Exploring the potential of computational graph-based gradients for choice modelling

Yan Liu[abc]*, Chiara Calastri[c], Thijs Dekker[c]

[a]School of Electronic and Information Engineering, Beihang University, Beijing, China, liuyan5852@qq.com, mail to:15021232@buaa.edu.cn; [b]Shenyuan Honors College, Beihang University, Beijing, China; [c]Institute for Transport Studies & Choice Modelling Centre, University of Leeds, UK

Exploring the potential of computational graph-based gradients for choice modelling

Choice modelling is widely used to analyse travel behaviour, but increasing model complexity leads to estimation challenges including increased model run times and multiple local optima. Computational Graph (CG) offers quick and accurate approximation of the likelihood function's gradient, thereby addressing a key limitation of traditional gradient calculation methods. This study contrasts the performance of CG-based, analytical, and numerical gradient calculation methods for latent class and mixed logit models. Our findings highlight that CG achieves precise gradient estimates whilst significantly reducing estimation time. Analytical and CG-based gradient methods are less likely to result in bad local optima compared to numerical derivatives when testing across a wide range of starting values. Although local optima still occur, CG's faster estimation allows feasible testing over a range of starting values. As such, it represents a valuable tool, given the significant implications of poor local optima in terms of key model outputs.

Keywords: choice modelling, estimation, computational graph, automatic differentiation, machine learning, travel behaviour

1 Introduction

Choice modelling is widely recognised in travel behaviour research for its ability to explain the factors driving variation in individual decisions, such as mode and route choice. By considering variables such as socio-demographic, travel and contextual characteristics, choice models provide insights that directly inform transport policy and urban planning. In line with an increasingly complex and connected society and the availability of large datasets on human behaviour, the complexity and dimensionality of choice models have grown substantially, necessitating improved computational techniques for estimation (Arteaga et al., 2022; Jiang and Anderson, 2024). Discrete Choice Model (DCM) coefficients are typically estimated using Maximum Likelihood Estimation (MLE). It is well known that the linear-in-parameters Multinomial Logit (MNL) model converges to a global optimum, except in cases of severe misspecification (Dow and Endersby, 2004). However, this simplistic model has rarely been used in recent literature, where analysts favour specifications that are more suited to capture behavioural heterogeneity, such as latent class (LC) or random parameter structures. The presence of unobserved individual-level parameters in these models, which are integrated out in the corresponding (log-)likelihood function, often results in the presence of multiple local optima, reducing the probability of arriving at the global optimum from an arbitrary vector of starting values for the parameters of interest (Lancsar et al., 2017; Liu et al., 2004). In this context, the global optimum is defined as the parameter vector that maximizes the log-likelihood function across the entire parameter space, representing the absolute best solution independent of initialization. In contrast, the best local optimum is defined as the highest log-likelihood solution obtained across a sufficiently large set of starting values. While the best local optimum may not necessarily be the global optimum, testing multiple starting values increases the probability of identifying a more optimal solution. Indeed, it is recognised that it is unlikely to arrive at the global optimum, and achieving the best local optimum is generally considered an acceptable outcome (Hess et al., 2006; McFadden, 2022).

Recent years have seen a growing interest in leveraging Machine Learning (ML) to enhance the estimation process of DCMs. Notably, Kim et al. (2021) and Ma et al. (2022) have demonstrated the significant speed advantages of using ML-based Computational Graphs (CG) alongside Automatic Differentiation (AD) for gradient approximation. Inspired by their work, our research introduces a new perspective by focusing on local optima, an often-overlooked aspect in the existing literature. While speed is a clear benefit, it is crucial to prioritize reaching the correct maximum—even if it requires more time—over quickly settling for a poor local optimum. This consideration is particularly significant in complex models, where different local optima can substantially impact key model outputs.

Whilst Kim et al. (2021) already established these properties, their empirical estimation approach lacked consistency because the use of different software packages limits fair comparison. We overcome this limitation by contrasting different gradient calculation mechanisms within the same software package (Python, TensorFlow, and TensorFlow Probability (Dürr et al., 2020) allowing for a fairer comparison and more robust results. While advanced ML modules (e.g., deep neural networks (DNNs) and kernel logistic regression (KLR)) are gaining traction in choice modeling, our focus is on evaluating computational graph-based automatic differentiation (CG-AD) for enhancing the estimation performance of existing DCMs. Beyond computational precision and speed, we also examine how various gradient methods influence the likelihood of attaining the best local optimum and reducing the risk of encountering poor local optima, which we refer to as stability. By testing numerous starting values, we evaluate the CG framework against other derivative calculation methods which are typically applied in choice modelling. Using the Latent Class (LC) and mixed logit (MIXL) models, we demonstrate that CG-AD is a flexible and effective approach for complex DCM estimation and large-scale model testing, enabling researchers to refine, modify, and iterate their econometric models more efficiently. Building on this, our research advances the intersection of ML and DCM by demonstrating how ML techniques enhance the choice modelling estimation process, with key contributions in estimation precision, speed, and stability:

(1) The CG framework ensures highly precise gradient estimation. Our results show that CG-AD corresponds to the analytical gradient up to the 10th or 11th decimal level of precision.

(2) CG-AD method significantly outperforms ND-based techniques in computational efficiency across all models. It achieves speed improvements exceeding tenfold for the MIXL models that require Monte Carlo simulations, making it efficient for large-scale applications.

(3) Despite their level of precision, CG-AD and analytical derivatives remain subject to the risk of local optima, particularly in latent class models. However, in the context of mixed logit, CG mitigates the risk of converging to poor local optima and thereby improves the stability of optimisation processes.

The remainder of this paper is organised as follows: Section 2 reviews the literature, exploring the current state of the integration of ML techniques and choice modelling in travel behaviour research. Section 3 outlines the methodological approach, presenting the mathematical formulation of the DCMs estimated in this paper. It also introduces the CG-AD framework and compares it with other gradient mechanisms. Section 4 details the empirical approach, describing the dataset and experimental setup, followed by Section 5, which presents the experimental results. In Section 6, we engage in a deeper discussion and detailed analysis of these results. Finally, Section 7 concludes the paper by offering closing remarks and suggesting directions for future research.

2 Literature Review

This section reviews the application of machine learning (ML) models, techniques, and practices in Discrete Choice Models (DCMs) and identifies gaps in current knowledge. The inherent similarities and complementary nature of ML and DCMs have led researchers to integrate the two methods in various ways (Salas et al., 2022; van Cranenburgh et al., 2022) , providing new insights into travel behaviour(Wang et al., 2018; Welch and Widita, 2019).

2.1 ML to model choice behaviour

Some studies have focused on using ML to directly model choice behaviour, treating travel decisions as classification problems (Hillel et al., 2021). By leveraging algorithms like support vector machines and multilayer perceptrons, researchers have demonstrated improvements in prediction accuracy over traditional logit models (García-García et al., 2022; Han et al., 2022; Lederrey et al., 2021; Nam et al., 2017; Omrani, 2015; Wang et al., 2021a; Wang et al., 2021b). Simultaneously, several researchers have explored the use of ML techniques to improve model building, leveraging their nonlinear representation ability to augment utility specifications (Sifringer et al., 2020; Wang et al., 2021a; Wong and Farooq, 2021), and learn taste parameters as flexible functions of input attributes (Han et al., 2022; Phan et al., 2022). For instance, Rodrigues et al.

(2020) introduced an approach combining Bayesian inference with automatic relevance determination, which aids in the automatic identification of optimal utility functions for DCMs. Martín-Baos et al. (2021) developed the PyKernelLogit package for Kernel Logistic Regression (KLR) model, which replaces the utility with kernel functions, thus freeing the analyst from the need to specify a functional form for the utilities. Łukawska et al. (2025) proposed the context-aware Bayesian mixed multinomial logit model (C-MMNL), using neural networks to capture systematic, context-dependent heterogeneity in preference parameters, allowing for flexible interactions while maintaining computational efficiency. Kamal and Farooq (2024) introduced the Ordinal-ResLogit model, integrating Wong and Farooq (2021)'s ResLogit into the consistent rank logit model to ensure consistency among binary classifiers and capture unobserved heterogeneity. This integration of ML with DCMs facilitates more efficient utility function specification, allowing for more flexibility in model construction.

However, both approaches—whether using ML alone or in combination with DCMs—face similar challenges in terms of interpretability. ML models, especially deep learning techniques, are often criticised for their "black box" nature and lead to studies in explainability (Górriz et al., 2023; Tjoa and Guan, 2021; Vilone and Longo, 2021). Despite their ability to achieve high prediction accuracy, the internal mechanisms of these models, involving numerous connections between input features, hidden layers, and output layers, remain difficult to fully comprehend. Emerging research has made efforts to investigate the interpretability of ML-integrated DCMs to bridge current knowledge gaps. Ali (2024) reviewed ML approaches for identifying key influencing factors, emphasizing the superiority of adjusted kernel and optimized ML models in capturing nonlinear relationships. Wang et al. (2020) demonstrated that DNNs can provide economic information as comprehensively as classical DCMs. Martín-Baos et al. (2024) derived economic indicators such as Willingness to Pay (WTP) and the Value of Time (VOT) from the proposed KLR model. However, challenges such as high sensitivity to hyperparameters and frequent non-identification due to local optima undermine the reliability of economic insights. Ali et al. (2023) compares the performance of classical DCMs and ML methods (NN and GBT), finding that although ML can reveal nonlinearities and threshold effects through PDP and SHAP, the direction of variable elasticities sometimes contradicts those of CM, limiting policy interpretability and underscoring the irreplaceable role of CM in transport research.

Moreover, ML methods can suffer from overfitting, particularly when applied to small datasets with complex structures and numerous parameters (Vabalas et al., 2019; Wang et al., 2021b). In contrast, DCMs are founded on a more transparent mathematical framework, with interpretable parameters that make them better suited for explaining causal relationships and understanding travel behaviour patterns (Wang et al., 2020). Thus, while ML offers powerful predictive capabilities, DCMs remain advantageous in terms of interpretability and theoretical robustness.

2.2 ML to support DCM estimation

Another important research direction involves leveraging ML techniques to enhance the estimation process in DCMs, which is also the focus of the present paper. A key challenge in this process is ensuring the stability and robustness of model estimation. Traditional estimation methods often struggle with multiple local optima, which complicates the identification of the best-identified solution among several suboptimal ones (Pacheco Paneque et al., 2021). A common strategy to mitigate this problem is experimenting with different starting values during the estimation process (Hess and Palma, 2019; Pál and Sándor, 2023; Vermeulen et al., 2008), which increases the likelihood of identifying a global optimum.

According to van Cranenburgh et al. (2022), ML can improve DCM estimation processes by addressing the limitations of traditional theory-driven modelling approaches. Inspired by the stochastic gradient descent (SGD) method commonly used in ML, Lederrey et al. (2021) proposed a mini-batch estimation approach for DCMs, significantly accelerating estimation speed and enhancing convergence stability. The estimation process can be improved by transitioning from maximum likelihood estimation (MLE) to penalized MLE (PMLE) with Ridge and LASSO regularization to mitigate overfitting and exclude extreme parameters (Martín-Baos et al., 2024). Martín-Baos et al. (2023) implemented the Nyström KLR model to reduce time complexity for large-scale datasets and found L-BFGS-B to be the most efficient optimization method compared to gradient descent, Momentum, and Adam. Within data science and ML, it is widely recognised that deep learning models depend fundamentally on the Computational Graph (CG) framework (LeCun et al., 2015). This framework provides the structural basis for neural networks, enabling efficient execution of complex operations, gradient calculations for backpropagation, and dynamic adjustment of network parameters during training. Verma (2000) and Margossian (2019) have shown that automatic differentiation (AD) and backpropagation are critical components of the CG mechanism, freeing researchers from manually coding gradients and preventing the truncation and round-off errors associated with numerical gradients (Baydin et al., 2018; Paszke et al., 2017).

Previous research has demonstrated CG's utility across domains, including traffic equilibrium and assignment problems (Liu et al., 2023; Wu et al., 2018), and traffic state and queue profile estimation (Lu et al., 2023). However, only a few studies, such as Kim et al. (2021) and Ma et al. (2022), have explored CG's potential in DCMs.

Kim et al. (2021) demonstrated CG's computational advantages in estimating models like MNL, Nested Logit (NL), and Integrated Choice and Latent Variable (ICLV) model, yielding parameter estimates comparable to traditional tools like Apollo and Biogeme. Similarly, Ma et al. (2022) implemented CG-based Multinomial Probit-based ICLV models, underscoring the efficiency of the AD mechanism in removing the need for manually coded gradients. Despite these contributions, most studies have focused on basic software-level comparisons without thoroughly investigating estimation stability. Moreover, cross-software comparisons limit result comparability, leaving a gap in understanding the underlying mechanisms behind CG's performance improvements. In what follows, we examine and contrast the performance of CG, within the same software environment, against alternative gradient-based methods not just in terms of estimation speed, but also in its ability to find the best local optimum.

3 Methods

This section introduces the Computational Graph (CG)-based framework for estimating Discrete Choice Models (DCMs). We begin by introducing the choice models of interest, i.e. the Multinomial Logit (MNL) model, the Latent Class (LC) model, and the Mixed Logit (MIXL) model. Using the MNL model as a simplified example, we introduce traditional analytical and numerical gradient methods. This is followed by an in-depth explanation of CG with Automatic Differentiation (CG-AD). Finally, a comprehensive comparison of these gradient methods is provided, highlighting the theoretical advantages of AD. Throughout the paper, we refer to CG-AD and AD interchangeably as they effectively refer to the same concept in the context of this paper. For clarity, all symbols used in this section are defined in the Table 1.

3.1 Choice modelling approaches

3.1.1 Multinomial Logit (MNL) model

The MNL model stands as a foundational choice modelling approach within the family of generalised linear models. The MNL model operates under the assumption that the error terms are Independent and Identically Distributed (IID) and follow an extreme-value type I (Gumbel) distribution. Notably, due to the convex nature of its likelihood function, the MNL model exhibits only a global optimum.

Given a choice set with I alternatives, the utility for individual q choosing alternative i at choice occasion t is represented as the sum of an observed component, V_{itq}, and a random component, ε_{itq}:

$$U_{itq} = V_{itq} + \varepsilon_{itq} \tag{1}$$

where V_{itq} is typically a linear function of alternative attributes and individual characteristics, $\boldsymbol{X}_{itq}$, and their corresponding coefficients, $\boldsymbol{\beta}$:

$$V_{itq} = \boldsymbol{X}_{itq}\boldsymbol{\beta} \tag{2}$$

Given that the error terms, ε_{itq}, follow a Gumbel distribution, the probability that individual q chooses alternative i can be expressed as:

$$P_{y_{itq}=1} = \frac{e^{V_{itq}}}{\sum_{i\prime=1}^{I} e^{V_{i\prime tq}}} = \frac{e^{\boldsymbol{X}_{itq}\boldsymbol{\beta}}}{\sum_{i\prime=1}^{I} e^{\boldsymbol{X}\prime_{itq}\boldsymbol{\beta}}} \tag{3}$$

where y_{itq} is the choice variable of the q^{th} traveller, which equals 1 if the traveller selects the i^{th} alternative and 0 otherwise.

The joint probability, P_{y_q}, for decision maker q choosing y_q from I alternatives across T time periods is then calculated as:

$$P_{y_q} = \prod_{t=1}^{T_q} \prod_{i=1}^{I} P_{y_{itq}=1} \tag{4}$$

The coefficients are estimated by maximising the log-likelihood function over Q independent decision-makers, expressed as:

$$LL(y) = \sum_{q=1}^{Q} \ln P_{y_q} = \sum_{q=1}^{Q} \ln \left(\prod_{t=1}^{T_q} \prod_{i=1}^{I} P_{y_{itq}=1} \right) \tag{5}$$

3.1.2 Latent Class (LC) model

The LC model extends the MNL model by accounting for unobserved heterogeneity among decision-makers. While the MNL model assumes that all individuals make choices based on the same set of parameters, the LC model allows for distinct segments or classes within the respondents, each characterised by its own set of parameters, thereby capturing latent heterogeneity (Heckman and Singer, 1984; Swait, 1994). This paper focuses on the LC-MNL model. Consider a decision-maker, q, facing I alternatives in each of T choice situations. The probability that q will choose alternative i on occasion t is given by:

$$P_{y_{itq}=1} = \sum_{j=1}^{J} P_{y_{itq}=1|j}\, P_{c_{qj}=1} \tag{6}$$

Assume that in the conditional utility $U_{itq|j}$ in Equation (7), the error terms $\varepsilon_{itq|j}$ are IID across alternatives, decision-makers and classes, and follow a Gumbel distribution. The conditional probability, $P_{y_{itq}=1|j}$, for individual q choosing alternative i within class j during the choice occasion t can then be derived as follows:

$$U_{itq|j} = V_{itq|j} + \varepsilon_{itq|j} = \boldsymbol{X'}_{itq} \boldsymbol{\beta}_j + \varepsilon_{itq|j} \tag{7}$$

$$P_{y_{itq}=1|j} = \frac{e^{V_{itq|j}}}{\sum_{i'=1}^{I} e^{V_{i'tq|j}}} = \frac{e^{X_{itq}\beta_j}}{\sum_{i'=1}^{I} e^{X_{i'tq}\beta_j}} \tag{8}$$

where $\boldsymbol{X}_{itq}$ is a vector of exogenous attributes, including a constant, and $\boldsymbol{\beta}_j$ is the corresponding coefficients of class j to be estimated that could be accommodated for class heterogeneity.

$P_{c_{qj}=1}$ represents the probability that respondent q belongs to latent class j, also known as class membership, as shown in Equation (9). This probability is derived from the class utility in Equation (10).

$$P_{c_{qj}=1} = \frac{e^{V_{qj}}}{\sum_{j'=1}^{J} e^{V_{qj}}} = \frac{e^{S'_q \gamma_j}}{\sum_{j'=1}^{J} e^{S'_q \gamma_{j'}}} \tag{9}$$

$$U_{qj} = V_{qj} + \zeta_{qj} = \boldsymbol{S'}_q \boldsymbol{\gamma}_j + \zeta_{qj} \tag{10}$$

where $\boldsymbol{S}_q$ represents the explanatory variables of individual q for class allocation, including a constant, $\boldsymbol{\gamma}_j$ is the corresponding coefficient vector, and ζ_{qj} is an error term that captures random disturbances, following an IID Type I extreme value distribution across classes.

The joint probability P_{y_q} and the log-likelihood function are computed as Equations (4) and (5). Coefficients are estimated using the maximum likelihood method, but careful selection of initial values is essential, as the LC-MNL model may exhibit multiple local optima.

3.1.3 Mixed Logit (MIXL) model

The MIXL model is a flexible approach that overcomes the restrictive IID assumption inherent in the MNL model and can approximate any random utility model (McFadden and Train, 2000). Unlike the LC-MNL model where choice preferences are assumed to follow a discrete distribution and unobserved heterogeneity is associated with class-specific parameters, the MIXL model assumes that parameters follow continuous distributions, such as normal or lognormal. In contrast to both the MNL and LC-MNL models, the mixed logit model does not have closed-form choice probability expressions (Hensher and Greene, 2003). Through maximum simulated likelihood estimation, researchers can solve such open-form models by drawing pseudo-random realisations, simulating choice probabilities, and estimating the corresponding parameters.

Similar to the LC model, individuals are faced with I alternatives during T choice occasions. The probability that an individual q chooses alternative i in situation t is expressed as shown in Equation (11).

$$P_{y_{itq}=1} = \int L_{itq} f(\boldsymbol{\beta}) d\boldsymbol{\beta} \tag{11}$$

Here, $f(\beta)$ is a density function, which can take various forms, such as normal, lognormal, or uniform. The MIXL model can be viewed as a mixture of logit functions evaluated at different $\boldsymbol{\beta}$ values, with $f(\boldsymbol{\beta})$ serving as the mixing distribution. Due to the IID assumption and the Gumbel distribution of the random disturbance ε_{itq} in Equation (13), L_{itq} can be computed as shown in Equation (12).

$$L_{itq} = \frac{e^{V_{itq}}}{\sum_{i'=1}^{I} e^{V_{i'tq}}} = \frac{e^{X_{itq}\boldsymbol{\beta}}}{\sum_{i'=1}^{I} e^{X_{i'tq}\boldsymbol{\beta}}} \tag{12}$$

$$U_{itq} = V_{itq} + \varepsilon_{itq} = \boldsymbol{X'}_{itq}\boldsymbol{\beta} + \varepsilon_{itq} \tag{13}$$

In this expression, $\boldsymbol{X}_{itq}$ represents the vector of explanatory variables considered in the choice model, which includes attributes of alternatives and socio-economic characteristics. $V_{itq}(\boldsymbol{\beta})$ denotes the observable component of utility.

By employing methods such as Halton sequences to draw values of $\boldsymbol{\beta}$ based on its distribution, we can simulate the non-closed probability $P_{y_{itq}=1}$ using Monte Carlo simulation methods, as presented in Equation (14).

$$SP_{y_{itq}=1} = \frac{1}{R}\sum_{r=1}^{R} L_{itq}\,(\boldsymbol{\beta}_r) \tag{14}$$

where R denotes the number of replications (i.e., draws of $\boldsymbol{\beta}_r$), $\boldsymbol{\beta}_r$ represents the r^{th} draw of $\boldsymbol{\beta}$, and SP_{itq} is the simulated probability that individual q chooses alternative i on occasion t. By substituting the simulated value of $P_{y_{itq}=1}$ into Equation (15), the log-likelihood function can be computed using the simulation method.

3.2 Traditional gradient methods: analytical and numerical algorithms

3.2.1 Analytical Differentiation (AnaD)

Analytical Differentiation (AnaD) provides an exact gradient expression derived through calculus. It is typically associated with fewer function evaluations, making it particularly fast in high-dimensional spaces and highly efficient for estimating DCMs (Mai et al., 2015). Moreover, AnaD is less sensitive to hyperparameters of the estimation routine such as step size, which are crucial in numerical methods. However, the likelihood function of DCMs is often complex, making AnaD challenging to compute and implement (Bansal et al., 2018).

In the context of the MNL model, the log-likelihood function LL as presented in Equation (15) is both smooth and differentiable (Train, 2009). Its derivative can be efficiently computed as follows:

$$\begin{aligned}\frac{\partial LL}{\partial \boldsymbol{\beta}} &= \frac{\sum_q \sum_t \sum_i y_{itq}\left(\boldsymbol{X}'_{itq}\boldsymbol{\beta}\right)}{\partial \boldsymbol{\beta}} - \frac{\sum_q \sum_t \sum_i y_{itq}\ln\left(\sum_j e^{\boldsymbol{X}'_{itq}\boldsymbol{\beta}}\right)}{\partial \boldsymbol{\beta}} \\ &= \sum_q \sum_t \sum_i y_{itq}\,\boldsymbol{X}_{itq} - \sum_q \sum_t \sum_i y_{itq} \sum_j P_{jtq}\,\boldsymbol{X}_{jtq} \\ &= \sum_q \sum_t \sum_i \left(y_{itq} - P_{itq}\right)\boldsymbol{X}_{itq}\end{aligned} \tag{15}$$

3.2.2 Numerical Differentiation (ND)

Numerical Differentiation (ND) refers to the gradient (or derivative) approximation of the given function, using finite differences evaluated at selected sample points. The simplest approach for computing the ND is to employ the definition of the limit derivative (Oliver, 1980). Within this category, the forward difference method, backward difference method, and central difference method are commonly utilised for approximating derivatives (Smith, 1985).

For a small positive number h, the gradient of a multivariate function $f(\mathbf{x})$ at a point $\mathbf{x}$) is $\nabla f = \left(\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n}\right)$, which can be approximated as follows:

(1) Forward Difference Method:

$$\frac{\partial f}{\partial \mathbf{x}} \approx \frac{f(\mathbf{x}+h\mathbf{e}_\mathrm{v}) - f(\mathbf{x})}{h} \tag{16}$$

(2) Backward Difference Method:

$$\frac{\partial f}{\partial \mathbf{x}} \approx \frac{f(\mathbf{x}) - f(\mathbf{x}-h\mathbf{e}_\mathrm{v})}{h} \tag{17}$$

(3) Central Difference Method:

$$\frac{\partial f}{\partial \mathbf{x}} \approx \frac{f(\mathbf{x}+h\mathbf{e}_\mathrm{v}) - f(\mathbf{x}-h\mathbf{e}_\mathrm{v})}{2h} \tag{18}$$

The forward difference method calculates the derivative at a point by considering the difference between the function values at that point and at a slightly larger point. Conversely, the backward difference method calculates the derivative by considering the difference between the function values at that point and at a slightly smaller point. These two methods primarily differ in the direction in which they evaluate the function values relative to the point of interest. In contrast, the central difference method, also known as the symmetric difference quotient method, estimates the derivative by averaging the forward and backward differences, thereby mitigating first-order errors and offering a more accurate approximation. These methods can also be extended to compute partial derivatives and gradients of multivariable functions.

In practice, ND methods, particularly the central difference method, are commonly employed when gradients are required but difficult to compute analytically, as often encountered in optimisation problems. Consequently, these methods are also widely used in software packages for DCMs. However, despite their ease of implementation, ND methods have notable limitations (Baydin et al., 2018):

- They are susceptible to round-off errors, and the accuracy of the gradient depends on the choice of step size h. A small h can lead to sensitivity to numerical errors, while a large h may result in inaccurate approximations (Jerrell, 1997).
- They are computationally expensive for high-dimensional functions, as the function must be evaluated at least twice for each dimension at a single point of interest (Margossian, 2019).

3.3 Automatic Differentiation (AD): underlying mechanism of computational graph

In the field of machine learning, it is widely recognised that CG combined with AD typically result in faster, more accurate, and more stable derivative calculations compared to ND. This makes AD a fundamental mechanism for building models in deep learning platforms such as TensorFlow and PyTorch (Margossian, 2019; Paszke et al., 2017). CG is a visual representation of mathematical operations for a complex function, where nodes represent variables or operations and edges represent dependencies between them. AD is a technique used within CG to compute the derivatives of the function by decomposing them into elementary operations, each of which is executed precisely and swiftly by the computer, achieving numerical precision up to machine accuracy. This process ensures that the computed derivatives closely align with AnaD from a theoretical perspective (Guarda et al., 2024; Ma et al., 2020; Wu et al., 2018). Therefore, AD allows for accurate and efficient computations during the training process, enhancing model estimation and inherently avoiding the approximation errors common with numerical methods.

This paper employs the TensorFlow platform (Abadi et al., 2016) to develop the CG-based MNL model. The forward and reverse modes of AD, when applied within CG, optimise memory usage and facilitate effective derivative calculations. The forward propagation process of the CG-based MNL model is depicted in Figure 1, where input data traverse through a network to generate an output in the forward direction, as described by Equations (1) to (4). In Figure 1, input nodes x_i represent attributes for the i^{th} alternative and personal characteristics from the empirical dataset, while the β_i are estimated coefficients, including alternative-specific constants. Intermediate CG nodes in grey colour such as N_1,N_{11},N_{21},N_{31} represent elementary decomposition functions and facilitate transitions between inputs and outputs. During estimation, variables are estimated by maximising the log-likelihood function LL, as expressed in Equation (4). In the forward pass, a single sweep through the CG computes both the numerical values and derivatives for all nodes, delivering results with machine-level accuracy. In the backward pass, the only necessities are access to the expression graph and the numerical values of the intermediate variables (Baydin et al., 2018). The calculation of derivatives is then automatically performed by applying the chain rule to these operations. This unique combination of graph structure, the forward-backward mode, the memory mechanism, and the chain-rule-based computation process contributes to the efficiency of the CG.

Specifically, the derivative of the variable with respect to the negative log-likelihood function can be computed using the nodes and links illustrated in Figure 1. The derivative of $\boldsymbol{\beta}$ can be expressed as $\frac{d\mathrm{LL}(\beta)}{\partial\boldsymbol{\beta}} = \left(\frac{\partial LL}{\partial\beta_1}, \dots, \frac{\partial LL}{\partial\beta_I}\right)$, where $\frac{\partial LL}{\partial\beta_k}$ is determined using the chain rule:

$$\frac{\partial LL}{\partial\beta_k} = \sum_{i=1}^{I} \frac{\partial LL}{\partial P_i}\frac{\partial P_i}{\partial N_{2i}}\frac{\partial N_{2i}}{\partial N_{1i}}\frac{\partial N_{1i}}{\partial\gamma_i}\frac{\partial\gamma_i}{\partial\beta_k} \tag{19}$$

3.4 Comparison of gradient methods

To clarify the distinctions between the derivation methods discussed in this study, we present a comparative analysis of their merits and drawbacks, as shown in Table 2.

These differentiation methods include (1) AnaD, (2) ND, and (3) AD:

(1) *AnaD* involves manually calculating derivative expressions and coding them for model calibration. While it provides highly optimised computation when performed accurately, the manual calculation of derivatives is prone to errors and is time-consuming(Griewank and Walther, 2008).

(2) *ND* employs a straightforward formula to approximate derivatives. Despite its simplicity and widespread use, it is susceptible to calculation errors and can be excessively time-consuming. The time complexity for approximating derivatives

increases linearly with the dimensions of partial derivatives (Gebremedhin and Walther, 2020).

(3) *AD* computes exact derivatives by deconstructing functions into elementary operations and applying the chain rule. It offers attractive time complexity for gradient computation but requires substantial memory to store all intermediate gradient values. Efficient implementation is necessary, as demonstrated in frameworks like TensorFlow and PyTorch (Paszke et al., 2017).

Compared to the AnaD approach which requires manually specifying derivatives, CG-based structures can automatically and swiftly compute derivatives with respect to the log-likelihood function, ensuring both accuracy and efficiency. In contrast to ND, the CG-based approach avoids truncation and round-off errors, thereby enhancing computational efficiency (Baydin et al., 2018; Gebremedhin and Walther, 2019; Margossian, 2019).

4 Empirical Approach

Our objective is to explore the potential of Computational Graph (CG) for the precise and efficient estimation of Discrete Choice Models (DCMs), specifically the Latent Class (LC) model and mixed logit (MIXL) model.

The computational experiments involved in this study are conducted on a system equipped with an Intel Core i7 processor and 16GB of RAM, operating on Microsoft Windows 11 (version 10.0.22621). The software environment utilised for our experiments is composed of Python version 3.8.15, augmented by TensorFlow version 2.9.1, and TensorFlow Probability version 0.17.0. This operating environment furnishes the necessary computational and statistical capabilities for our research.

4.1 Empirical Dataset

The dataset used in this research is derived from a Stated Preference (SP) survey conducted in Switzerland, which investigates public transport route choices (Axhausen et al., 2006). The dataset is publicly available through the Apollo website (Hess and Palma, 2019). A summary of the dataset is provided in Table 3.

The panel dataset contains 3,492 observations from 388 participants, with each participant facing nine choice situations, each involving a selection between two public transport routes. Each alternative route is characterised by four attributes: travel time, travel cost, headway (the interval between consecutive buses or trains), and the number of interchanges required. Additionally, the dataset includes socio-economic characteristics and trip-specific details for each respondent, such as income, household car availability, and the purpose of the journey (e.g., commuting, shopping, business, or leisure). This dataset provides a robust foundation for evaluating various choice modelling approaches in the context of public transport route selection.

4.2 Empirical setting

Our experiments began with the Multinomial Logit (MNL) model. However, due to its convex nature, starting values and gradient methods have minimal impact on the optimal solution, hence it is not detailed in this paper. As outlined in Section 4.1, the decision framework involves two route choices, each characterised by four distinct attributes. The LC model involves two latent classes, devoid of any covariates within the class allocation model, with each class having a unique set of parameters. When considering the MIXL with either lognormal or normal density, every attribute-related parameter follows an uncorrelated lognormal or normal distribution within the preference space. Given that the MIXL models require simulation to approximate the integrals for choice probabilities, we employ 5,000 Halton draws to simulate the likelihood. To comprehensively evaluate aspects such as precision, speed, and robustness, we repeated the experiment with 200 sets of starting values for each model, assessing the estimation time and convergence behaviour. This testing approach ensured that our conclusions were not dependent on specific initial conditions but were generalisable across a diverse range of starting values. Detailed experimental settings are provided in

Table 4.

The LC model's coefficients ($\beta_{tt}, \beta_{tc}, \beta_{hw}, \beta_{ch}$) and the specific alternative constant (ASC) were sampled from a range of [-2,2], reflecting the potential heterogeneity of traveller preferences. For the MIXL models with normal density, mean coefficients ($\mu_{tt}, \mu_{tc}, \mu_{hw}, \mu_{ch}$), covariance coefficients ($\sigma_{tt}, \sigma_{tc}, \sigma_{hw}, \sigma_{ch}$) and ASC were randomly drawn from [-1,1]. In contrast, the Mixed Logit models with lognormal density drew μ values from [-6,0] and σ values from [-0.5,0.5], acknowledging the distributional characteristics of lognormal and normal distributions. This variability is intended to provide a comprehensive understanding of the models' behaviour under different conditions.

In our framework, we implemented Analytical Differentiation (AnaD), Numerical Differentiation (ND), and Automatic Differentiation (AD) for the LC model. Our primary objective with AnaD was determining whether AD could achieve a comparable level of accuracy, as suggested theoretically. After this was confirmed for LC (see results below), we decided not to pursue the complex coding of AnaD for MIXL models. Instead and for completeness, we turned to the Apollo package to compare the precision of gradients obtained through AD and ND. In the comparison of speed and stability, our analysis encompassed a comprehensive threefold comparison

(AD, ND, AnaD) for the LC model and a twofold comparison (AD, ND) for the MIXL model, ensuring a fair and thorough evaluation within the CG framework.

5 Results

This section presents the empirical results for our wide range of estimation settings. The results show that Computational Graph-Automatic Differentiation (CG-AD) exhibits good performance in three primary dimensions: gradient precision, estimation speed, and optimisation stability.

5.1 Gradient precision

The primary objective of this section is to validate that the gradient precision of CG-AD closely aligns with that of analytical differentiation (AnaD), while numerical differentiation (ND) may not achieve such high levels of precision. Table 5 therefore provides a detailed comparison of derivative precision for different models. The reported 'starting value' is, for illustrative purposes, randomly chosen from the 200 starting values used in the following section. These starting values include both an all-zero parameter vector and center of randomly drawn starting values in the MIXL-lognormal model, as shown in Table 4. Notably, across all tested starting values in this section, the three gradient methods consistently yield similar experimental results, demonstrating a good model fit. Therefore, in Table 11, we present the log-likelihood, parameter estimates, standard errors, and t-ratios for CG-AD, further confirming the precision and effectiveness of CG-AD estimation.

As Table 5 shows, the gradients computed by CG-AD are nearly exactly replicating those obtained by AnaD. Specifically, the CG-AD results for the three models align with the AnaD up to the 10th or 11th decimal, indicating a very high level of gradient precision for CG-AD[1]. This level of accuracy is consistently observed across all parameters of the model. In sharp contrast, the ND already displays differences at the 2nd or 3rd decimal places for some parameters. Take μ_{tt} in the Mixed Logit (MIXL) model with normal density for example, the gradient computed via CG-AD and AnaD is 427.6213, while for ND it is 427.6204. The level of precision observed for CG-AD, as opposed to ND, ensures the true gradient landscape is closely mirrored, potentially increasing the probability of accurate estimation.

5.2 Estimation speed

This section highlights the advantages of CG in accelerating the estimation process. The distribution of estimation times across the 200 experiments is presented in the box plots in Figure 2. Table 6 shows that the average estimation time for the Latent Class (LC) model using AD is approximately 4.93 seconds, while ND takes about 20.53 seconds on average. For the MIXL model with normal density, AD requires approximately 1,059 seconds, whereas ND takes a significantly longer 12,481 seconds. For the MIXL model with lognormal density, the estimation time for AD is around 693 seconds, compared to 8,579 seconds for ND. Overall, the AD method reduces the time usage for the LC model to about half of that needed by the ND method. For the more complex mixed logit methods employing 5,000 Halton draws, the estimation time decreases by over tenfold. Our study emphasizes the superior computational performance of the CG-AD method over ND-based techniques, particularly evident when conducting Monte Carlo simulations for model estimation.

[1] Interested readers can visit our GitHub page to run the program and check the level of precision: https://github.com/lyliuyanly/CG-for-CM. The Excel dataset serves as an extended version of Table 4, allowing verification of precision beyond 12 significant digits.

5.3 Log-likelihood convergence

Besides exploring CG's potential for the precise and efficient estimation of DCMs, we are also interested in model convergence. Across the 200 starting values, we interpret the estimates with the highest log-likelihood as the best local optimum but cannot guarantee this is the global optimum. Our primary interest is in the extent to which different gradient calculation methods can consistently arrive at the 'best' log-likelihood value across the range of starting values. Figure 3 presents an overview of the frequency at which our different gradient calculation methods arrive at given log-likelihood values for respectively the LC model, the MIXL model with normal density, and the MIXL model with lognormal density.

As shown in Figure 3, finding the best local optimum is challenging. For the LC model, CG-AD and AnaD are only able to reach the best local optimum in a few more instances than ND, and the number of times the best local optimum is achieved is only 12% of the time for CG-AD and 11% for ND. In this case, we don't see a substantial benefit of using AnaD, AD, or ND-based derivatives and can only re-iterate that the speed benefit of LC allows testing across multiple starting values to circumvent the issue.

For the MIXL model, for a large enough number of draws, the final log-likelihood values in many cases are very close to the best one. After contrasting the parameter estimates in these locations, we can interpret these as arriving at the same optimum. The majority of mixed logit models therefore reach the best local optimum (around -1464 for MIXL-normal and around -1445 for MIXL-lognormal). However, we still observe instances of non-convergence and convergence to bad local optima. This is particularly the case for the MIXL model with lognormal densities where this applies to nearly 30% of instances. Notably, for the LC model, we find that the parameter estimates of the top four identified local optima differ significantly from each other, highlighting the challenges in achieving stable estimates.

Since the focus in this section is on the CG-based approach's ability to improve estimate robustness and reduce the likelihood of non-convergence and poor local optima, we employed a Chi-square test for all models, as presented in Tables 7–10, to provide a comprehensive assessment. Among them, the MIXL-lognormal model exhibits statistically significant results. Using the final log-likelihood of the Multinomial Logit (MNL) model as a benchmark (-1,666), the final LL values for the MIXL-lognormal model are categorised into three groups: 1) *Worse than MNL*: non-convergence or $LL \in (-\infty, -1{,}666)$; 2) Better than MNL but significantly worse than the best local optimum and therefore can be viewed as *undesirable local optima*: $LL \in [-1{,}666, -1{,}445)$; 3) *Close to the best local optimum*: LL around -1,445. We test the null hypothesis (H0) that under CG-AD there are no significantly different occurrences of undesirable local optima that worse than the MNL or non-convergence relative to ND. Based on our analysis (see Table 10), the Chi-square statistic is calculated to be 5.82 with 2 degrees of freedom, resulting in a p-value less than 0.1. This p-value allows us to reject the H0. Thus, we conclude that the CG-AD notably enhances the stability of the MIXL-lognormal model by mitigating the occurrences of landing in undesirable local optima and reducing instances of non-convergence. For the LC and MIXL-normal models reported in Table 8 and Table 9, they do not show strong statistical significance in the Chi-square test. For the general trends observed across all models in Table 7, the Chi-square statistic is 6.69 with 2 degrees of freedom ($p < 0.1$), indicating a significant difference. These findings suggest that CG-AD enhances model estimation stability by reducing the likelihood of undesirable local optima and non-convergence, with its impact varying across model specifications and being particularly notable in the MIXL-lognormal model.

Let us further analyse the reasons for the instability: the lack of convergence and tendency to achieve poor local optima may be related to the complexity of calculations involving the lognormal distribution, which can encounter numerical issues. According to the Apollo manual (Hess and Palma, 2019), ND may be susceptible to generating zero probabilities due to specific calculations in the process, particularly in complex models. This can impact the precision and reliability of estimation results. Although our CG approach is not immune to zero probabilities and numerical issues, our findings suggest that the CG approach with AD reduces the likelihood of falling into local optima, thus providing a more accurate and reliable choice modelling optimisation mechanism. While convergence may be achieved more frequently with AD, optimisation remains a pressing and unresolved challenge that should be treated with prudence.

6 Discussion

We have explored the potential advantages of Computational Graph-Automatic Differentiation (CG-AD) in three key areas: gradient precision, estimation speed and stability. Building on the results presented in the previous section, we now discuss how these results can help choice modelling researchers in developing their models more efficiently.

As previously mentioned, the local optima problem cannot be avoided except for the Multinomial Logit (MNL) model, which has a convex likelihood function. Despite the high precision of CG-AD’s gradient approximation, local optima and numerical issues are still inevitable, particularly with poor starting values. Some additional investigation shows that numerical issues related to starting values often arise early in the optimisation process for the Latent Class (LC) model. However, when these issues emerge later, they are typically caused by the optimiser passing through a flat or complex region with local optima, where even small differences in the gradient can lead the optimiser along a different path.

Similar findings can be observed with the Mixed Logit (MIXL) model. When dealing with a large number of parameters, the simulated likelihood function often contains multiple local optima in a high-dimensional solution space (Chen et al., 2010). Both Monte Carlo and Halton numerical integration procedures introduce inherent "noise" that depends on the number of draws taken and the sequence in which the draws are generated (Palma et al., 2020). However, increasing the number of draws (whether Halton or random) reduces the simulation noise and improves integration precision. Hess et al. (2007) and Hess et al. (2006) tested multiple runs using pseudo-random draws and concluded that 2,000 draws are sufficient to guarantee stable estimation results and mitigate the impact of local optima. To reduce such noise, we used 5,000 draws in our experiments and found that this mostly led to the best local optima. Nevertheless, even with such high gradient precision provided by CG-AD, local optima could not be avoided solely by increasing the number of draws due to the existence of bad numerical issues.

The best and most straightforward way to mitigate the local optima issue is to test multiple starting values (Hess and Palma, 2019). Recognising that this testing approach can be computationally intensive, Bierlaire et al. (2010) proposed an algorithm that can dynamically filter out unpromising candidates, thereby reducing the computational cost of the testing process.

However, testing a very large number of starting values is challenging for some sophisticated models and large datasets. In such a situation, analytical differentiation (AnaD) can be hard to calculate and numerical differentiation (ND) is time-consuming. CG-AD significantly accelerates the gradient computation and model calibration process. For example, the average estimation time for MIXL with normal density is 12,481 seconds (using 5,000 Halton draws) using ND, while only 1,058 seconds using CG-AD. Furthermore, CG-AD's results closely align with those obtained from the AnaD method and ensure a high degree of precision. Although not entirely accurate, its efficiency is particularly beneficial for the analyst to test large sets of starting values quickly and therefore serves as a substitution for AnaD, which is difficult to calculate and prone to human errors. Apart from enhancing practical applicability for complex scenarios and data-intensive fields, a quicker estimation process also signifies a reduction in computational resources (CPU time, memory, etc.), which is particularly beneficial when resources are limited. Therefore, the speed of the CG-AD allows advanced choice modelling practitioners and researchers to refine, modify, and iterate their models more efficiently, enhancing the overall process of model development and adjustment.

7 Conclusion

The growing complexity of choice models and the context of big data present challenges to the speed, accuracy, and stability of the parameter estimation process.

Building on the understanding that both choice modelling and classification/regression problems are non-convex optimisation processes aimed at finding the optimal parameter, we aim to leverage machine learning (ML)-based Computational Graph (CG) techniques to address classical choice modelling problems.

Drawing from previous work by Kim et al. (2021), we developed Latent Class (LC) and Mixed Logit (MIXL) models for travel mode choice using an open-access dataset, and further illustrated the core advantages of CG both theoretically and experimentally. Using a consistent compilation environment, coding language, and optimiser, we ensured a fair comparison of the properties of Automatic Differentiation (AD) and Numerical Differentiation (ND) methods.

Our experiments provide compelling evidence and practical insights into the CG framework by extensively testing various Discrete Choice Models (DCMs), demonstrating their estimation efficiency, accuracy, and stability in econometric applications. The results demonstrate that CG, underpinned by its unique AD mechanism, significantly accelerates the parameter calibration process. Specifically, we observed more than a tenfold reduction in estimation time for mixed logit methods employing 5,000 Halton draws. Beyond efficiency, we also observed CG's stability potential. When adjusting starting values and comparing the distribution of log-likelihoods and estimates, we found that CG is more likely to reach the best local optimum and less prone to stagnation at suboptimal solutions for the MIXL models. Consequently, integrating CG into DCMs enables more efficient model runs, simplifying testing and validation for researchers.

Based on the results, we conclude that CG-AD is a valuable tool for choice modellers. This powerful estimation technique not only promises advancements in econometric modelling and innovative applications but also underscores the need for continued research and refinement. Expanding validation across diverse datasets will be a crucial step in further strengthening its reliability and real-world applicability. While we have shown that AD can improve estimation stability, ensuring a guaranteed convergence to the global optimum in DCMs remains a significant challenge. Moving forward, we aim to fully leverage CG-AD as an estimation tool to enhance the calibration of discrete choice models (DCMs). Beyond traditional BFGS methods, CG-AD enables efficient derivative computation and can be seamlessly integrated with various optimization techniques within the TensorFlow framework, such as stochastic gradient descent (SGD) and Adaptive Moment Estimation (Adam). Additionally, regularization techniques (e.g., dropout, L2) and normalization methods (e.g., batch normalization) can help smooth the negative log-likelihood loss landscape and mitigate the risk of extreme parameter estimates leading to local optima. Furthermore, diverse initialization strategies—including heuristic algorithms (e.g., simulated annealing, genetic algorithms) and reinforcement learning—could enhance optimization robustness and accelerate convergence.

Beyond estimation improvements, broader ML techniques offer promising avenues for advancing choice modeling. The TensorFlow platform provides CG-AD with greater flexibility to integrate various ML architectures, such as deep neural networks (DNNs), attention mechanisms, and Transformers, to better capture nonlinear relationships in utility functions, taste coefficients, and latent variables, thereby improving model generalization. While these hybrid models introduce additional complexity, key economic indicators such as elasticity and willingness to pay (WTP) can still be efficiently derived using the derivatives obtained through automatic differentiation. Although CG-AD does not directly enhance model interpretability, it facilitates the computation of economic indicators, thereby preserving transparency. In this light, CG-AD acts as a bridge between classical theory and modern ML, preserving the core strengths that make DCMs valuable for transport applications—namely, theoretical consistency and economic insight—while adding flexibility and scalability. By leveraging these capabilities, future research can further integrate ML-driven techniques into classical frameworks, balancing predictive performance, computational efficiency, and behavioral transparency.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant U2333218; International Joint Doctoral Education Fund of Beihang University.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., 2016. Tensorflow: A system for large-scale machine learning. *In 12th symposium on operating systems design and implementation(16)*, 265-283.

Ali, A., Kalatian, A., Choudhury, C.F., 2023. Comparing and contrasting choice model and machine learning techniques in the context of vehicle ownership decisions. *Transportation Research Part A: Policy and Practice* 173.

Ali, M., 2024. Discrete Choice Models and Artificial Intelligence Techniques for Predicting the Determinants of Transport Mode Choice—A Systematic Review. *Computers, Materials & Continua* 81.

Arteaga, C., Park, J., Beeramoole, P.B., Paz, A., 2022. xlogit: An open-source Python package for GPU-accelerated estimation of Mixed Logit models. *Journal of Choice Modelling* 42.

Axhausen, K.W., Hess, S., König, A., Abay, G., Bates, J., Bierlaire, M., 2006. State of the art estimates of the Swiss value of travel time savings. *Arbeitsberichte Verkehrs-und Raumplanung* 383.

Bansal, P., Daziano, R.A., Guerra, E., 2018. Minorization-Maximization (MM) algorithms for semiparametric logit models: Bottlenecks, extensions, and comparisons. *Transportation Research Part B: Methodological* 115, 17-40.

Baydin, A.G., Pearlmutter, B.A., Radul, A.A., Siskind, J.M., 2018. Automatic differentiation in machine learning: A survey. *Journal of Machine Learning Research* 18, 1-43.

Bierlaire, M., Thémans, M., Zufferey, N., 2010. A heuristic for nonlinear global optimization. *INFORMS Journal on Computing* 22, 59-70.

Chen, S., Zhang, Y., Zhang, X., Jiao, J., 2010. A dynamic differential evolution algorithm for mixed logit discrete choice model estimation, *2010 IEEE International Conference on Industrial Engineering and Engineering Management*. IEEE, pp. 33-37.

Dow, J.K., Endersby, J.W., 2004. Multinomial probit and multinomial logit: a comparison of choice models for voting research. *Electoral studies* 23, 107-122.

Dürr, O., Sick, B., Murina, E., 2020. *Probabilistic deep learning: With python, keras and tensorflow probability*. Manning.

García-García, J.C., García-Ródenas, R., López-Gómez, J.A., Martín-Baos, J.Á., 2022. A comparative study of machine learning, deep neural networks and random utility maximization models for travel mode choice modelling. *Transportation Research Procedia* 62, 374-382.

Gebremedhin, A.H., Walther, A., 2019. An introduction to algorithmic differentiation. *WIREs Data Mining and Knowledge Discovery* 10.

Górriz, J.M., Álvarez-Illán, I., Álvarez-Marquina, A., Arco, J.E., Atzmueller, M., Ballarini, F., Barakova, E., Bologna, G., Bonomini, P., Castellanos-Dominguez, G., 2023. Computational approaches to explainable artificial intelligence: advances in theory, applications and trends. *Information Fusion* 100, 101945.

Griewank, A., Walther, A., 2008. *Evaluating derivatives: principles and techniques of algorithmic differentiation*. SIAM.

Guarda, P., Battifarano, M., Qian, S., 2024. Estimating network flow and travel behavior using day-to-day system-level data: A computational graph approach. *Transportation Research Part C: Emerging Technologies* 158, 104409.

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.

Heckman, J., Singer, B., 1984. A method for minimizing the impact of distributional assumptions in econometric models for duration data. *Econometrica: Journal of the Econometric Society*, 271-320.

Hensher, D.A., Greene, W.H.J.T., 2003. The mixed logit model: the state of practice. 30, 133-176.

Hess, S., Bierlaire, M., Polak, J.W., 2007. A systematic comparison of continuous and discrete mixture models. *European Transport \ Trasporti Europei*, 35-61.

Hess, S., Palma, D., 2019. Apollo: A flexible, powerful and customisable freeware package for choice model estimation and application. *Journal of Choice Modelling* 32.

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., Bierlaire, M., Elshafie, M.Z., Jin, Y., 2021. A systematic review of machine learning classification methodologies for modelling passenger mode choice. *Journal of choice modelling* 38, 100221.

Jerrell, M.E., 1997. Automatic differentiation and interval arithmetic for estimation of disequilibrium models. *Computational Economics* 10, 295-316.

Jiang, J., Anderson, C.K., 2024. Discrete Choice Models: Model Selection and Challenges in Applications. Available at SSRN 4810909.

Kamal, K., Farooq, B., 2024. Ordinal-ResLogit: Interpretable deep residual neural networks for ordered choices. *Journal of choice modelling* 50, 100454.

Kim, T., Zhou, X., Pendyala, R.M., 2021. Computational graph-based framework for integrating econometric models and machine learning algorithms in emerging data-driven analytical environments. *Transportmetrica A: Transport Science*, 1-30.

Lancsar, E., Fiebig, D.G., Hole, A.R., 2017. Discrete Choice Experiments: A Guide to Model Specification, Estimation and Software. *Pharmacoeconomics* 35, 697-716.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. *nature* 521, 436-444.

Lederrey, G., Lurkin, V., Hillel, T., Bierlaire, M., 2021. Estimation of discrete choice models with hybrid stochastic adaptive batch size algorithms. *Journal of Choice Modelling* 38.

Liu, H.X., Recker, W., Chen, A., 2004. Uncovering the contribution of travel time reliability to dynamic route choice using real-time loop data. *Transportation Research Part A: Policy and Practice* 38, 435-453.

Liu, Z., Yin, Y., Bai, F., Grimm, D.K., 2023. End-to-end learning of user equilibrium with implicit neural networks. *Transportation Research Part C: Emerging Technologies* 150.

Lu, J., Li, C., Wu, X.B., Zhou, X.S., 2023. Physics-informed neural networks for integrated traffic state and queue profile estimation: A differentiable programming approach on layered computational graphs. *Transportation Research Part C: Emerging Technologies* 153.

Łukawska, M., Jensen, A.F., Rodrigues, F., 2025. Context-aware Bayesian mixed multinomial logit model. *Journal of Choice Modelling* 54.

Ma, J., Ye, X., Huang, K., Miwa, T., 2022. Development of integrated choice and latent variable (ICLV) models using matrix-based analytic approximation and automatic differentiation methods on TensorFlow platform. *Journal of Advanced Transportation* 2022, 1-19.

Ma, W., Pi, X., Qian, S., 2020. Estimating multi-class dynamic origin-destination demand through a forward-backward algorithm on computational graphs. *Transportation Research Part C: Emerging Technologies* 119.

Mai, T., Fosgerau, M., Frejinger, E., 2015. A nested recursive logit model for route choice analysis. *Transportation Research Part B: Methodological* 75, 100-112.

Margossian, C.C., 2019. A review of automatic differentiation and its efficient implementation. *Wiley interdisciplinary reviews: data mining and knowledge discovery* 9, 1305.

Martín-Baos, J.Á., García-Ródenas, R., García, M.L.L., Rodriguez-Benitez, L., 2024. PyKernelLogit: Penalised maximum likelihood estimation of Kernel Logistic Regression in Python. *Software Impacts* 19, 100608.

Martín-Baos, J.Á., García-Ródenas, R., Rodriguez-Benitez, L., 2021. A Python package for performing penalized maximum likelihood estimation of conditional logit models using Kernel Logistic Regression. *Transportation Research Procedia* 58, 61-68.

Martín-Baos, J.Á., García-Ródenas, R., Rodriguez-Benitez, L., 2023. Optimization techniques for Kernel Logistic Regression on large-scale datasets: A comparative study, *5th International Conference on Advanced Research Methods and Analytics (CARMA 2023)*. Editorial Universitat Politècnica de València, pp. 239-240.

McFadden, D., 2022. Instability in mixed logit demand models. *Journal of Choice Modelling* 43.

McFadden, D., Train, K., 2000. Mixed MNL models for discrete response. *Journal of Applied Econometrics* 15, 447-470.

Nam, D., Kim, H., Cho, J., Jayakrishnan, R., 2017. A model based on deep learning for predicting travel mode choice, *Proceedings of the transportation research board 96th annual meeting transportation research board, Washington, DC, USA*, pp. 8-12.

Oliver, J., 1980. An algorithm for numerical differentiation of a function of one real variable. *Journal of Computational and Applied Mathematics* 6, 145-160.

Omrani, H., 2015. Predicting travel mode of individuals by machine learning. *Transportation research procedia* 10, 840-849.

Pacheco Paneque, M., Bierlaire, M., Gendron, B., Sharif Azadeh, S., 2021. Integrating advanced discrete choice models in mixed integer linear optimization. *Transportation Research Part B: Methodological* 146, 26-49.

Pál, L., Sándor, Z., 2023. Comparing procedures for estimating random coefficient logit demand models with a special focus on obtaining global optima. *International Journal of Industrial Organization* 88, 102950.

Palma, M.A., Vedenov, D.V., Bessler, D., 2020. The order of variables, simulation noise, and accuracy of mixed logit estimates. *Empirical Economics* 58, 2049-2083.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A., 2017. *Automatic differentiation in pytorch*.

Phan, D.T., Vu, H.L., Currie, G., 2022. Attentionchoice: Discrete choice modelling supported by a deep learning attention mechanism *Available at SSRN 4305637*.

Rodrigues, F., Ortelli, N., Bierlaire, M., Pereira, F.C., 2020. Bayesian automatic relevance determination for utility function specification in discrete choice models. *IEEE Transactions on Intelligent Transportation Systems* 23, 3126-3136.

Salas, P., De la Fuente, R., Astroza, S., Carrasco, J.A., 2022. A systematic comparative evaluation of machine learning classifiers and discrete choice models for travel mode choice in the presence of response heterogeneity. *Expert Systems with Applications* 193, 116253.

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

Smith, G.D., 1985. *Numerical solution of partial differential equations: finite difference methods*. Oxford university press.

Swait, J., 1994. A structural equation model of latent segmentation and product choice for cross-sectional revealed preference choice data. *Journal of retailing and consumer services* 1, 77-89.

Tjoa, E., Guan, C., 2021. A Survey on Explainable Artificial Intelligence (XAI): Toward Medical XAI. *IEEE Trans Neural Netw Learn Syst* 32, 4793-4813.

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

Vabalas, A., Gowen, E., Poliakoff, E., Casson, A.J., 2019. Machine learning algorithm validation with a limited sample size. *PLoS One* 14, e0224365.

van Cranenburgh, S., Wang, S., Vij, A., Pereira, F., Walker, J., 2022. Choice modelling in the age of machine learning - Discussion paper. *Journal of Choice Modelling* 42.

Verma, A., 2000. An introduction to automatic differentiation. *Current Science*, 804-807.

Vermeulen, B., Goos, P., Vandebroek, M., 2008. Models and optimal designs for conjoint choice experiments including a no-choice option. *International Journal of Research in Marketing* 25, 94-103.

Vilone, G., Longo, L., 2021. Notions of explainability and evaluation approaches for explainable artificial intelligence. *Information Fusion* 76, 89-106.

Wang, S., Mo, B., Zhao, J., 2021a. Theory-based residual neural networks: A synergy of discrete choice models and deep neural networks. *Transportation Research Part B: Methodological* 146, 333-358.

Wang, S., Wang, Q., Bailey, N., Zhao, J., 2021b. Deep neural networks for choice analysis: A statistical learning theory perspective. *Transportation Research Part B: Methodological* 148, 60-81.

Wang, S., Wang, Q., Zhao, J., 2020. Deep neural networks for choice analysis: Extracting complete economic information for interpretation. *Transportation Research Part C: Emerging Technologies* 118.

Wang, Z., He, S.Y., Leung, Y., 2018. Applying mobile phone data to travel behaviour research: A literature review. *Travel Behaviour and Society* 11, 141-155.

Welch, T.F., Widita, A., 2019. Big data in public transportation: a review of sources and methods. *Transport Reviews* 39, 795-818.

Wong, M., Farooq, B., 2021. ResLogit: A residual neural network logit model for data-driven choice modelling. *Transportation Research Part C: Emerging Technologies* 126.

Wu, X., Guo, J., Xian, K., Zhou, X., 2018. Hierarchical travel demand estimation using multiple data sources: A forward and backward propagation algorithmic framework on a layered computational graph. *Transportation Research Part C: Emerging Technologies* 96, 321-346.

Table List

Table 1. Notation table

Symbol	Description of Dimensions
I	Number of alternatives in the choice set
T	Number of choice occasions
Q	Number of decision-makers
J	Number of latent classes in the LC model
R	Number of replications for Monte Carlo simulation
Symbol	**Description of Indices**
i	Index for alternatives
t	Index for choice occasions
q	Index for individuals
j	Index for latent classes
Symbol	**Description of Utility Function Components**
U_{itq}	Utility of alternative i for individual q at choice occasion t
$U_{itq\|j}$	For class j, utility of alternative i for individual q at choice occasion t
U_{qj}	Utility of individual q belonging to class j
V_{itq}	Observed component of U_{itq}
$V_{itq\|j}$	Observed component of $U_{itq\|j}$
V_{qj}	Observed component of U_{qj}
ε_{itq}	Random error term in U_{itq} following an i.i.d. Gumbel distribution
$\varepsilon_{itq\|j}$	For class j, random error term in U_{itq} following an i.i.d. Gumbel distribution
ζ_{qj}	Random error term in class membership model
$\boldsymbol{X}_{\boldsymbol{itq}}$	Vector of explanatory variables for alternative i, individual q, and occasion t
$\boldsymbol{S}_q$	Vector of explanatory variables of individual q for class allocation
$\boldsymbol{\beta}$	Vector of model parameters
$\boldsymbol{\beta}_j$	Vector of model parameters for the latent class j
$\boldsymbol{\beta}_r$	Vector of model parameters for the Halton draw r
$\boldsymbol{\gamma}_j$	Vector of model parameters for class membership model in LC model
Symbol	**Description of Choice Variables**
y_{itq}	Choice variable of the q^{th} traveller, 1 if alternative i is chosen at occasion t, 0 otherwise
P_{y_q}	Joint probability of observed choices for individual q
$P_{y_{itq}=1}$	Probability that individual q chooses alternative i at occasion t
$P_{c_{qj}=1}$	Probability that individual q belongs to class j
$L_{itq}(\beta_r)$	Logit probability expression in MIXL model for Halton draw β_r
$P_{y_{itq}=1\|j}$	For class j, conditional probability that individual q chooses alternative i at occasion t
$SP_{y_{itq}=1}$	Simulated probability in MIXL model
$f(\beta)$	Density function of parameter distribution in MIXL model
$LL(y)$	Log-likelihood function
Symbol	**Description of Gradient and Numerical Computation**
∇f	Gradient of function f, which can be the log-likelihood function

h	Step size in numerical differentiation
$\mathbf{e}_v$	Unit vector in direction v
$\frac{\partial f}{\partial \mathbf{x}}$	Partial derivative of function f with respect to point $\mathbf{x}$
$\frac{\partial LL}{\partial \boldsymbol{\beta}}$	Gradient of the log-likelihood function with respect to parameter vector $\boldsymbol{\beta}$

Table 2. Summary of techniques to calculate derivatives

Technique	Advantages	Disadvantages
Analytical Differentiation (AnaD)	Exact and fast	Time-consuming to code, human error-prone, and difficult for complicated functions.
Numerical Differentiation (ND)	Easy to code and implement	Potential truncation/round-off errors, slow especially in high dimensions, as the method requires at least D evaluations, where D is the number of partial derivatives required.
Automatic differentiation (AD)	Exact, speed is comparable to or even quicker than hand-coding derivatives, highly applicable.	Needs to be carefully implemented, although this is already done in several packages including TensorFlow and PyTorch.

Table 3. Data dictionary for apollo_swissRouteChoiceData

Variable	Description	Values
ID	Unique respondent ID	Min: 1, max: 84525
choice	Public transport route choices	1 for route 1; 2 for route 2
tt1	Travel time of route 1	Min: 2, mean: 52.59, max: 389
tc1	Travel cost of route 1	Min: 1, mean: 19.67, max: 206
hw1	Headway of route 1	15; 30; 60
ch1	Interchange number of route 1	0; 1; 2
tt2	Travel time of route 2	Min: 2, mean: 52.47, max: 385
tc2	Travel cost of route 2	Min: 1, mean: 19.69, max: 268
hw2	Headway of route 2	15; 30; 60
ch2	Interchange number of route 2	0; 1; 2

hh_inc_abs	Household income	Min: 10000, mean: 76507.73, max: 167500
car_availability	Household car availability	1 for car available (37.89 %), 0 otherwise
commute	Purpose of travel is commute	1 for commute purpose (28.61%), 0 otherwise
shopping	Purpose of travel is shopping	1 for shopping purpose (8.24%), 0 otherwise
business	Purpose of travel is business	1 for business purpose (9.29%), 0 otherwise
leisure	Purpose of travel is leisure	1 for leisure purpose (53.87%), 0 otherwise
Individuals	388	
Observations	3492	

Table 4. Estimation Settings for Model Comparison

Model Type	**Latent Class Model**	**Mixed Logit- Normal Density**	**Mixed Logit- Lognormal Density**
Parameter Setting	Coefficients β_{tt}, β_{tc}, β_{hw}, β_{ch} of two latent class a/b, alternative route specific constant ASC_1	Mean coefficients μ_{tt}, μ_{tc}, μ_{hw}, μ_{ch} covariance coefficients σ_{tt}, σ_{tc}, σ_{hw}, σ_{ch}, alternative route specific constant ASC_1	Mean coefficients μ_{tt}, μ_{tc}, μ_{hw}, μ_{ch} covariance coefficients σ_{tt}, σ_{tc}, σ_{hw}, σ_{ch}
Gradient Method	AD, ND, AnaD	AD, ND, AnaD (in Apollo)	AD, ND, AnaD (in Apollo)
Starting Value	200 random draws, β, ASC in $[-2,2]$	200 random draws, μ, σ, ASC in $[-1,1]$	200 random draws, μ in $[-6,0]$, σ in $[-0.5,0.5]$

Table 5: Derivative comparison of different models

LC

Parameter	$\beta_{tt,a}$	$\beta_{tt,b}$	$\beta_{tc,a}$	$\beta_{tc,b}$	$\beta_{hw,a}$	$\beta_{hw,b}$	$\beta_{ch,a}$	$\beta_{ch,b}$	ASC_1	Δ_b
Value	0	0	0	0	0	0	0	0	0	0
CG-AD	1.9995E3	1.9995E3	1.1250E1	1.1250E1	7.5675E3	7.5675E3	4.5525E2	4.5525E2	1.2000E1	0
ND	1.9995E3	1.9994E3	1.1285E1	1.1273E1	7.5675E3	7.5671E3	4.5525E2	4.5525E2	1.2000E1	0
AnaD	1.9995E3	1.9995E3	1.1250E1	1.1250E1	7.5675E3	7.5675E3	4.5525E2	4.5525E2	1.2000E1	0
Value	-1.55181E-1	-3.47948E-1	1.75951	1.25873	-2.54204E-1	-1.47778	-1.34748	-2.18778E-1	-1.48403	-1.38234
CG-AD	-8.79764E3	-2.30280E3	3.02961E3	1.49242E3	-1.10860E4	2.68779E1	7.77773E2	7.26236E1	-1.60148E2	-2.07999E2
ND	-8.79764E3	-2.30281E3	3.02961E3	1.49242E3	-1.10860E4	2.68781E1	7.77773E2	7.26235E1	-1.60148E2	-2.07999E2
AnaD	-8.79764E3	-2.30280E3	3.02961E3	1.49242E3	-1.10860E4	2.68779E1	7.77773E2	7.26236E1	-1.60148E2	-2.07999E2

MIXL-normal

Parameter	μ_{tt}	μ_{tc}	μ_{hw}	μ_{ch}	σ_{tt}	σ_{tc}	σ_{hw}	σ_{ch}	ASC_1
Value	0	0	0	0	0	0	0	0	0
CG-AD	-3.9990E3	-2.2500E1	-1.5135E4	-9.1050E2	2.5245E-1	-9.5841E-1	-7.0502E-1	-2.5600E-1	-1.2000E1
ND	-3.9990E3	-2.9520E1	-1.5135E4	-9.1071E2	2.5098E-1	-9.5521E-1	-7.2384E-1	-2.5418E-1	-1.1984E1
AnaD	-3.9990E3	-2.2500E1	-1.5135E4	-9.1050E2	2.5245E-1	-9.5841E-1	-7.0502E-1	-2.5600E-1	-1.2000E1
Value	2.3430E-1	8.9392E-2	4.9030E-2	-5.9195E-1	-3.9652E-1	3.5190E-1	4.0572E-1	-6.5085E-1	-8.0343E-1
CG-AD	-4.2762E2	-3.1630E2	-5.7545E2	-3.0033E2	1.0991E2	1.0890E2	-4.5562E2	-1.7143E2	1.8898E2
ND	-4.2762E2	-3.1629E2	-5.7545E2	-3.0034E2	1.1041E2	1.0869E2	-4.5570E2	-1.7141E2	1.8898E2
AnaD	-4.2762E2	-3.1630E2	-5.7545E2	-3.0033E2	1.0991E2	1.0890E2	-4.5562E2	-1.7143E2	1.8898E2

MIXL-lognormal

Parameter	μ_{tt}	μ_{tc}	μ_{hw}	μ_{ch}	σ_{tt}	σ_{tc}	σ_{hw}	σ_{ch}
Value	0	0	0	0	0	0	0	0
CG-AD	-6.5905E3	2.1962E3	-1.8311E4	8.1497E2	3.1328E-1	1.6340	6.0495	-5.1259E-1
ND	-6.5905E3	2.1959E3	-1.8311E4	8.1496E2	3.1344E-1	1.7017	5.9796	-5.1128E-1
AnaD	-6.5905E3	2.1962E3	-1.8311E4	8.1497E2	3.1328E-1	1.6340	6.0495	-5.1259E-1
Value	-3	-3	-3	-3	-1E-2	-1E-2	-1E-2	-1E-2
CG-AD	-1.4925E2	6.9259E1	-3.9286E2	4.6522E1	-9.0437	-1.6925	-1.4227E1	-5.5218E-1
ND	-1.4925E2	6.9248E1	-3.9287E2	4.6522E1	-9.0411	-1.6920	-1.4222E1	-5.5211E-1
AnaD	-1.4925E2	6.9259E1	-3.9286E2	4.6522E1	-9.0437	-1.6925	-1.4227E1	-5.5218E-1
Value	-3.5231	-5.1831E-1	-5.4063	-3.2638	1.2180E-1	-4.3868E-1	-2.2918E-1	1.4683E-1
CG-AD	3.4643E2	-8.4733E2	5.9703E1	3.0486E1	1.0305E2	-1.3252E3	-1.7327E1	4.5958
ND	3.4643E2	-8.4735E2	5.9703E1	3.0486E1	1.0304E2	-1.3251E3	-1.7326E1	4.5955
AnaD	3.4643E2	-8.4733E2	5.9703E1	3.0486E1	1.0305E2	-1.3252E3	-1.7327E1	4.5958

Note: Some numbers are expressed in scientific notation (e.g., 1.23E4 = 1.23 × 10^4, 5.67E-8 = 5.67 × 10^{-8}).

Table 6. Estimation time (seconds) for AD and ND derivatives

	Index	**LC**	**MIXL-Normal**	**MIXL-Lognormal**
AD	Mean	4.93	1058.58	693.19
	Std	2.40	1124.77	599.10
	Medium	4.36	621.90	517.95
ND	Mean	20.53	12481.11	8579.01
	Std	10.71	5812.18	6039.07
	Medium	17.60	11566.84	7262.64

Table 7. Final LL Statistics for the three models

	non-convergence or (-inf,-1666)	**[-1666,best local optima)**	**best local optima**
AD	45	180	375
ND	71	180	349
Total	116	360	724

Note: $\chi^2 = 6.69, df = 2 \Longrightarrow p \approx 0.035$ (significant， $p < 0.1$)

Table 8. Final LL Statistics for LC

	non-convergence or (-inf,-1666)	**[-1666,best local optima)**	**best local optima**
AD	/	176	24
ND	/	178	22
Total	/	354	46

Note: $\chi^2 = 0.20, df = 1 \Longrightarrow p \approx 0.654$ (non-significant)

Table 9. Final LL Statistics for MIXL-normal

	non-convergence or (-inf,-1666)	**[-1666,best local optima)**	**best local optima**
AD	4	1	195
ND	9	0	191
Total	13	1	386

Note: $\chi^2 = 2.96, df = 2 \Longrightarrow p \approx 0.227$ (non-significant)

Table 10. Final LL Statistics for MIXL-lognormal

	non-convergence or (-inf,-1666)	**[-1666,best local optima)**	**best local optima**
AD	41	3	156
ND	62	2	136
Total	103	5	292

Note: $\chi^2 = 5.82, df = 2 \Longrightarrow p \approx 0.054$ (significant， $p < 0.1$)

Table 11. Parameter estimation for different models of CG-AD

LC

Parameter	$\beta_{tt,a}$	$\beta_{tt,b}$	$\beta_{tc,a}$	$\beta_{tc,b}$	$\beta_{hw,a}$	$\beta_{hw,b}$	$\beta_{ch,a}$	$\beta_{ch,b}$	ASC_1	Δ_b
Starting Value	0	0	0	0	0	0	0	0	0	0
Coef.	-7.3549E-2	-9.7726E-2	-9.5717E-2	-5.3342E-1	-3.9622E-2	-4.7482E-2	-7.6379E-1	-2.1676	-4.4836E-2	3.9177E-2
Std.err	4.3533E-4	7.1754E-4	8.3114E-4	4.7765E-3	1.9809E-4	2.8814E-4	5.3843E-3	9.3931E-3	2.4242E-3	1.3710E-2
t-ratio	-1.6895E2	-1.3619E2	-1.1516E2	-1.1168E2	-2.0003E2	-1.6479E2	-1.4186E2	-2.3076E2	-1.8495E1	2.8575
LL (initial) //LL (final)	-2420.469955//-1564.098668									
Starting Value	-1.55181E-1	-3.47948E-1	1.75951	1.25873	-2.54204E-1	-1.47778	-1.34748	-2.18778E-1	-1.48403	-1.38234
Coef.	-6.0076E-2	-1.6953	-1.3206E-1	-3.5591	-3.4771E-2	-2.6911	-1.1890	-1.3757	-1.5679E-2	2.9333
Std.err	2.2188E-4	4.3547E-2	6.9576E-4	9.3150E-2	9.7529E-5	7.3174E-2	2.3476E-3	3.9052E-2	2.2112E-3	1.7066E-2
t-ratio	-2.7076E2	-3.8930E1	-1.8980E2	-3.8209E1	-3.5652E2	-3.6777E1	-5.0646E2	-3.5228E1	-7.0907	1.7188E2
LL (initial) //LL (final)	-12105.27645//-1644.145721									

MIXL-normal

Parameter	μ_{tt}	μ_{tc}	μ_{hw}	μ_{ch}	σ_{tt}	σ_{tc}	σ_{hw}	σ_{ch}	ASC_1
Starting Value	0	0	0	0	0	0	0	0	0
Coef.	-1.4786E-1	-5.0308E-1	-6.5385E-2	-2.1353	6.7190E-2	-4.0232E-1	-4.1196E-2	1.3271	-4.9420E-2
Std.err	1.2940E-2	6.3793E-2	4.9086E-3	1.3534E-1	1.1080E-2	4.8440E-2	5.3598E-3	1.4096E-1	6.2576E-2
t-ratio	-1.1426E1	-7.8862	-1.3321E1	-1.5777E1	6.0642	-8.3056	-7.6862	9.4142	-7.8977E-1
LL (initial) //LL (final)	-2420.469955//-1464.147211								
Starting Value	2.3430E-1	8.9392E-2	4.9030E-2	-5.9195E-1	-3.9652E-1	3.5190E-1	4.0572E-1	-6.5085E-1	-8.0343E-1
Coef.	-1.4627E-1	-4.7425E-1	-6.5286E-2	-2.1605	-5.6939E-2	4.2932E-1	4.1014E-2	-1.2314	-5.1195E-2
Std.err	1.1293E-2	4.6599E-2	4.8246E-3	1.3460E-1	8.1581E-3	5.1951E-2	5.3728E-3	1.2305E-1	5.9710E-2
t-ratio	-1.2952E1	-1.0177E1	-1.3532E1	-1.6051E1	-6.9794	8.2640	7.6337	-1.0007E1	-8.5740E-1
LL (initial) //LL (final)	-2410.888953//-1463.67735								

MIXL-lognormal

Parameter	μ_{tt}	μ_{tc}	μ_{hw}	μ_{ch}	σ_{tt}	σ_{tc}	σ_{hw}	σ_{ch}
Starting Value	0	0	0	0	0	0	0	0
Coef.	-2.0163	-1.0268	-2.9272	6.3097E-1	-4.9016E-1	1.0496	-7.7476E-1	-8.4999E-1
Std.err	8.7437E-2	1.2932E-1	8.3340E-2	7.4054E-2	6.8734E-2	9.8757E-2	1.1921E-1	9.8623E-2
t-ratio	-2.3059E1	-7.9405	-3.5123E1	8.5204	-7.1313	1.0628E1	-6.4992	-8.6186
LL (initial) //LL (final)	-22106.1493//-1444.869248							
Starting Value	-3	-3	-3	-3	-1E-2	-1E-2	-1E-2	-1E-2
Coef.	-1.9920	-1.0311	-2.9347	6.2198E-1	-4.4940E-1	-1.0206	-8.3249E-1	-8.2274E-1
Std.err	8.6804E-2	1.3492E-1	8.5001E-2	7.2994E-2	6.8924E-2	9.0372E-2	1.1734E-1	1.0300E-1
t-ratio	-2.2948E1	-7.6421	-3.4526E1	8.5210	-6.5202	-1.1294E1	-7.0946	-7.9877
LL (initial) //LL (final)	-2253.782757//-1444.830455							
Starting Value	-3.5231	-5.1831E-1	-5.4063	-3.2638	1.2180E-1	-4.3868E-1	-2.2918E-1	1.4683E-1
Coef.	-1.9904	-1.0461	-2.9239	6.4083E-1	4.8288E-1	-1.0025	-8.4345E-1	8.2969E-1
Std.err	8.8225E-2	1.3985E-1	8.6196E-2	7.5123E-2	5.7391E-2	8.5501E-2	1.3251E-1	1.1466E-1
t-ratio	-2.2560E1	-7.4802	-3.3921E1	8.5304	8.4140	-1.1725E1	-6.3651	7.2363
LL (initial) //LL (final)	-2833.202652//-1444.39905							

Note: Some numbers are expressed in scientific notation (e.g., 1.23E4 = 1.23 × 10^4, 5.67E-8 = 5.67 × 10^{-8}).

Figure List

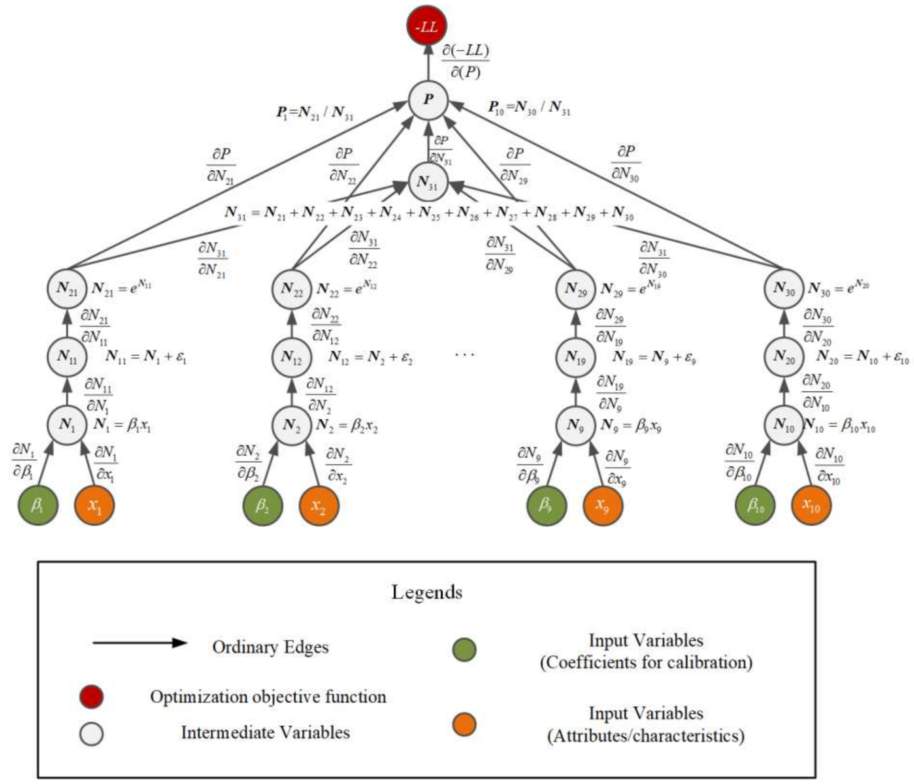
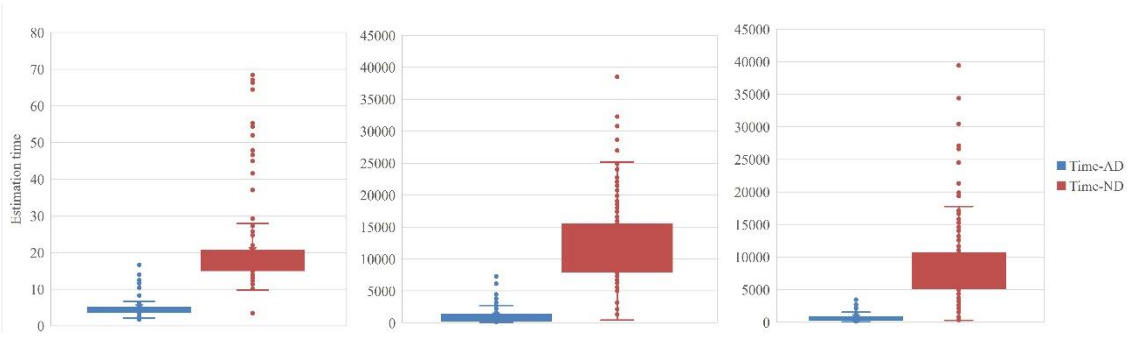
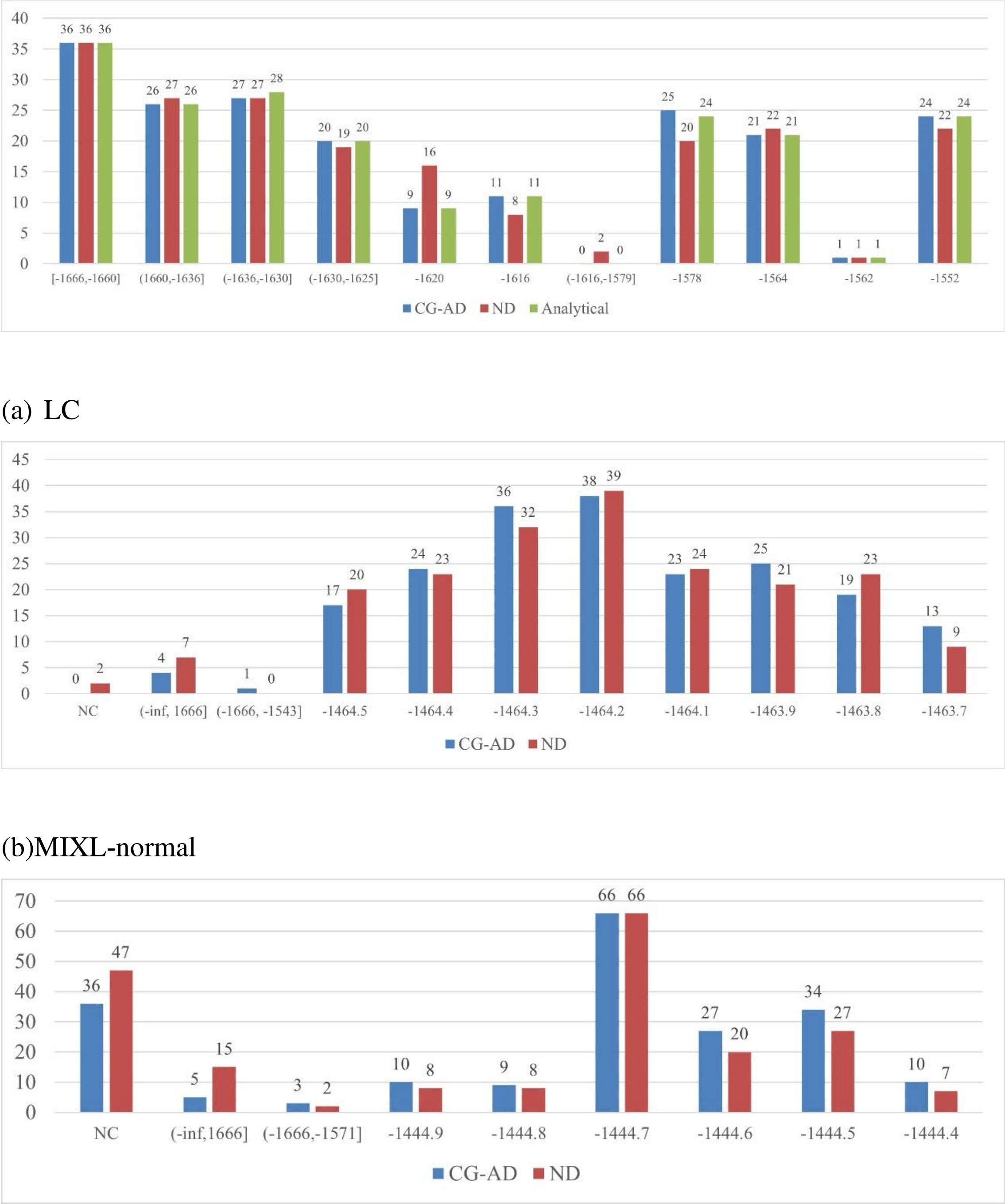


Figure 1: Computational graph (CG) structure of a simple multinomial logit (MNL) model



(a) LC (b) MIXL-normal (c) MIXL-lognormal

Figure 2: Estimation Time for Different Models



(a) LC

(b)MIXL-normal

(c) MIXL-lognormal

Figure 3: LL-final distribution for Different Models