sum_{n=1}^{N} |y_n - l_n| \tag{A.9}$$

and the EMAE as:

$$EMAE = \frac{1}{N} \sum_{n=1}^{N} \sum_{j=1}^{J} P(\hat{y}_n = j) \cdot (j - y_n)^2 \tag{A.10}$$
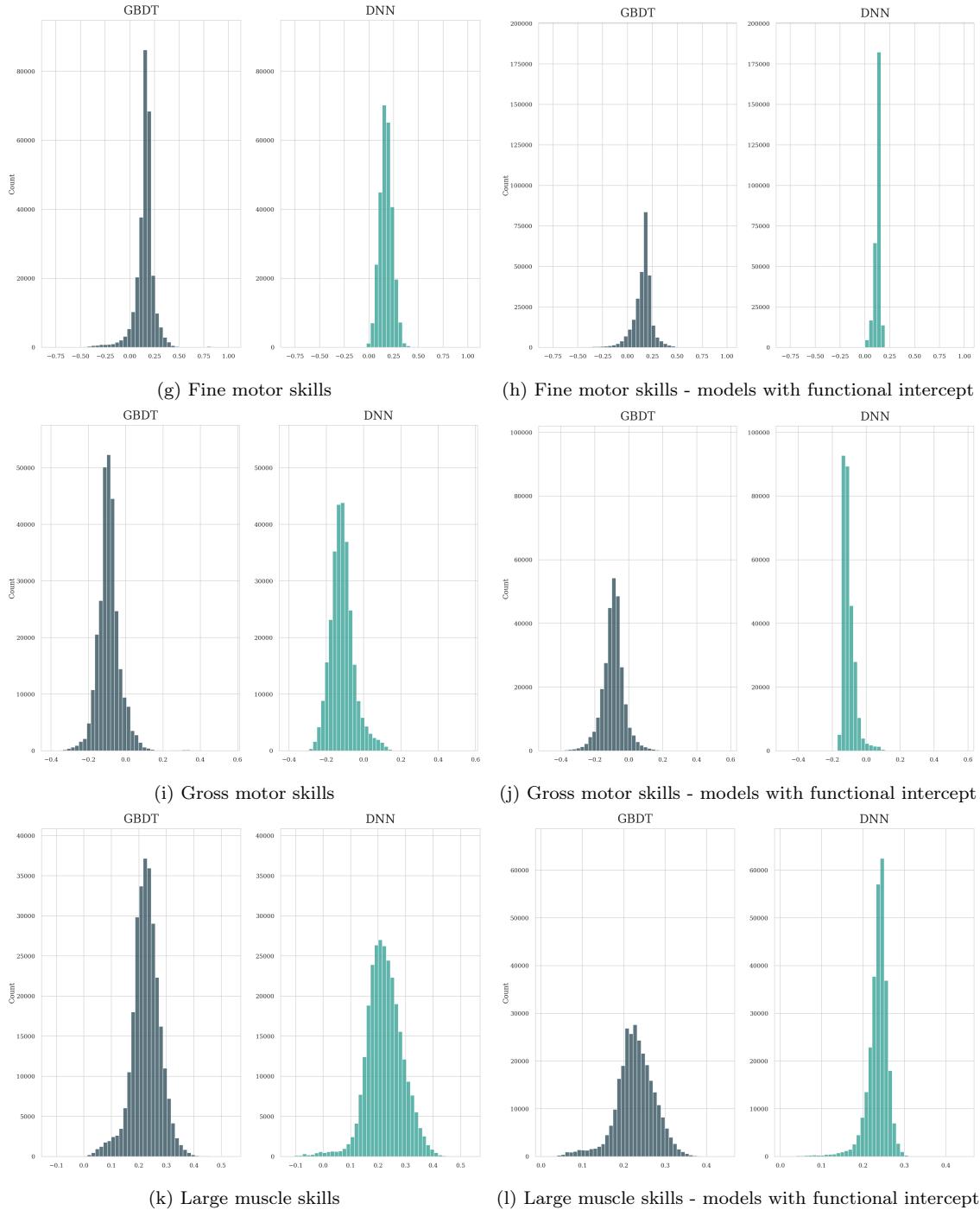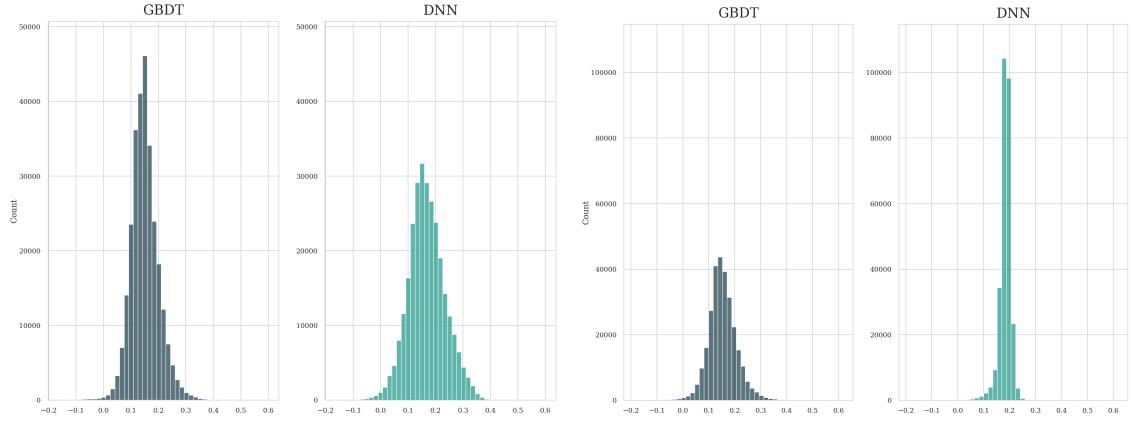
where:

- $N$ is the number of observations;

- $y_n$ is the observed choice for individual $n$;

- $P(\hat{y}_n = j)$ is the predicted probability that individual $n$ will choose alternative $j$; and

- $l_n = \sum_{j=1}^{J-1} \mathbb{1}(\hat{y}_n > \tau_j)$ is the discrete class prediction of the model for individual $n$.
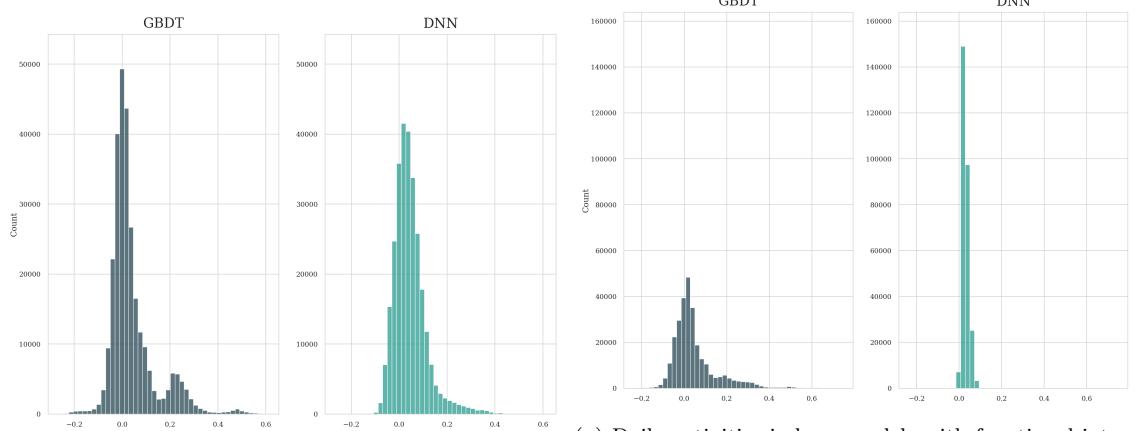
## Appendix B. Functional effects

Figure B.1 shows all functional slopes for the models of Section 5.2 with the easySHARE dataset.
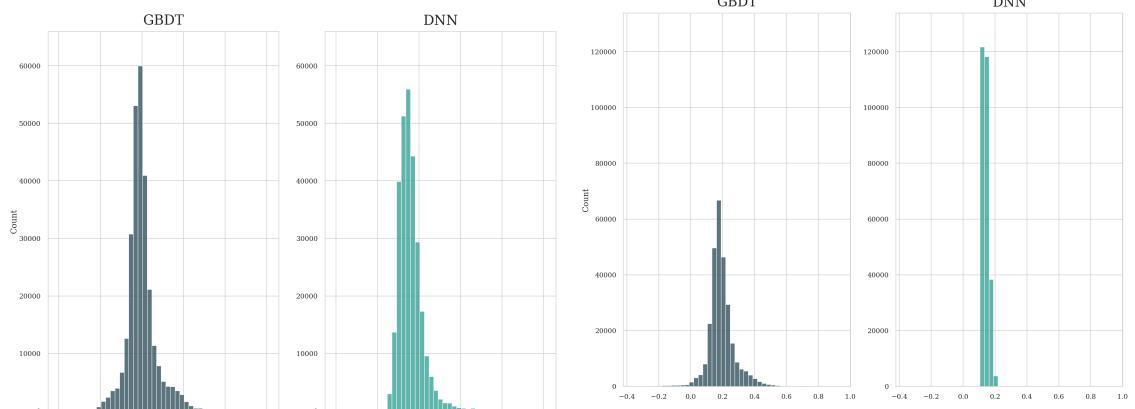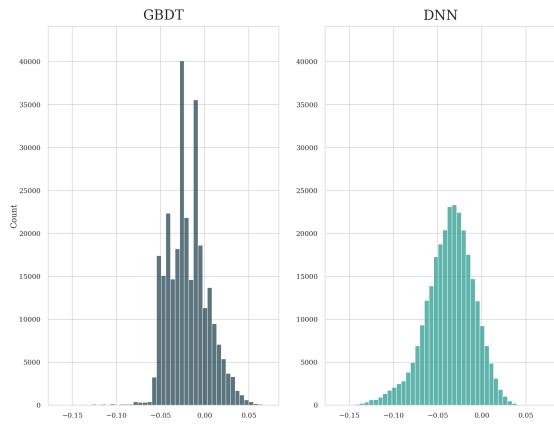
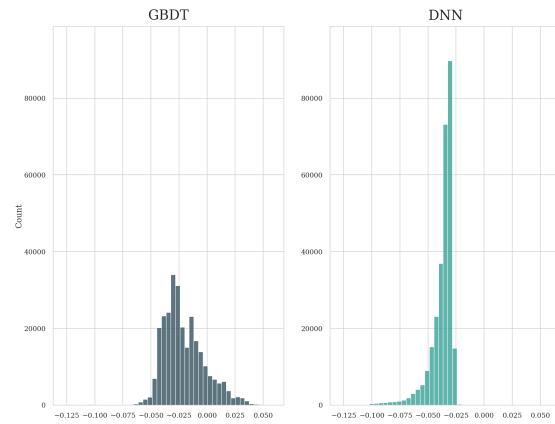(a) BMI

(b) BMI - models with functional intercept

(c) Number of chronic conditions

(d) Number of chronic conditions - models with functional intercept

(e) Number of doctor visits

(f) Number of doctor visits - models with functional intercept

(g) Fine motor skills

(h) Fine motor skills - models with functional intercept



(i) Gross motor skills

(j) Gross motor skills - models with functional intercept



(k) Large muscle skills

(l) Large muscle skills - models with functional intercept

(m) Mobility index

(n) Mobility index - models with functional intercept



(o) Daily activities index

(p) Daily activities index - models with functional intercept



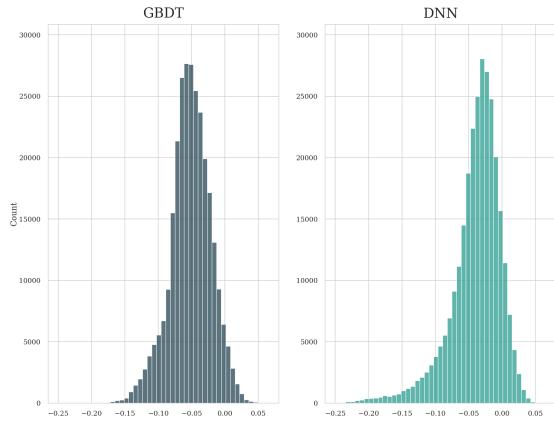(q) Instrumental activities index

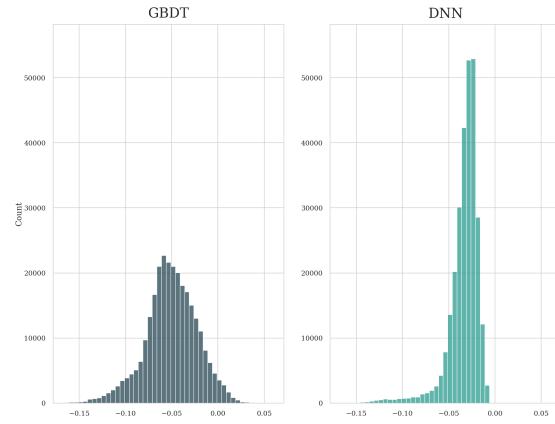(r) Instrumental activities index - models with functional intercept
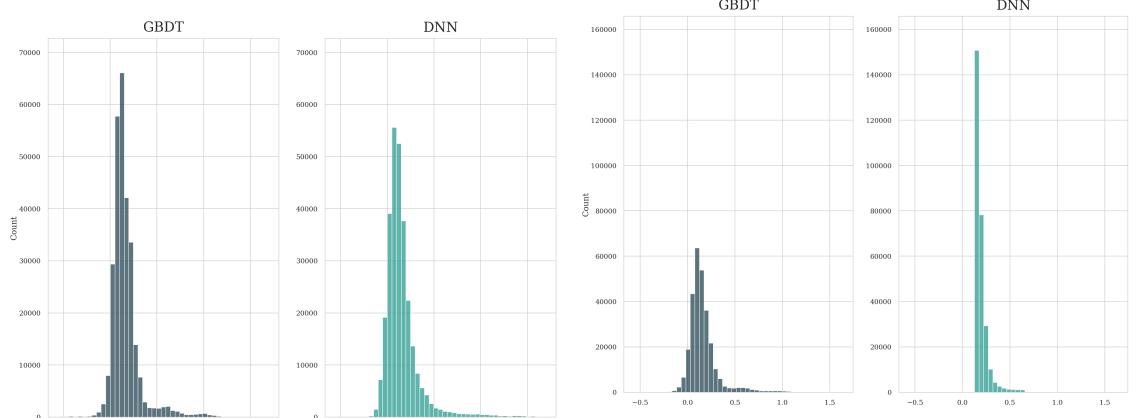
(s) Recall 1

(t) Recall 1 - models with functional intercept



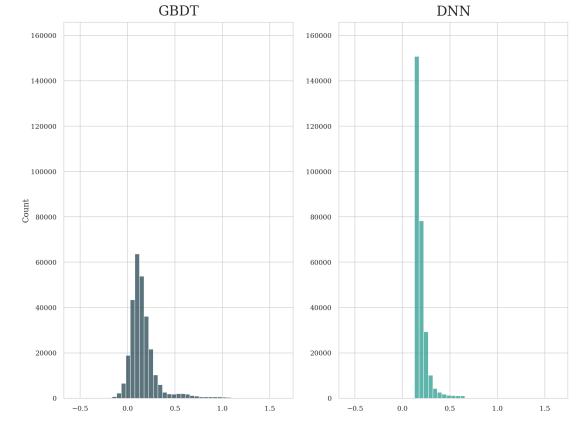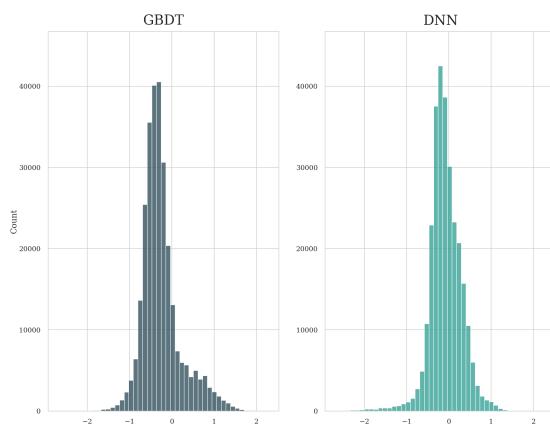(u) Recall 2

(v) Recall 2 - models with functional intercept

(w) Hospitalised last year



(x) Hospitalised last year - models with functional intercept



(y) Nursing home last year (permanently)



(z) Nursing home last year (permanently) - models with functional intercept



(aa) Nursing home last year (temporarily)



(ab) Nursing home last year (temporarily) - models with functional intercept

(ac) Self-perceived health - very good



(ad) Self-perceived health - very good - models with functional intercept



(ae) Self-perceived health - good



(af) Self-perceived health - good - models with functional intercept



(ag) Self-perceived health - fair



(ah) Self-perceived health - fair - models with functional intercept

32

(ai) Self-perceived health - poor

(aj) Self-perceived health - poor - models with functional intercept

Figure B.1: Histograms of functional slopes for FS-GBDT and FS-DNN (left side) and FIS-GBDT and FIS-DNN (right side).

## Appendix C. Hyperparameter search

Table C.1 summarises the hyperparameter search space and hyperparameters tuned for GBDT and DNN.

Table C.1: Hyperparameter search space and applicability.

| Parameter | Search space | Distribution | Applies to |
|---|---|---|---|
| Best iteration / epoch | Max 3000 iterations / 200 epochs | – | GBDT/DNN |
| lambda_l1 | $[10^{-8}, 1]$ | Log-uniform | GBDT/DNN |
| lambda_l2 | $[10^{-8}, 1]$ | Log-uniform | GBDT/DNN |
| num_leaves | $[2, 256]$ | Discrete uniform | GBDT |
| feature_fraction | $[0.4, 1]$ | Uniform | GBDT |
| bagging_fraction | $[0.4, 1]$ | Uniform | GBDT |
| bagging_freq | $[1, 7]$ | Discrete uniform | GBDT |
| min_data_in_leaf | $[1, 200]$ | Discrete uniform | GBDT |
| max_bin | $[64, 511]$ | Discrete uniform | GBDT |
| min_sum_hessian_in_leaf | $[10^{-8}, 10]$ | Log-uniform | GBDT |
| min_gain_to_split | $[10^{-8}, 10]$ | Log-uniform | GBDT |
| batch_size | $\{256, 512\}$ | Categorical | DNN |
| learning_rate | $[0.0001, 0.01]$ (fixed 0.05 or 0.1 for GBDT) | Log-uniform | GBDT/DNN |
| dropout | $[0.0, 0.9]$ | Uniform | DNN |
| act_func | $\{ReLU, Sigmoid, Tanh\}$ | Categorical | DNN |
| batch_norm | $\{True, False\}$ | Categorical | DNN |
| layer_sizes | $\{[32], [64], [128], [32,32], [64,64], [128,128], [64,128], [128,64], [64,128,64]\}$ | Categorical | DNN |

Table C.2 provides an overview of the hyperparameter search range and optimal values for all models for the synthetic dataset of Section 4.

Table C.3 provides an overview of the hyperparameter search range and optimal values for all models for the Swissmetro dataset of Section 5.1.

Table C.4 provides an overview of the hyperparameter search range and optimal values for all models for the LPMC dataset of Section 5.1.

Table C.5 provides an overview of the hyperparameter search for the case study of Section 5.2 while reporting all optimal values for all models.

33

Table C.2: Hyperparameter search for the synthetic dataset of Section 4.

|  | FI-RUMBoost | FI-DNN |
|---|---|---|
| Val. MCEL | 1.347 | 1.346 |
| Time [s] | 232 | 2788 |
| Best it./ep. | 340 | 14 |
| lambda_l1 | 0.000 | 0.012898 |
| lambda_l2 | 0.000 | 0.000004 |
| num_leaves | 94 | - |
| feature_fraction | 0.437 | - |
| bagging_fraction | 0.491 | - |
| bagging_freq | 1 | - |
| min_data_in_leaf | 57 | - |
| max_bin | 85 | - |
| min_sum_hessian_in_leaf | 0.108 | - |
| min_gain_to_split | 6.118 | - |
| batch_size | - | 512 |
| learning_rate | 0.1* | 0.002289 |
| dropout | - | 0.193 |
| act_func | - | sigmoid |
| batch_norm | - | True |
| layer_sizes | - | [128, 64] |

*fixed

Table C.3: Hyperparameter search for the Swissmetro dataset of Section 5.1. OL means Ordinal Logit.

|  | FIS-GBDT | FS-GBDT | FI-RUMBoost | RUMBoost | FIS-DNN | FS-DNN | FI-DNN | OL | GBDT | DNN |
|---|---|---|---|---|---|---|---|---|---|---|
| Val. MCEL | 0.639 | 0.695 | 0.633 | 0.768 | 0.607 | 0.648 | 0.676 | 0.835 | 0.610 | 0.630 |
| Time [s] | 807 | 770 | 219 | 129 | 684 | 523 | 852 | 1093 | 1037 | 344 |
| Best it./ep. | 797 | 785 | 399 | 960 | 58 | 37 | 130 | 162 | 3000 | 38 |
| lambda_l1 | 0.000 | 0.000 | 0.053 | 0.000 | 0.002 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| lambda_l2 | 0.000 | 0.003 | 0.000 | 0.000 | 0.005 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| num_leaves | 150 | 77 | 244 | 165 | - | - | - | - | 249 | - |
| feature_fra. | 0.892 | 0.768 | 0.677 | 0.403 | - | - | - | - | 0.501 | - |
| bagging_fra. | 1.000 | 0.879 | 0.443 | 0.999 | - | - | - | - | 0.478 | - |
| bagging_fre. | 4 | 6 | 7 | 2 | - | - | - | - | 1 | - |
| min_data. | 62 | 108 | 39 | 32 | - | - | - | - | 6 | - |
| max_bin | 426 | 270 | 272 | 282 | - | - | - | - | 104 | - |
| min_sum_h. | 0.258 | 0.000 | 0.000 | 0.000 | - | - | - | - | 0.000 | - |
| min_gain. | 0.003 | 0.002 | 0.000 | 0.001 | - | - | - | - | 0.652 | - |
| batch_size | - | - | - | - | 256 | 512 | 256 | 512 | - | 512 |
| learning_r. | 0.100* | 0.100* | 0.100* | 0.100* | 0.005 | 0.006 | 0.006 | 0.009 | 0.002 | 0.008 |
| dropout | - | - | - | - | 0.449 | 0.679 | 0.892 | 0.000 | - | 0.728 |
| act_func | - | - | - | - | tanh | relu | sigmoid | - | - | tanh |
| batch_norm | - | - | - | - | False | True | True | False | - | False |
| layer_sizes | - | - | - | - | [64, 128] | [64, 128, 64] | [64, 128] | - | - | [128, 128] |

*fixed

Table C.4: Hyperparameter search for the LPMC dataset of Section 5.1.

|  | FIS-GBDT | FS-GBDT | FI-RUMBoost | RUMBoost | FIS-DNN | FS-DNN | FI-DNN | MNL | GBDT | DNN |
|---|---|---|---|---|---|---|---|---|---|---|
| Val. MCEL | 0.668 | 0.677 | 0.645 | 0.809 | 0.657 | 0.670 | 0.664 | 0.835 | 0.795 | 0.794 |
| Time [s] | 1464 | 1169 | 264 | 296 | 5519 | 6027 | 5045 | 4718 | 2065 | 2508 |
| Best it./ep. | 511 | 410 | 506 | 886 | 85 | 84 | 91 | 61 | 3000 | 66 |
| lambda_l1 | 0.000 | 0.000 | 0.591 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.001 |
| lambda_l2 | 0.001 | 0.000 | 0.076 | 0.000 | 0.001 | 0.068 | 0.000 | 0.000 | 0.011 | 0.273 |
| num_leaves | 3 | 3 | 73 | 236 | - | - | - | - | 21 | - |
| feature_fra. | 0.583 | 0.860 | 0.960 | 0.978 | - | - | - | - | 0.803 | - |
| bagging_fra. | 0.759 | 0.720 | 0.592 | 0.634 | - | - | - | - | 0.436 | - |
| bagging_fre. | 4 | 7 | 2 | 2 | - | - | - | - | 5 | - |
| min_data. | 87 | 37 | 124 | 61 | - | - | - | - | 172 | - |
| max_bin | 470 | 152 | 85 | 474 | - | - | - | - | 337 | - |
| min_sum_h. | 0.000 | 0.000 | 0.000 | 3.083 | - | - | - | - | 0.000 | - |
| min_gain. | 0.000 | 0.870 | 2.373 | 0.001 | - | - | - | - | 0.000 | - |
| batch_size | - | - | - | - | 256 | 256 | 256 | 256 | - | 256 |
| learning_r. | 0.100* | 0.100* | 0.100* | 0.100* | 0.000 | 0.000 | 0.010 | 0.010 | 0.002 | 0.004 |
| dropout | - | - | - | - | 0.224 | 0.739 | 0.533 | 0.000 | - | 0.541 |
| act_func | - | - | - | - | tanh | tanh | tanh | - | - | tanh |
| batch_norm | - | - | - | - | True | True | False | False | - | False |
| layer_sizes | - | - | - | - | [32, 32] | [64, 64] | [32] | - | - | [64, 128, 64] |

*fixed

Table C.5: Hyperparameter search for the case study of Section 5.2 with the easySHARE dataset.

| | FIS-GBDT | FS-GBDT | FI-RUMBoost | RUMBoost | FIS-DNN | FS-DNN | FI-DNN | MNL |
|---|---|---|---|---|---|---|---|---|
| Val. MCEL | 0.253 | 0.253 | 0.253 | 0.261 | 0.253 | 0.253 | 0.254 | 0.262 |
| Time [s] | 4512 | 5683 | 7752 | 32617 | 9683 | 13214 | 12194 | 8795 |
| Best it./ep. | 385 | 517 | 867 | 2259 | 25 | 29 | 20 | 30 |
| lambda_l1 | 0.521 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| lambda_l2 | 0.000 | 0.000 | 0.001 | 0.000 | 0.000 | 0.000 | 0.036 | 0.000 |
| num_leaves | 3 | 3 | 11 | 74 | - | - | - | - |
| feature_fraction | 0.788 | 0.708 | 0.492 | 0.720 | - | - | - | - |
| bagging_fraction | 0.998 | 0.998 | 0.879 | 0.863 | - | - | - | - |
| bagging_freq | 2 | 1 | 1 | 3 | - | - | - | - |
| min_data_in_leaf | 159 | 173 | 194 | 127 | - | - | - | - |
| max_bin | 209 | 363 | 95 | 219 | - | - | - | - |
| min_sum_hessian_in_leaf | 0.000 | 0.098 | 2.234 | 0.000 | - | - | - | - |
| min_gain_to_split | 0.000 | 3.631 | 5.169 | 0.000 | - | - | - | - |
| batch_size | - | - | - | - | 256 | 256 | 256 | 256 |
| learning_rate | 0.05* | 0.052* | 0.05* | 0.052* | 0.001 | 0.002 | 0.004 | 0.001 |
| dropout | - | - | - | - | 0.659 | 0.003 | 0.553 | 0 |
| act_func | - | - | - | - | sigmoid | sigmoid | sigmoid | - |
| batch_norm | - | - | - | - | True | False | False | False |
| layer_sizes | - | - | - | - | [128, 64] | [32, 32] | [32, 32] | - |

*fixed

## Appendix D. Ordinal threshold values

Table D.6 summarises the ordinal threshold values for all models of Section 5.2.

Table D.6: Ordinal threshold values for the case study of Section 5.2.

| Thresholds | FIS-GBDT | FS-GBDT | FI-RUMBoost | RUMBoost | FIS-DNN | FS-DNN | FI-DNN | Ord. Logit |
|---|---|---|---|---|---|---|---|---|
| 1 | -1.019 | -1.003 | -1.865 | -1.585 | -1.438 | -1.265 | -1.190 | -1.509 |
| 2 | 0.148 | 0.163 | -0.694 | -0.454 | -0.257 | -0.085 | -0.012 | -0.374 |
| 3 | 1.033 | 1.052 | 0.203 | 0.614 | 0.614 | 0.803 | 0.874 | 0.458 |
| 4 | 1.792 | 1.806 | 0.956 | 1.137 | 1.387 | 1.572 | 1.650 | 1.217 |
| 5 | 2.528 | 2.541 | 1.688 | 1.846 | 2.131 | 2.314 | 2.396 | 1.934 |
| 6 | 3.262 | 3.276 | 2.420 | 2.556 | 2.846 | 3.044 | 3.097 | 2.608 |
| 7 | 3.999 | 4.013 | 3.149 | 3.268 | 3.556 | 3.764 | 3.811 | 3.300 |
| 8 | 4.742 | 4.757 | 3.883 | 3.986 | 4.283 | 4.478 | 4.509 | 4.032 |
| 9 | 5.560 | 5.571 | 4.682 | 4.775 | 5.084 | 5.290 | 5.288 | 4.822 |
| 10 | 6.568 | 6.572 | 5.711 | 5.711 | 6.035 | 6.267 | 6.232 | 5.748 |
| 11 | 7.755 | 7.789 | 6.900 | 6.978 | 7.260 | 7.510 | 7.463 | 6.975 |
| 12 | 8.890 | 8.957 | 8.136 | 8.366 | 8.805 | 9.137 | 9.097 | 8.571 |

## References

Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M., 2019. Optuna: A next-generation hyperparameter optimization framework, in: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, pp. 2623–2631.

Ansel, J., Yang, E., He, H., Gimelshein, N., Jain, A., Voznesensky, M., Bao, B., Bell, P., Berard, D., Burovski, E., Chauhan, G., Chourdia, A., Constable, W., Desmaison, A., DeVito, Z., Ellison, E., Feng, W., Gong, J., Gschwind, M., Hirsh, B., Huang, S., Kalambarkar, K., Kirsch, L., Lazos, M., Lezcano, M., Liang, Y., Liang, J., Lu, Y., Luk, C.K., Maher, B., Pan, Y., Puhrsch, C., Reso, M., Saroufim, M., Siraichi, M.Y., Suk, H., Zhang, S., Suo, M., Tillet, P., Zhao, X., Wang, E., Zhou, K., Zou, R., Wang, X., Mathews, A., Wen, W., Chanan, G., Wu, P., Chintala, S., 2024. Pytorch 2: Faster machine learning through dynamic python bytecode transformation and graph compilation, in: Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2, Association for Computing Machinery, New York, NY, USA. p. 929–947. URL: https://doi.org/10.1145/3620665.3640366, doi:10.1145/3620665.3640366.

Bierlaire, M., 2023. A short introduction to biogeme. Transport and Mobility Laboratory, ENAC, EPFL .

Bierlaire, M., Axhausen, K., Abay, G., 2001. The acceptance of modal innovation: The case of swissmetro, in: Swiss transport research conference.

Börsch-Supan, A., Brandt, M., Hunkler, C., Kneip, T., Korbmacher, J., Malter, F., Schaan, B., Stuck, S., Zuber, S., 2013. Data resource profile: the survey of health, ageing and retirement in europe (share). International journal of epidemiology 42, 992–1001.

Chen, T., Guestrin, C., 2016. Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, pp. 785–794.

Fokkema, M., Smits, N., Zeileis, A., Hothorn, T., Kelderman, H., 2018. Detecting treatment-subgroup interactions in clustered data with generalized linear mixed-effects model trees. Behavior research methods 50, 2016–2034.

Friedman, J.H., 2001. Greedy function approximation: a gradient boosting machine. Annals of statistics , 1189–1232.

Goodfellow, I., Bengio, Y., Courville, A., Bengio, Y., 2016. Deep learning. volume 1. MIT press Cambridge.

Greene, W., 2015. Panel data models for discrete choice, in: The Oxford Handbook of Panel Data. Oxford University Press. URL: https://doi.org/10.1093/oxfordhb/9780199940042.013.0006, doi:10.1093/oxfordhb/9780199940042.013.0006.

Gruber, S., Hunkler, C., Stuck, S., 2014. Generating easyshare: guidelines, structure, content and programming. SHARE Work. Pap. Ser .

Hajjem, A., Bellavance, F., Larocque, D., 2014. Mixed-effects random forest for clustered data. Journal of Statistical Computation and Simulation 84, 1313–1328.

Hajjem, A., Larocque, D., Bellavance, F., 2017. Generalized mixed effects regression trees. Statistics & Probability Letters 126, 114–118.

Han, Y., Pereira, F.C., Ben-Akiva, M., Zegras, C., 2022. A neural-embedded discrete choice model: Learning taste representation with strengthened interpretability. Transportation Research Part B: Methodological 163, 166–186.

Hess, S., Train, K.E., Polak, J.W., 2006. On the use of a modified latin hypercube sampling (mlhs) method in the estimation of a mixed logit model for vehicle choice. Transportation Research Part B: Methodological 40, 147–163.

Hillel, T., Elshafie, M.Z., Jin, Y., 2018. Recreating passenger mode choice-sets for transport simulation: A case study of london, uk. Proceedings of the Institution of Civil Engineers-Smart Infrastructure and Construction 171, 29–42.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.Y., 2017. Lightgbm: A highly efficient gradient boosting decision tree. Advances in neural information processing systems 30.

Kilian, P., Ye, S., Kelava, A., 2023. Mixed effects in machine learning–a flexible mixedml framework to add random effects to supervised machine learning regression. Transactions on Machine Learning Research .

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 .

Krueger, R., Bierlaire, M., Daziano, R.A., Rashidi, T.H., Bansal, P., 2021. Evaluating the predictive abilities of mixed logit models with unobserved inter-and intra-individual heterogeneity. Journal of choice modelling 41, 100323.

Mandel, F., Ghosh, R.P., Barnett, I., 2023. Neural networks for clustered and longitudinal data using mixed effects models. Biometrics 79, 711–721.

Mendorf, S., Schönenberg, A., Heimrich, K.G., Prell, T., 2023. Prospective associations between hand grip strength and subsequent depressive symptoms in men and women aged 50 years and older: insights from the survey of health, aging, and retirement in europe. Frontiers in Medicine 10, 1260371.

Ngufor, C., Van Houten, H., Caffo, B.S., Shah, N.D., McCoy, R.G., 2019. Mixed effect machine learning: A framework for predicting longitudinal change in hemoglobin a1c. Journal of biomedical informatics 89, 56–67.

Panda, D.K., Ray, S., 2022. Approaches and algorithms to mitigate cold start problems in recommender systems: a systematic literature review. Journal of Intelligent Information Systems 59, 341–366.

Salvadé, N., Hillel, T., 2025. Rumboost: Gradient boosted random utility models. Transportation Research Part C: Emerging Technologies 170, 104897.

Sela, R.J., Simonoff, J.S., 2012. Re-em trees: a data mining approach for longitudinal and clustered data. Machine learning 86, 169–207.

SHARE-ERIC, 2024. easyshare. Release version: 9.0.0. SHARE-ERIC. Dataset. doi:10.6103/SHARE.easy.900.

Shi, X., Cao, W., Raschka, S., 2023. Deep neural networks for rank-consistent ordinal regression based on conditional probabilities. Pattern Analysis and Applications 26, 941–955.

Sifringer, B., Lurkin, V., Alahi, A., 2020. Enhancing discrete choice models with representation learning. Transportation Research Part B: Methodological 140, 236–261.

Sigrist, F., 2023. Latent gaussian model boosting. IEEE Transactions on Pattern Analysis and Machine Intelligence 45, 1894–1905. doi:10.1109/TPAMI.2022.3168152.

Train, K.E., 2009. Discrete choice methods with simulation. Cambridge university press.

Xiong, Y., Kim, H.J., Singh, V., 2019. Mixed effects neural networks (menets) with applications to gaze estimation, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 7743–7752.