Comparing hundreds of machine learning and discrete choice models for travel demand modeling: an empirical benchmark

Shenhao Wang[1,4,5*], Baichuan Mo[3], Yunhan Zheng[3,5], Stephane Hess[2], Jinhua Zhao[5]

1 - Department of Urban and Regional Planning, University of Florida
2 - Choice Modeling Centre & Institute for Transport Studies, University of Leeds
3 - Department of Civil and Environment Engineering, Massachusetts Institute of Technology
4 - Media Lab, Massachusetts Institute of Technology
5 - Department of Urban Studies and Planning, Massachusetts Institute of Technology

December 2023

Abstract

Numerous studies have compared machine learning (ML) and discrete choice models (DCMs) in predicting travel demand. However, these studies often lack generalizability as they compare models deterministically without considering contextual variations. To address this limitation, our study develops an empirical benchmark by designing a tournament model to learn the intrinsic predictive values of ML and DCMs. This novel approach enables us to efficiently summarize a large number of experiments, quantify the randomness in model comparisons, and use formal statistical tests to differentiate between the model and contextual effects. This benchmark study compares two large-scale data sources: a database compiled from literature review summarizing 136 experiments from 35 studies, and our own experiment data, encompassing a total of 6,970 experiments from 105 models and 12 model families, tested repeatedly on three datasets, sample sizes, and choice categories. This benchmark study yields two key findings. Firstly, many ML models, particularly the ensemble methods and deep learning, statistically outperform the DCM family and its individual variants (i.e., multinomial, nested, and mixed logit), thus corroborating with the previous research. However, this study also highlights the crucial role of the contextual factors (i.e., data sources, inputs and choice categories), which can explain models' predictive performance more effectively than the differences in model types alone. Model performance varies significantly with data sources, improving with larger sample sizes and lower dimensional alternative sets. After controlling all the model and contextual factors, significant randomness still remains, implying inherent uncertainty in such model comparisons. Overall, we suggest that future researchers shift more focus from context-specific and deterministic model comparisons towards examining model transferability across contexts and characterizing the inherent uncertainty in ML, thus creating more robust and generalizable next-generation travel demand models.

Keywords: Machine learning; choice modeling; travel behavior; prediction.

Corresponding author: Shenhao Wang; email: shenhaowang@ufl.edu

1. Introduction

In recent years, there has been a growing interest in incorporating machine learning (ML) models into travel demand analysis, alongside the traditional choice models (DCMs) that have been widely used for decades [4, 10, 64, 16]. These studies seek to compare the differences in model performance, system robustness, and ease of interpretation between ML and DCMs [28, 70, 12]. These comparative studies play a crucial role in generating valuable insights into novel behavioral mechanisms and diversifying travel demand models in practical applications [73, 72].

However, existing comparative studies suffer from limitations that hinder their robustness and generalizability. They often draw definitive conclusions about the superiority of ML models over DCMs based on a limited number of models and datasets that are specific to particular contexts. Such an approach overlooks the significant sources of randomness in model performance, resulting in potentially misleadingly high confidence in the superiority of ML models. What further adds to the limitations is an implicit assumption that model choice alone drives performance, while disregarding the impact of random contextual factors such as sample sizes and input/output dimensions. To address these shortcomings, it is essential to strive for comparative studies that provide generalizable insights and quantify the confidence in performance comparisons between ML models and DCMs. Unfortunately, past studies have not yet achieved these goals, underscoring the critical need for a benchmark research to overcome these limitations and provide a comprehensive and robust evaluation of the comparative performance of ML models and DCMs in travel demand modeling.

To address the limitations, we develop a large-scale empirical benchmark by designing a tournament model to compare the performance of hundreds of ML and DCMs. The tournament model uses the winning model in an experiment as the output, and alternative models and contextual factors as the inputs[1]. It captures the essence of comparative experiments by conducting pairwise comparison for the intrinsic predictive values of models using statistical methods. With a formal statistical approach, our benchmark study can efficiently summarize findings, differentiate between the model and contextual effects, and highlight the shared patterns from a large number of comparative experiments. With the tournament model, this benchmark study can filter out the context-specific noises and reveal common patterns in a specific topic, resembling a conventional meta-analysis while strengthening it with our own large-scale experiments [21, 35, 46]. As far as we know, such an empirical benchmark is still missing for travel demand modeling.

This benchmark study leverages two large-scale data sources: data compiled from literature review and from our own experiments. We documented choice categories, sample sizes, alternative models, best models, and many other contextual factors from 136 experiments in 35 past studies, constituting the literature data. This literature data facilitates the design of our 6,970 experiments and the analysis of the large-scale experiment data, in which we compared 105 models from 12 model families, repeatedly on 3 data sets, 3 sample sizes, and 3 outputs. The two data sets are collected, processed, and analyzed separately, but their results from the tournament models are

---

[1]*Inputs* and *outputs* are commonly used in the ML literature, and they correspond to *independent variables* and *dependent variables* in choice modeling.

compared in the result section. The literature and experiment data are juxtaposed because of their complementarity. The experiments prevail in the completeness of models, while the literature data is rich in contextual diversity. By comparing the literature and experiment data, this benchmark research can identify the consistent and inconsistent patterns. The patterns consistent between literature and experiment data could have more validity than the findings from only literature or experiment data, thus enhancing the generalizability of our empirical findings. The patterns inconsistent between literature and experiment data could reveal potential publication biases, arising from past authors' or journals' preferences for specific findings.

This paper is organized as follows. Section 2 introduces the design of the tournament model. Section 3 and Section 4 introduce the literature and experiment data, and discusses the model and contextual factors. Section 5 presents the empirical findings, starting with the results from the tournament models and further delving into each contributing factor to model performance. Section 6 presents limitations, and Section 7 concludes our findings and discuss future research directions. The acronyms are summarized in Appendix I to facilitate the reading of this study.

2.   Empirical benchmark with a tournament model

The tournament model compares M models, where model $m \in \{1, 2, ..., M\}$ has an intrinsic predictive value $\tilde{x}_m^*$. This intrinsic predictive value can be decomposed as $\tilde{x}_m^* = x_m^* + \epsilon_m$, in which x_m^* represents the deterministic component and ϵ_m the random component. The tournament model compares a pair of such models m_0 and m_1 with $m_0, m_1 \in \{1, 2, ..., M\}$. If the intrinsic predictive value of model m_0 is higher than model m_1, then it is more likely for model m_0 to win the comparison with the following probability:

$$P(m_0 \succ m_1) = \sigma(x_{m_0}^*, x_{m_1}^*) \tag{1}$$

in which $x_{m_0}^*$ and $x_{m_1}^*$ are the deterministic intrinsic predictive values of models m_0 and m_1, and σ takes the form of a link function. Under the assumption of extreme value distributions for ϵ_m, the link function takes the logistic form:

$$\sigma(x_{m_0}^*, x_{m_1}^*) = \frac{e^{x_{m_0}^*}}{e^{x_{m_0}^*} + e^{x_{m_1}^*}}, \tag{2}$$

guaranteeing that $\sigma(x_{m_0}^*, x_{m_1}^*) + \sigma(x_{m_1}^*, x_{m_0}^*) = 1$. Using i to indicate the comparison between model pairs, the output of each comparison is $y_i \in \{0, 1\}$ with 0 representing the winning of model m_0 and 1 representing the winning of model m_1, or vice versa. The winner of each model pair is determined by their prediction accuracy (i.e., hit rates), which allows broad comparison across deterministic (e.g. SVM) and probabilistic (e.g., neural networks) models. After observing a large number of such comparisons, we could infer x_m^* using statistical methods.

It is highly likely that the intrinsic predictive values of model m varies with contextual factors, such as sample size and choice categories. To incorporate the contextual factors, x_m^* can be

parameterized as:

$$x^*_{m_i} = \sum_{\tilde{\mathcal{M}} \in \mathcal{D}} \beta_{\tilde{\mathcal{M}}} \cdot \mathbb{1}_{m_i \in \tilde{\mathcal{M}}} + \boldsymbol{\gamma}' \boldsymbol{z}_{m_i} \tag{3}$$

in which $\mathbb{1}_{m_i \in \tilde{\mathcal{M}}}$ represents the indicators for whether the model m_i belongs to model family $\tilde{\mathcal{M}}$ with an associated parameter $\beta_{\tilde{\mathcal{M}}}$, and $\boldsymbol{z}_{m_i}$ represents the contextual factors of model m_i with an associated vector of parameters γ. The set of all model families is $\mathcal{D}$. We distinguish between model m_i and model family $\tilde{\mathcal{M}}$; for example, a deep neural network with a specific architecture is a model, while DNNs with various architectures belong to the same model family. In our experiments, we analyze 105 models ($M = 105$) and 12 model families ($|\mathcal{D}| = 12$). Since the number of model families is significantly smaller, it is more efficient to estimate the coefficients for model families rather than for individual models.

Equation 3 can be further enriched by specifying the interaction and nonlinear contextual effects, such as

$$x^*_{m_i} = \sum_{\tilde{\mathcal{M}} \in \mathcal{D}} \beta_{\tilde{\mathcal{M}}} \cdot \mathbb{1}_{m_i \in \tilde{\mathcal{M}}} + \boldsymbol{\gamma}'_{1,\tilde{\mathcal{M}}}(z_{m_i} \cdot \mathbb{1}_{m_i \in \tilde{\mathcal{M}}}) + \boldsymbol{\gamma}'_{2,\tilde{\mathcal{M}}}(z^2_{m_i} \cdot \mathbb{1}_{m_i \in \tilde{\mathcal{M}}}) \tag{4}$$

in which $\boldsymbol{\gamma}_{1,\tilde{\mathcal{M}}}$ and $\boldsymbol{\gamma}_{2,\tilde{\mathcal{M}}}$ capture the linear and quadratic contextual effects specific to model families. This equation is more general than Equation 3 because it releases the constraint of equal contextual effects across model families. To fully identify the parameters, our tournament models use discrete choice models (DCMs) as the baseline, so β's and γ's are interpreted as measuring the differences of intrinsic predictive values between ML models and DCMs.

This tournament model resembles a choice model with alternative sets varying across choice situations. The intrinsic predictive value could be seen as the "utility" of model m, and in every game (i.e., comparison), a better model m is chosen from a pair of alternatives m_0 and m_1, which are randomly sampled from a large model set $\{0, 1, ..., M\}$. Therefore, maximum likelihood estimation (MLE) could be adopted to estimate the parameters in the tournament model.

$$LL(g(\mathbb{1}_{m_i \in \tilde{\mathcal{M}}}, z; \beta, \gamma)) = \sum_{(m_j, m_k) \subset \{1,2,...M\}} W_{m_j \succ m_k} \log[\sigma(x^*_{m_j}, x^*_{m_k})] \tag{5}$$

in which $W_{m_j \succ m_k}$ represents the counts of model m_j outperforming model m_k in the re-sampled model pairs. Hence in the tournament model, the unit of analysis is a re-sampled model pair, rather than a single model. Within a model pair, the two models are compared by their prediction accuracy. Notably, the model pairs are sampled from either the literature or the experiment data, without any cross-comparison between the two data groups.

The following sections comprise two steps for utilizing the tournament model. Firstly, we gather information from individual experiments in the literature, documenting model performance (i.e., prediction accuracy) and corresponding model-family and contextual factors. Simultaneously, we conduct our own experiments, structured similarly to the literature data. In the second step, we randomly select pairs of models separately for the experiment and literature data to compare performance and record the winning model. The pairwise comparisons are then used in the tournament

models for subsequent statistical analysis.

3. Literature data

We compile the literature data by organizing the past studies along several dimensions as shown in Table 1. The dimensions are categorized into the treatment and control dimensions, corresponding to our analytical process in the result section. The treatment dimension includes the compared models (Column 5) and the best model (Column 6), and the control dimensions include the choice categories (Column 3), sample sizes (Column 4), among many other contextual factors, which are not presented in Table 1. The collection process and description of the literature data are incorporated in Appendix IV.

1. Study Index	2. Author (Year)	3. Choice Types	4. Sample Size	5. Number of Experiments - Compared Models	6. Best Model
1	Nijkamp et al. (1996) [50]	MC	1,396	2 - DNN, MNL	DNN
2	Rao et al. (1998) [56]	MC	4,335	2 - DNN, MNL	DNN
3	Hensher and Ton (2000) [32]	MC	1,500	2 - DNN, NL	DNN/NL
4	Sayed and Razavi (2000) [58]	MC	7,500	2 - MNL, DNN	DNN
5	Cantarella et al. (2002) [9]	MC	2,808	3 - DNN, MNL, NL	DNN
6	Mohammadian et al. (2002) [48]	CO	597	2 - DNN, NL	DNN
7	Doherty and Mohammadian (2003) [19]	TP	5,583	2 - DNN, GLM	DNN
8	Xie et al. (2003) [80]	MC	4,747	3 - DT, DNN, MNL	DNN
9	Cantarella et al. (2005) [10]	MC	1,067	2 - DNN, MNL	DNN
10	Celikoglu (2006) [11]	MC	381	4 - DNN, MNL	DNN
11	Tortum et al. [68]	MC	441	3 - MNL, GLM, DNN	DNN
12	Zhang and Xie (2008) [82]	MC	1,000	3 - MNL, DNN, SVM	SVM
12	Zhang and Xie (2008) [82]	MC	2,000	3 - MNL, DNN, SVM	SVM
13	Biagioni et al. (2008) [5]	MC	19,118	4 - MNL, BM, BOOSTING, DT	BOOSTING
14	Xian and Jian (2011) [79]	MC	4,725	3 - SVM, NL, DNN	SVM
15	Allahviranloo and Recker (2013) [3]	TP	3,671	2 - SVM, MNL	SVM
16	Omrani et al. (2013) [52]	MC	3,673	6 - DNN, SVM, KNN, MNL, DT, BM	DNN
17	Ermagun et al. (2015) [20]	MC	4,700	2 - NL, RF	RF
18	Jahangiri et al. (2015) [39]	MC	N.A.	6 - KNN, SVM, DT, BAGGING, RF, MNL	RF
19	Tang et al. (2015) [66]	MC	72,536	2 - DT, MNL	DT
20	Omrani (2015) [51]	MC	9,500	4 - DNN, RBFNN, MNL, SVM	DNN
21	Shafique et al. (2015) [60]	MC	1,968	4 - SVM, BOOSTING, DT, RF	RF
21	Shafique et al. (2015) [60]	MC	1,488	4 - SVM, BOOSTING, DT, RF	RF

21	Shafique et al. (2015) [60]	MC	$2,754$	4 - SVM, BOOSTING, DT, RF	RF
22	Shukla et al. (2015) [62]	MC	$100,000$	2 - DNN, DT	DNN
23	Sekhar and Madhu (2016) [59]	MC	$4,976$	3 - RF, DT, MNL	RF
24	Hagenauer and Helbich (2017) [28]	MC	$230,608$	8 - MNL, DNN, NB, SVM, CTs, BOOSTING, BAGGING, RF	RF
25	Paredes et al. (2017) [53]	CO	$15,211$	5 - MNL, BOOSTING, DT, SVM, RF	RF
26	Hillel et al. (2018) [34]	MC	N.A.	8 - DNN, BAGGING, BOOSTING, KNN, GLM, BM, RF, SVM	BOOSTING
27	Golshani et al. (2018) [26]	MC	$9,450$	2 - MNL, DNN	DNN
28	Tang et al. (2018)	MC	$14,000$	2 -MNL, DT	DT
29	Wang and Ross (2018) [70]	MC	$51,910$	2 - BOOSTING, MNL	BOOSTING
30	Lee et al. (2018)	MC	$4,764$	2 - MNL, DNN	DNN
31	Cheng et al. (2019) [12]	MC	$7,276$	4 - RF, SVM, BOOSTING, MNL	RF
32	Zhou et al. (2019) [83]	MC	$30,000$	8 - MNL, KNN, DT, SVM, BM, BOOSTING, BAGGING, RF	BAGGING
33	Wang et al. (2020) [72]	MC	$8,418$	8 - MNL, NL, DNN, SVM, BM, KNN, BOOSTING, GLM	DNN
33	Wang et al. (2020) [72]	MC	$2,929$	8 - MNL, NL, DNN, SVM, BM, KNN, BOOSTING, GLM	DNN
34	Wang et al. (2020) [73]	MC	$80,000$	2 - MNL, DNN	DNN
34	Wang et al. (2020) [73]	MC	$8,418$	2 - MNL, DNN	DNN
35	Wang et al. (2020) [74]	MC	$8,418$	3 - NL, MNL, DNN	DNN

Notes: MC - mode choice; CO - car ownership; TP - trip purposes. Each row is not a study, but an experiment that compares models conditioning on the same choice category and sample size

Table 1: Literature data of 35 studies and 136 experiments

Table 1 can inform the design of our own experiments to complement the existing studies. In the past literature, the common models include MNL, NL, DNN, GLM, BM, BOOSTING, BAGGING, DT, RF, and SVM. The contextual factors include data sets, sample sizes, and choice categories. The sample size ranges from 381 to 230,608 observations[2], and choice categories include travel mode choice, car ownership, and trip purposes. These values are used as baselines for our own experiment design. However, the literature table also demonstrates the weaknesses of past studies. Each study incorporated only a small number of models, and typically failed to recognize the richness within the model families. For example, DNNs can take a variety of architectures and hyperparameters, so a single DNN architecture may not represent the performance of a vast DNN model family. Past

[2]The observations are the trips, so the number of individuals and households is smaller.

studies also miss some important models. On the ML side, the missing models are discriminant analysis (DA) and generalized linear models (GLM). On the DCM side, the missing models are the mixed logit model (MXL), which is a workhorse in practical travel demand modeling. Most importantly, such context-specific studies cannot reveal the general trends and the variations across these findings, which shall be addressed through this benchmark study.

4. Experimental design

Informed by the literature data, our experiments expand upon four major dimensions ($|H| = 4$), including 105 models ($|T_m| = 105$) belong to 12 model families, 3 data sets ($|T_s| = 3$), 3 sample sizes ($|T_n| = 3$), and 3 choice categories ($|T_y| = 3$). In total, it leads to the number of experiment points given by $T = \prod_{h \in H} |T_h|$; $h \in H = \{m, s, n, y\}$, in which m represents models; s datasets; n sample sizes; y outputs; T the total number of experiments; and T_h the cardinality of each dimension. Roughly speaking, our experiments can be treated as a grid search along the four dimensions. Among the four dimensions, the models are the main factor of interest, because the majority of the past studies focused on model comparison. The other three dimensions are contextual factors.

4.1. Treatment dimension: models and model families

Table 2 summarizes the list of 105 individual models (Column 1) from 12 model families (Column 2). Instead of analyzing single models, we create a two-tier structure for models and model families, thus recognizing the richness within each model family. For example, the DNN family includes 16 DNNs with different architectures, thus enabling us to analyze the variation across the DNN architectures. While far from exhaustive, Table 2 has included the relatively representative models and the most relevant ones for travel behavioral analysis. For example, DCMs are highly relevant given their consistent use in the travel demand analysis, and DNNs are also highly relevant due to their rising popularity in many subdomains in transportation [40, 72, 74].

Classifiers	Model Families	Description	Language & Function
1. Discrete Choice Models (3 Models)			
mnl_B	DCM	Multinomial logit model	Python Biogeme
nl_B	DCM	Nested logit model	Python Biogeme
mxl_B	DCM	Mixed logit model (ASC's as random variables)	Python Biogeme
2. Deep Neural Networks (15 Models)			
mlp_R	DNN	Multi-layer perceptrons (MLP)	R RSNNS mlp
avNNet_R	DNN	Neural network with random seeds with averaged scores; [57]	R Caret avNNet
nnet_R	DNN	Single layer neural network with BFGS algorithm	R Caret nnet
pcaNNet_R	DNN	PCA pretraining before applying neural networks	R Caret pcaNNet
monmlp_R	DNN	MLP with monotone constraints [81]	R Caret monmlp
mlp_W	DNN	MLP with sigmoid hidden neurons and unthresholded linear output neurons	Weka MultilayerPerceptron

DNN_1_30_P	DNN	MLP with one hidden layer and 30 neurons in each layer	Python Tensorflow
DNN_3_30_P	DNN	MLP with three hidden layers and 30 neurons in each layer	Python Tensorflow
DNN_5_30_P	DNN	MLP with five hidden layer and 30 neurons in each layer	Python Tensorflow
DNN_1_100_P	DNN	MLP with one hidden layer and 100 neurons in each layer	Python Tensorflow
DNN_3_100_P	DNN	MLP with three hidden layers and 100 neurons in each layer	Python Tensorflow
DNN_5_100_P	DNN	MLP with five hidden layers and 100 neurons in each layer	Python Tensorflow
DNN_1_200_P	DNN	MLP with one hidden layer and 200 neurons in each layer	Python Tensorflow
DNN_3_200_P	DNN	MLP with three hidden layers and 200 neurons in each layer	Python Tensorflow
DNN_5_200_P	DNN	MLP with five hidden layers and 200 neurons in each layer	Python Tensorflow

3. Discriminant Analysis (12 Models)

lda_R	DA	Linear discriminant analysis (LDA) model	R Caret lda
lda2_R	DA	LDA tuning the number of components to retain up to #classes - 1	R Caret lda2
lda_P	DA	LDA solved by singular value decomposition without shrinkage	Python sklearn LinearDiscriminantAnalysis
sda_R	DA	LDA with Correlation-Adjusted T (CAT) scores for variable selection	R Caret sda
lda_shrink_P	DA	LDA solved by least squares with automatic shrinkage based on Ledoit-Wolf lemma used.	Python sklearn LinearDiscriminantAnalysis
slda_R	DA	LDA developed based on left-spherically distributed linear scores	R Caret ipred
stepLDA_R	DA	LDA model with forward/backward stepwise feature selection	R Caret stepLDA
pda_R	DA	Penalized discriminant analysis (PDA) with shrinkage penalty coefficients [30]	R mda gen.ridge
mda_R	DA	Mixture discriminant analysis (MDA) where the number subclass is tuned to 3 [31]	R mda
rda_R	DA	Regularized discriminant analysis (RDA) with regularized group covariance matrices [24]	R klaR
hdda_R	DA	High dimensional discriminant analysis (hdda) assuming each class in a Gaussian subspace [6]	R HD
qda_P	DA	Quadratic discriminant analysis (qda)	Python sklearn QuadraticDiscriminantAnalysis

4. Bayesian Models (7 Models)

naive_bayes_R	BM	Naive Bayes (NB) classifier with the normal kernel density (Laplace correction factor = 2 and Bandwidth Adjustment = 1)	R naivebayes

nb_R	BM	NB classifier with the normal kernel density (Laplace correction factor = 2 and Bandwidth Adjustment = 1)	R Caret nb
BernoulliNB_P	BM	NB model with Bernoulli kernel density function	Python sklearn BermoulliNB
GaussianNB_P	BM	NB model with Gaussian kernel density function (smoothing = 5, according to the variance portions)	Python sklearn GaussianNB
MultinomialNB_P	BM	NB model with multinomially distributed data (smoothing = 1 and learn class prior probabilities)	Python sklearn MultinomialNB
BayesNet_W	BM	Bayes network models by hill climbing algorithm [13]	Weka BayesNet
NaiveBayes_W	BM	NB model with Gaussian kernel density function	Weka NaiveBayes

5. Support Vector Machines (9 Models)

svmLinear_R	SVM	Support Vector Machine (SVM) model with linear kernel (inverse kernel width = 1)	R Caret kernlab
svmRadial_R	SVM	Support Vector Machine (SVM) model with Gaussian kernel (inverse kernel width = 1)	R Caret kernlab
svmPoly_R	SVM	SVM with polynomial kernel	R Caret kernlab
lssvmRadial_R	SVM	Least Squares SVM model with Gaussian kernel	R Caret kernlab
LinearSVC_P	SVM	SVM with linear kernel and l2 penalty	Python sklearn LinearSVC
SVC_linear_P	SVM	SVM with linear kernel (regularization parameter = 1)	Python sklearn SVC
SVC_poly_P	SVM	SVM with polynomial kernel (regularization parameter = 1)	Python sklearn SVC
SVC_rbf_P	SVM	SVM with radial basis function (rbf) kernel (regularization parameter = 1)	Python sklearn SVC
SVC_sig_P	SVM	SVM with sigmoid function kernel (regularization parameter = 1)	Python sklearn SVC

6. K Nearest Neighbors (4 Models)

KNN_1_P	KNN	k-nearest neighbors (KNN) classifier with number of neighbors equal to 1	Python sklearn KNeighborsClassifier
KNN_5_P	KNN	KNN classifier with number of neighbors equal to 5	Python sklearn KNeighborsClassifier
lBk_1_W	KNN	KNN classifier with number of neighbors equal to 1 (brute force searching and Euclidean distance) [2]	Weka lBk
lBk_5_W	KNN	KNN classifier with number of neighbors equal to 5 (brute force searching and Euclidean distance) [2]	Weka lBk

7. Decision Tree (14 Models)

rpart_R	DT	Recursive partitioning and regression trees (RPART) model (max depth = 30)	R rpart
rpart2_R	DT	RPART (max depth = 10)	R Caret klaR
C5.0Tree_R	DT	C5.0 decision tree (confidence factor = 0.25)	R Caret C5.0Tree
C5.0Rules_R	DT	Rule-based models using Quinlan's C5.0 algorithm [55]	R Caret C5.0Rules
ctree_R	DT	Conditional inference trees [37]	R Caret ctree
ctree2_R	DT	Conditional inference trees (max depth = 10)	R Caret ctree2
DecisionTree_P	DT	Decision tree classification model with Gini impurity split measure	Python sklearn DecisionTreeClassifier

ExtraTree_P	DT	Tree classifier with best splits and features chosen from random splits and randomly selected features [25]	Python sklearn ExtraTreeClassifier
DecisionStump_W	DT	Tree model with decision stump	Weka DecisionStump
HoeffdingTree_W	DT	An incremental tree with inductive algorithm. [38]	Weka HoeffdingTree
REPTree_W	DT	Tree model using information gain/variance	Weka REPTree
J48_W	DT	Pruned C4.5 decision tree model	Weka J48
Attribute Selected_W	DT	Use J48 trees to classify patterns reduced by attribute selection (Hall, 1998)	Weka AttributeSelected
DecisionTable_W	DT	Simple decision table majority classier that uses BestFirst as search method [41]	Weka DecisionTable

8. Generalized Linear Models (10 Models)

Logistic Regression_l1_P	GLM	Logistic regression model with l1 penalty	Python sklearn LogisticRegression
Logistic Regression_l2_P	GLM	Logistic regression model with l2 penalty	Python sklearn LogisticRegression
Logistic_W	GLM	Logistic regression model with a ridge estimator [43]	Weka Logistic
SimpleLogistic_W	GLM	Linear logistic regression models fitted by using LogitBoost [42]	Weka SimpleLogistic
Ridge_P	GLM	Classifier using Ridge regression	Python sklearn RidgeClassifier
Passive Aggressive_P	GLM	Passive-aggressive algorithms for classification with hinge loss [14]	Python sklearn PassiveAggressiveClassifier
SGD_Hinge_P	GLM	Linear classifier with hinge loss and SGD training	Python sklearn SGDClassifier
SGD_Squared Hinge_P	GLM	Linear classifiers of SGD training with squared hinge loss function	Python sklearn SGDClassifier
SGD_Log_P	GLM	Linear classifiers of SGD training with log loss function	Python sklearn SGDClassifier
SGD_Modified Huber_P	GLM	Linear classifiers of SGD training with modified huber loss function	Python sklearn SGDClassifier

9. Gaussian Process (3 Models)

GP_Constant_P	GP	Gaussian Processes classification model with constant kernel	Python sklearn GaussianProcessClassifier
GP_DotProduct_P	GP	Gaussian Processes classification model with DotProduct kernel	Python sklearn GaussianProcessClassifier
GP_Matern_P	GP	Gaussian Processes classification model with Matern kernel	Python sklearn GaussianProcessClassifier

10. Bagging (3 Models)

Bagging_SVM_P	BAGGING	A bagging classifier that fits base classifiers based on random subsets of the original dataset; SVM is the base classifier	Python sklearn BaggingClassifier
Bagging_Tree_P	BAGGING	A bagging classifier with DecisionTree as the base classifier	Python sklearn BaggingClassifier
Voting_P	BAGGING	A classifier which combine machine learning classifiers and use a majority vote. We use lda_P, LinearSVM and Logistic classifiers here.	Python sklearn VotingClassifier

11. Random Forests (2 Models)

RandomForest_P	RF	A random forest model with 10 trees in the forest	Python sklearn RandomForestClassifier
ExtraTrees_P	RF	A meta estimator that fits 10 ExtraTree classifiers	Python sklearn ExtraTreeClassifier

12. Boosting (23 Models)

AdaBoost_P	BOOSTING	AdaBoost classifier. The DecisionTree with maximum depth =10 is set as the base estimator. [22]	Python sklearn AdaBoostClassifier
AdaBoostM1_W	BOOSTING	Boosting method with DecisionStump as the base classifier	Weka AdaboostM1
AdaBoostM1_R	BOOSTING	Boosting method with DecisionTree as the base classifier	R adabag Adaboost.M1
LogitBoost_R	BOOSTING	Logitboost classification algorithm using decision stumps (one node decision trees) as base learners.	R LogitBoost
Gradient Boosting_P	BOOSTING	An additive model trained in a forward stage-wise fashion [23]	Python sklearn GradientBoostingClassifier
DNN_1_30 AdaBoost_P	BOOSTING	AdaBoosting method with DNN_1_30_P as the base classifier	Python Tensorflow
DNN_3_30 AdaBoost_P	BOOSTING	AdaBoosting method with DNN_3_30_P as the base classifier	Python Tensorflow
DNN_5_30 AdaBoost_P	BOOSTING	AdaBoosting method with DNN_5_30_P as the base classifier	Python Tensorflow
DNN_1_100 AdaBoost_P	BOOSTING	AdaBoosting method with DNN_1_100_P as the base classifier	Python Tensorflow
DNN_3_100 AdaBoost_P	BOOSTING	AdaBoosting method with DNN_3_100_P as the base classifier	Python Tensorflow
DNN_5_100 AdaBoost_P	BOOSTING	AdaBoosting method with DNN_5_100_P as the base classifier	Python Tensorflow
DNN_1_200 AdaBoost_P	BOOSTING	AdaBoosting method with DNN_1_200_P as the base classifier	Python Tensorflow
DNN_3_200 AdaBoost_P	BOOSTING	AdaBoosting method with DNN_3_200_P as the base classifier	Python Tensorflow
DNN_5_200 AdaBoost_P	BOOSTING	AdaBoosting method with DNN_5_200_P as the base classifier	Python Tensorflow
DNN_1_30 GradientBoost_P	BOOSTING	Gradient boosting method with DNN_1_30_P as the base classifier	Python Tensorflow
DNN_3_30 GradientBoost_P	BOOSTING	Gradient boosting method with DNN_3_30_P as the base classifier	Python Tensorflow
DNN_5_30 GradientBoost_P	BOOSTING	Gradient boosting method with DNN_5_30_P as the base classifier	Python Tensorflow
DNN_1_100 GradientBoost_P	BOOSTING	Gradient boosting method with DNN_1_100_P as the base classifier	Python Tensorflow
DNN_3_100 GradientBoost_P	BOOSTING	Gradient boosting method with DNN_3_100_P as the base classifier	Python Tensorflow
DNN_5_100 GradientBoost_P	BOOSTING	Gradient boosting method with DNN_5_100_P as the base classifier	Python Tensorflow

DNN_1_200 Gradient-Boost_P	BOOSTING	Gradient boosting method with DNN_1_200_P as the base classifier	Python Tensorflow
DNN_3_200 Gradient-Boost_P	BOOSTING	Gradient boosting method with DNN_3_200_P as the base classifier	Python Tensorflow
DNN_5_200 Gradient-Boost_P	BOOSTING	Gradient boosting method with DNN_5_200_P as the base classifier	Python Tensorflow

Table 2: List of 105 ML classifiers from 12 model families

This list of models has substantially enriched the models from the literature data. Particularly, the DCM family incorporates MNL, NL, and MXL, thus generating insights from three major DCMs and comparing the performance within the DCM family. Theoretically, we can further enrich the DCMs beyond Table 2. For example, the NL models can be expanded using different nest structures; the MXL models by incorporating a flexible correlation matrix or modeling the panel structure in data. It is also possible to expand the substantial scope by examining the departures from purely rational models in choice modelling. However, given the already tremendous scale of our experiments, we decided to use only baseline DCM configurations for simplicity. Nonetheless, the three DCMs are repeatedly trained for various sample sizes, choice categories, and data sets, leading to thousands of experiments and providing sufficiently broad insights and a certain guarantee of robustness in relation to the DCMs' performance.

4.2. Contextual factors

The contextual variables are data sets, sample sizes, and choice categories, which can also influence the model performance besides the model choice. The data sets include NHTS2017, LTDS2015, and SGP2017. NHTS2017 refers to the national household travel survey 2017, collected by the Federal Highway Administration[3], which provides travel, vehicle, and individual information across the United States. LTDS2015 refers to the London travel demand survey 2015, which was collected in London with the history of trips from April 2012 to March 2015 and augmented with mode-specific level-of-service variables (e.g. in-vehicle travel time, public transport fares, fuel cost, etc.) [33]. SGP2017 was collected in 2017 in Singapore through a stated preference survey for travel mode choice [75, 61, 47], in which the experiments followed the orthogonal survey design [45]. SGP2017 contains information about travel mode choices, socio-demographics, and alternative specific variables. The three data sets are chosen to cover a variety of data collection procedures, geographical locations, and local contexts. The NHTS2017 was collected through revealed preference surveys by the US government, LTDS2015 by combining a transit agency's survey with simulation-based travel information, and SGP2017 by a standard stated preference survey. The three datasets cover the continents of America, Europe, and Asia, thus jointly creating a geographically diverse set.

---

[3]Available at https://nhts.ornl.gov/

The sample sizes span 1,000, 10,000, and 100,000 observations, which are of the same magnitude as the sample sizes from the literature data. When a data set has a large sample size (>100,000), it is re-sampled repeatedly to test the impacts of sample size on performance. For example, the NHTS2017 data contains 781,831 trips, among which 1,000, 10,000, and 100,000 observations are repeatedly resampled for predicting travel modes and trip purposes. It also contains 110,565 households, among which the observations are resampled for car ownership prediction. Similar sampling schemes are applied to the LTDS2015 and SGP2017 data sets.

The choice categories include three travel behavior variables: travel modes, trip purposes, and car ownership, which are amongst the most common choice categories from the literature. Travel mode choice is available for all three data sets, while trip purposes and car ownership are only available for the NHTS2017 data. In the NHTS2017 data, the initial 21 travel mode categories are aggregated into six major modes: walk and bike, car, SUV, van and track, public transit, and others. The NHTS2017 data has five trip purposes, including home-based work, home-based shopping, home-based social, home-based others, and none-home-based trips. The car ownership in the NHTS2017 data also has five categories, ranging from zero to more than three cars. The outputs from the three data sets are summarized in Figure 1. In the NHTS2017 data, the most common travel mode is car, accounting for 45.3%, while only 1.8% of the trips are taken by public transit. In the LTDS2015 and SGP2017 data sets, the largest mode share is driving, which accounts for 44.1% and 44.7%, respectively. Public transit is the second largest mode share in London and Singapore, accounting for 35.3% and 23.0% of the total trips.

Due to the large scale of our experiments, we have to somewhat compromise the completeness of our experiments for successful implementation. We chose the most important input features from three data sets using χ^2 feature selection, without further randomization in feature selection algorithms. Although the theoretical maximal number of experiments equals to $105 \times 3 \times 3 \times 3 \times 5 = 14,175$, we completed only $6,970$ experiments due to limitations in data availability, storage, and computational difficulty (See Appendix II for details).

(a) NHTS2017-MC   (b) NHTS2017-TP   (c) NHTS2017-CO
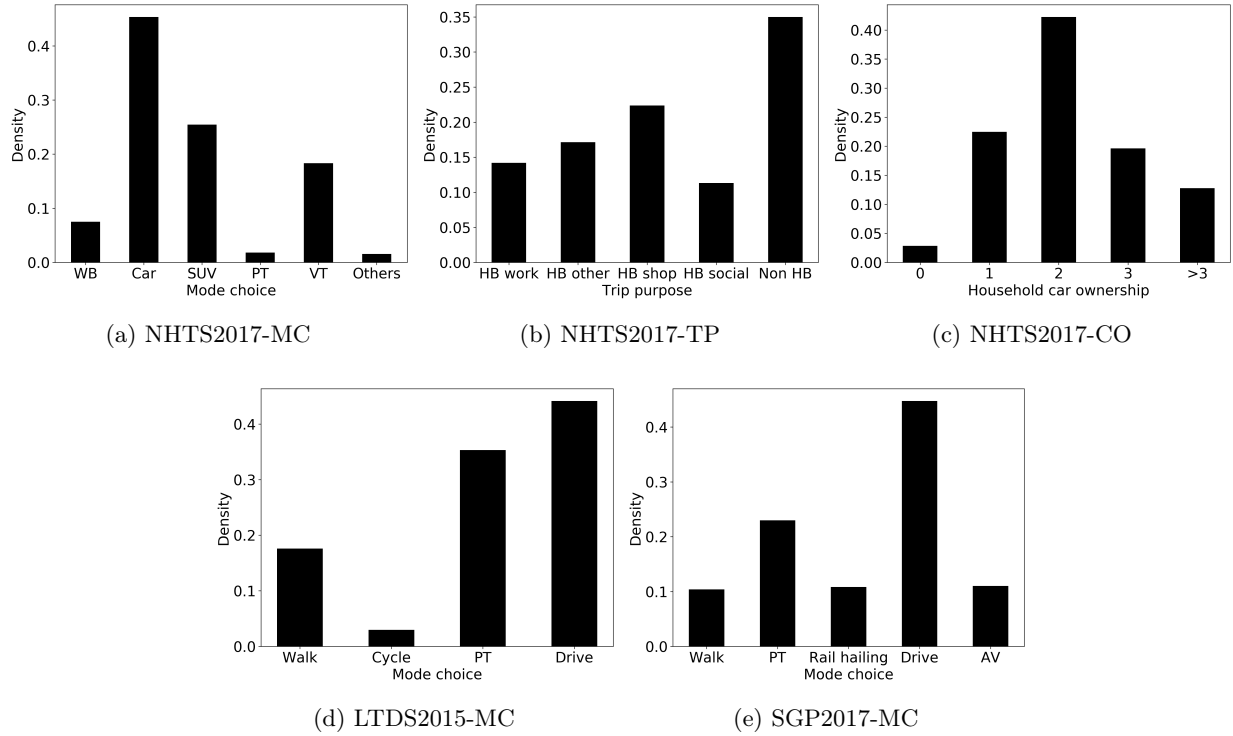
(d) LTDS2015-MC   (e) SGP2017-MC

Fig. 1. Output distributions in three datasets

5. Results

5.1. Tournament models

Tables 3 and 4 present the results of tournament models for the literature and the experiment data. In both tables, Models 1-3 correspond to the regressions using only model-family effects, combining model-family and contextual factors, and adding interactions of model-family and contextual factors. The main dimension of interests is the twelve model families represented by eleven dummy variables with DCM as the baseline category. Hence the $\beta's$ measure the difference of the other families from the DCMs. The contextual variables include data sources, choice categories, input and output dimensions, and sample size. In both tables, Model 2 is used mainly for model interpretation because the linear model-family and contextual effects are directly interpretable, while Model 3 is designed to showcase the potential improvement in the fitting of the tournament model. For computational efficiency, we randomly sampled 2,000 pairs of models from the two data sources as inputs of the tournament models for fair comparison. The tournament models yield four findings below.

First, we find that many ML models indeed outperform DCMs, as suggested by the statistical significance from the positive coefficients of the ML model families. In the literature data (Columns 1 and 2 in Table 3), the majority of the ML families have positive coefficients with high levels of statistical significance. Specifically, bagging, boosting, DNNs, decision trees, and random forests

13

Table 3: Results of Tournament Models (literature data)

Parameters	Tournament Models		
	1. Model families	2. With contexts	3. With interactions
Model family effects (β)			
DCM	Fix to 0	Fix to 0	Fix to 0
BAGGING	2.870 (0.746) ***	4.630 (0.840) ***	1.110 (1.330)
BM	-1.030 (0.317) **	-0.183 (0.375)	-3.800 (1.040) ***
BOOSTING	1.050 (0.138) ***	1.910 (0.187) ***	-2.040 (0.970) *
DNN	0.270 (0.096) **	0.921 (0.124) ***	-2.150 (0.975) *
DT	1.280 (0.126) ***	1.450 (0.166) ***	-1.920 (0.965) *
GLM	-7.380 (12.30)	-5.730 (18.80)	-8.790 (16.70)
KNN	0.096 (0.544)	0.973 (0.631)	-2.230 (1.160)
RF	1.520 (0.162) ***	2.300 (0.217) ***	-1.370 (0.989)
SVM	1.010 (0.125) ***	1.110 (0.155) ***	-2.170 (0.998) *
Contextual effects (γ)			
If_US	/	Fix to 0	Fix to 0
If_Asian	/	-0.559 (0.122) ***	-0.233 (0.156)
If_AM	/	-2.200 (0.353) ***	-2.850 (0.386) ***
If_Europe	/	-1.790 (0.142) ***	-1.830 (0.196) ***
If_TM	/	Fix to 0	Fix to 0
If_CO	/	-2.840 (0.250) ***	-2.510 (0.295) ***
If_TP	/	-1.520 (0.183) ***	-1.610 (0.228) ***
Input dimension	/	-0.067 (0.008) ***	/
Num of alternatives	/	-0.442 (0.038) ***	/
Sample size ($\times 10^5$)	/	-0.067 (0.097)	/
Input dimension (DCM)	/	/	0.097 (0.059)
(Input dimension (DCM))^2	/	/	-0.003 (0.002)
Input dimension (ML)	/	/	-0.272 (0.034) ***
(Input dimension (ML))^2	/	/	0.008 (0.001) ***
Num of alternatives (DCM)	/	/	-4.040 (0.373) ***
(Num of alternatives (DCM))^2	/	/	0.355 (0.036) ***
Num of alternatives (ML)	/	/	-1.390 (0.233) ***
(Num of alternatives (ML))^2	/	/	0.104 (0.022) ***
Sample size ($\times 10^5$) (DCM)	/	/	0.098 (0.881)
(Sample size ($\times 10^5$) (DCM))^2	/	/	-0.052 (0.437)
Sample size ($\times 10^5$) (ML)	/	/	-1.300 (0.448) **
(Sample size ($\times 10^5$) (ML))^2	/	/	0.673 (0.214) **
Statistical summary			
Num of Observations	2000	2000	2000
Final log likelihood	-1,233.868	-916.866	-802.047
Null log likelihood	-1,386.294	-1,386.294	-1,386.294
AIC	2,487.736	1,869.731	1,658.094
ρ^2	0.11	0.339	0.421
Adjusted ρ^2	0.103	0.326	0.402
Average probability of correct prediction	0.57	0.7	0.739
Hit rate	0.652	0.774	0.792

can outperform DCMs in predicting travel behavior in a statistically significant manner with and without controlling the contextual variables. By comparing the magnitudes of the coefficients (e.g., Model 2), we can rank the performance from the highest to the lowest as bagging, random forest,

boosting, decision tree, DNNs, DCMs, etc. This finding is further reinforced in our experiment data. In Columns 1-2 of Table 4, the ensemble methods and DNNs continue to outperform the benchmark DCMs, although the statistical significance is much lower than the literature data. Since the model-family coefficients only reveal the average effects across model families, we will delve into the variance in and the best models of the model families in the sections below. In both literature and experiment data, naive bayesian models and Gaussian process underperform DCMs. The performance of K-nearest neighbors and support vector machines is inconclusive due to their inconsistent signs of coefficients across the literature and experiment data. In short, our tournament models demonstrate that many ML models, particularly ensemble methods and DNNs, can statistically outperform DCMs in predicting travel behavior, consistent with the majority of the previous deterministic comparisons.

However, our results also demonstrate that the contextual effects ($\gamma's$) are more dominating than the choice of models ($\beta's$) in influencing model performance. In Model 1 of Tables 3 and 4, where contextual controls are not imposed, the pseudo R^2 reaches only 0.11 and 0.06. But after adding the contextual variables (Model 2 in Tables 3 and 4), the pseudo R^2 increases drastically to 0.34 and 0.43. This explanatory power of the contextual variables is also demonstrated by their statistical significance, since the contextual variables are much more significant than the model variables. Nearly all the contextual variables are statistically significant in literature and experiment data, while only around 50% of the model family coefficients are statistically significant. The coefficients of the contextual effects suggest that the model performance in the US context is typically higher than Asian and European contexts, and that the predictive performance in travel mode choice is higher than that in trip purposes and car ownership, at least in the literature data. The contextual effects also suggest that the intrinsic predictive power decreases with a higher number of alternatives and input dimensions, and increases with larger sample size. A large number of alternatives lead to challenges in model prediction, and large sample size facilitates the parameter training, thus improving model performance. Different from our expectations, high input dimensions reduce model performance, so we investigate such an issue in Appendix VI. Meanwhile, the contextual variables and the model families appear to exhibit some interaction effect, because the contextual variables can amplify the model family effects. Model 3 in Tables 3 and 4 demonstrate that such interaction effects exist, since Model 3 further improves the pseudo R^2 to 0.42 and 0.45. This finding highlights the significant contextual variations in model performance, implying that it might be inappropriate to abstractly compare model performance without explicitly examining the contextual effects.

Third, even after incorporating all the model and contextual factors, the tournament model exhibits significant residual randomness, implying that inherent uncertainty could exist in such model comparisons. In Model 3, the pseudo R^2 of the tournament models reaches only 0.42 and 0.45 for the literature and experiment data. It suggests that at least half of variation in model comparisons remains purely random. This residual randomness could be attributed to the lack of critical contextual variables, incomplete model specification, or inherent randomness in such comparison tasks. Although it is always possible to further improve the explanatory power by incorporating

Parameters	Tournament Models		
	1. Model families	2. With contexts	3. With interactions
Model family effects (β)			
DCM	Fix to 0	Fix to 0	Fix to 0
BAGGING	0.278 (0.285)	-0.138 (0.385)	0.552 (0.253) *
BM	-0.966 (0.251) ***	-2.010 (0.354) ***	-1.430 (0.207) ***
BOOSTING	0.271 (0.219)	0.221 (0.301)	0.826 (0.130) ***
DA	-0.131 (0.223)	-0.659 (0.309) *	-0.073 (0.146)
DNN	0.468 (0.226) *	0.482 (0.312)	1.100 (0.148) ***
DT	0.184 (0.224)	0.064 (0.306)	0.684 (0.135) ***
GLM	0.127 (0.225)	-0.120 (0.308)	0.546 (0.147) ***
GP	-0.909 (0.404) *	-1.740 (0.528) ***	-0.837 (0.403) *
KNN	-0.790 (0.272) **	-1.620 (0.374) ***	-1.020 (0.240) ***
RF	0.388 (0.306)	0.371 (0.416)	1.120 (0.298) ***
SVM	-0.748 (0.245) **	-1.600 (0.340) ***	-0.981 (0.182) ***
Contextual effects (γ)			
If_US	/	Fix to 0	Fix to 0
If_Asian	/	-0.599 (0.033) ***	-0.715 (0.050) ***
If_Europe	/	1.100 (0.091) ***	0.899 (0.112) ***
If_TM	/	Fix to 0	Fix to 0
If_CO	/	0.553 (0.102) ***	0.122 (0.103)
If_TP	/	-0.217 (0.099) *	-0.654 (0.106) ***
Input dimension	/	-0.013 (0.005) **	/
Num of alternatives	/	-1.750 (0.091) ***	/
Sample size ($\times 10^5$)	/	0.786 (0.115) ***	/
Input dimension (DCM)	/	/	-0.149 (0.289)
(Input dimension (DCM))^2	/	/	0.002 (0.004)
Input dimension (ML)	/	/	-0.114 (0.054) *
(Input dimension (ML))^2	/	/	0.002 (0.001) *
Num of alternatives (DCM)	/	/	-0.284 (1.130)
(Num of alternatives (DCM))^2	/	/	-0.173 (0.153)
Num of alternatives (ML)	/	/	-0.834 (1.120)
(Num of alternatives (ML))^2	/	/	-0.130 (0.110)
Sample size ($\times 10^5$) (DCM)	/	/	16.70 (8.490) *
(Sample size ($\times 10^5$) (DCM))^2	/	/	-14.60 (8.030)
Sample size ($\times 10^5$) (ML)	/	/	10.30 (1.390) ***
(Sample size ($\times 10^5$) (ML))^2	/	/	-9.120 (1.320) ***
Statistical summary			
Num of Observations	2000	2000	2000
Final log likelihood	-1,293.057	-789.98	-762.561
Null log likelihood	-1,386.294	-1,386.294	-1,386.294
AIC	2,610.114	1,617.960	1,581.123
ρ^2	0.067	0.43	0.45
Adjusted ρ^2	0.059	0.416	0.43
Average probability of correct prediction	0.545	0.748	0.759
Hit rate	0.617	0.832	0.84

more factors or enriching model specification, our finding suggests at least that researchers should not simply conclude the superiority of ML methods over DCMs without in-depth statistical analysis. Uncertainty prevails in such model comparisons, and researchers could seek to quantify the

confidence in the comparative results.

Lastly, besides the consistent patterns between the literature and experiment results, we also identify many inconsistent patterns, which could reveal potential publication biases. On average, ML models have more positive coefficients and higher statistical significance in the literature data than the experiment data. For example, the coefficients of bagging, boosting, and random forest in literature data are 4.63, 1.91, and 2.30 (Table 3), which are much larger than -0.14, 0.22, and 0.37 in the experiment data (Table 4). Meanwhile, the three coefficients are statistically significant at the 99.99% confidence level in the literature data, while they are insignificant in the experiment data. This finding implies that the published studies are more likely than our experiments to identify ML models as the dominating ones. Such publication biases could be caused by researchers' self-filtering or journals' preferences for ML models' dominating performance over traditional DCMs. Regarding the contextual variables, both data sets indicate that it becomes harder to compare models in travel purpose predictions and with higher dimensional outputs. But the location and other contextual effects are inconsistent between the two data sources, which implies some difficulty in identifying consistent contextual effects for model performance.

The tournament model summarizes ranking of model pairs, but it does not provide direct insights into prediction accuracy itself. In Appendix III, we introduce a simple linear model with prediction accuracy as outputs, which yield similar conclusions to the tournament models above. Although the tournament models can efficiently summarize experiments on average, they might conceal the rich contents in the variation of the model performance, particularly because modelers often compare across the best models from each model family, rather than the average ones. For example, various DNN architectures are lumped into the DNN family, and the MNL, NL, and MXL models lumped into a single DCM family. Therefore, we will examine the model performance along individual dimensions using visualization and descriptive statistics, thus further distinguishing the differences within and across model families in the two sections below.

5.2. Examining performance across model families and individual models

Figures 2 and 3 present visualizations of prediction accuracy for 12 model families and 105 individual models. Both figures are sorted based on average prediction accuracy, from highest to lowest. In Figure 2, the x-axis represents model families, while the y-axis represents prediction accuracy. Each white dot represents an experiment's accuracy, and the grey areas indicate density distributions of predictive performance. The blue, red, and green dots represent the mean, median, and maximum prediction accuracy, respectively. Figure 3 extends the 12 model families to 105 classifiers using a similar format, replacing the violin plots with simple blue bars to depict performance variance. In this figure, the blue and red dots respectively represent the mean and median prediction accuracy. Additionally, Appendix IV contains an analysis of literature data, yielding similar findings to the experiment data presented here.

Figure 2 illustrates the relative prediction accuracy of different model families. Among them, ensemble methods and DNNs exhibit relatively high prediction accuracy, while the DCM family
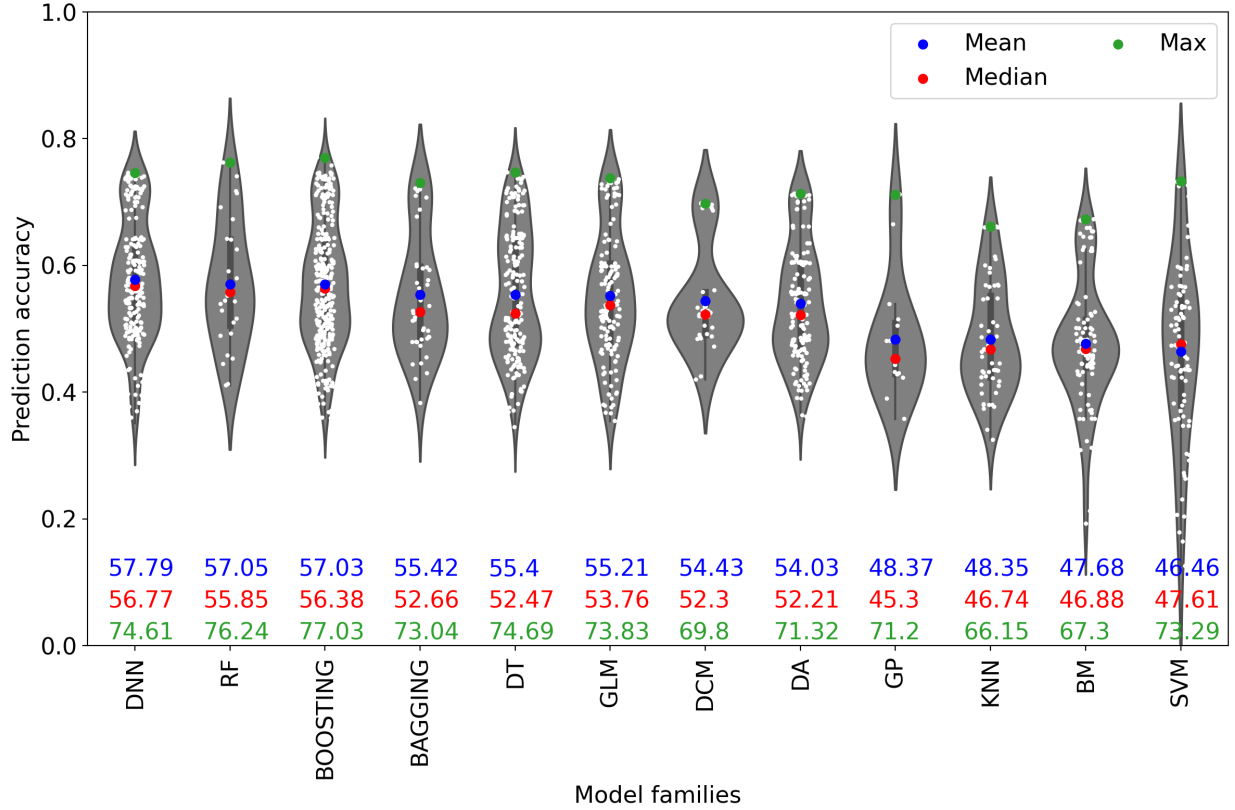
Fig. 2. Prediction accuracy of 12 model families

ranks 7th out of 12. The DNN family, comprising 15 classifiers, achieves mean and median accuracies of 57.79% and 56.77% respectively. The three ensemble methods, namely random forests, boosting, and bagging, also demonstrate high accuracy with mean values of 57.05%, 57.03%, and 55.42% respectively. Despite incorporating more complex error structures in the NL and MXL models, the DCMs still fail to outperform the ensemble methods and DNNs. The superior prediction accuracy of ML models can be attributed to their strong approximation power [36] and effective regularization techniques [65, 7]. This visualization further confirms the findings from our previous tournament model, providing additional ranking information for each model family.

Figure 2 also reveals significant variation and even multi-modal distributions in predictive performance. For instance, the DNN family exhibits an average accuracy of approximately 57.79%, while its maximum prediction accuracy reaches around 74.61%. Similarly, the average accuracy of the DCM family stands at 54.43%, which is 15.37% lower than its maximum value of 69.8%. Remarkably, each model family can achieve maximum values in the range of 70-80%, surpassing the average by 20-30 percentage points. Several model families display explicit multi-modal distributions, including DCM and BAGGING. The wide variation and multi-modal distributions in prediction accuracy visually demonstrate that the choice of models may not be the most crucial factor in determining the final predictive performance. This observation aligns with our previous finding regarding the substantial uncertainty inherent in such model comparisons.
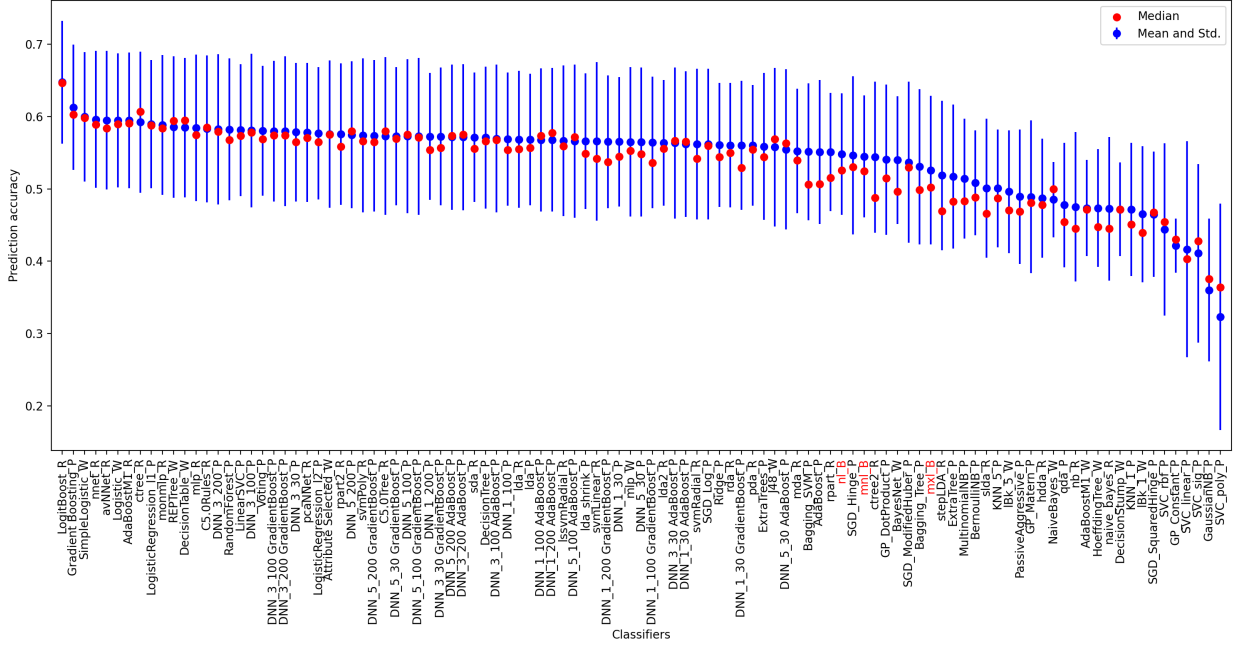
Fig. 3. Prediction accuracy of 105 models

Figure 3 demonstrates that individual models do not deviate significantly from the average performance of their respective model families, and substantial variation persists within each of the 105 individual models. Among these models, the top performers are LogitBoosting and GradientBoosting from the BOOSTING family, followed by DNNs such as nnet, avNNet, and monmlp, which rank 4th, 5th, and 10th out of 105 models. Within the DCM model family, MNL, NL, and MXL (highlighted in red) exhibit similar performance at the medium to lower end of the spectrum among the 105 models. Their average prediction accuracies are 54.50%, 55.01%, and 52.58%, respectively, only slightly deviating from the DCM family's average. However, each individual model represents a significant variation, with performance varying by at least 20 percentage points, as indicated by the long blue bars. The fact that MXL performs worst may come as a surprise to readers, given that in estimation, the inclusion of random heterogeneity almost invariably leads to big gains in performance. However, here, we are talking about prediction performance, which implies averaging again over the random components, and this removes much of the differences between MXL and non-mixture DCM models.

5.3. Examining performance across data sets, outputs, and sample sizes

The substantial uncertainty within model families and individual models demonstrates the importance of investigating the contextual factors' impacts on model performance. Therefore, we further decompose the predictive performance along the contextual dimensions. Figure 4 visualizes the distributions of prediction accuracy in three data sets, and Figure 5 examines the prediction accuracy regarding data sets, outputs, and sample sizes. Table 5 reports the model ranking and performance

19

in three panels and nine sub-tables. Within each sub-table, we report the top-10 models with the DCMs highlighted below, conditioning on specific data sets, outputs, and sample sizes. With such analysis, we could examine the performance consistency of individual models and model families across contexts.
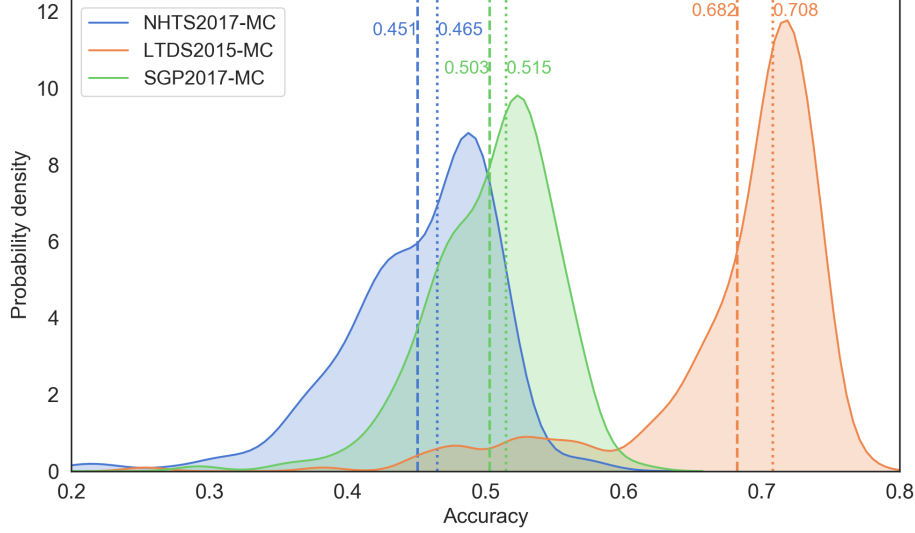

Fig. 4. Prediction accuracy by data sets (dashed and dotted lines represent mean and median values)
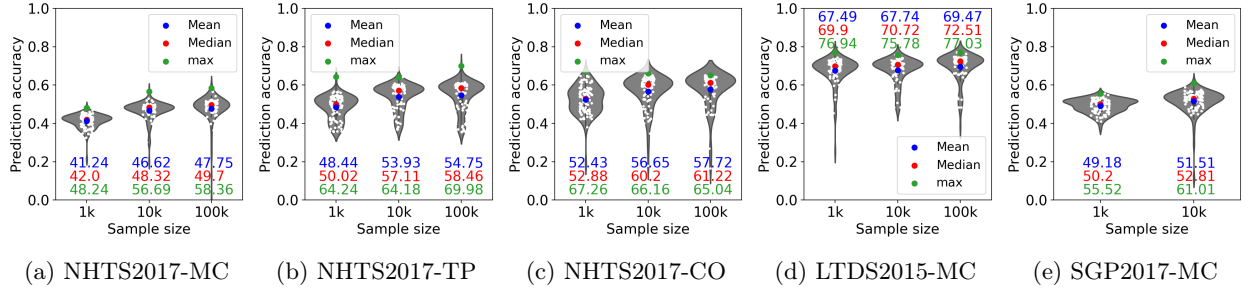

Fig. 5. Prediction accuracy by sample size, datasets, and outputs

Figures 4 and 5 demonstrate that predictive performance exhibits significant variation across datasets and shows a moderate increase with larger sample sizes. While model families display multi-modal performance distributions in Figure 2, the performance distributions of datasets appear more concentrated around the average in Figure 4. The highest accuracy is attained in LTDS2015 (76.58%), which is approximately 20 percentage points higher than NHTS2017 and SGP2017 data. The mean and median values of all three distributions are close, indicating a relatively concentrated distribution within each data set. Furthermore, predictive performance generally improves with sample size, although the rate of improvement appears to diminish slightly. For example, when the sample size is 1,000, the mean prediction accuracy is 51.63%, which increases to around 55.29%

and 57.42% as the sample size grows to 10,000 and 100,000. This trend of marginally diminishing improvement holds consistent across datasets and outputs, as demonstrated in all five subfigures of Figure 5.

Table 5: Model performance specific to data sets, outputs, and sample sizes

Panel 1. Performance across data sets

	NHTS2017-MC			LTDS2015-MC			SGP2017-MC	
Rank	Model	Accuracy	Rank	Model	Accuracy	Rank	Model	Accuracy
1	LogitBoost_R	0.5443	1	LogitBoost_R	0.7658	1	LogitBoost_R	0.5827
2	Gradient Boosting_P	0.5130	2	Gradient Boosting_P	0.7431	2	avNNet_R	0.5564
3	SimpleLogistic_W	0.5001	3	avNNet_R	0.7386	3	nnet_R	0.5552
4	LogisticRegression_l1_P	0.4919	4	nnet_R	0.7351	4	Gradient Boosting_P	0.5550
5	Logistic_W	0.4904	5	SimpleLogistic_W	0.7350	5	DNN_1_200_P	0.5487
6	ctree_R	0.4881	6	Logistic_W	0.7339	6	DNN_1_100 AdaBoost_P	0.5484
7	DecisionTable_W	0.4861	7	svmPoly_R	0.7329	7	DNN_3_100 AdaBoost_P	0.5481
8	monmlp_R	0.4834	8	lssvmRadial_R	0.7320	8	DNN_3_100_P	0.5480
9	LinearSVC_P	0.4822	9	monmlp_R	0.7317	9	DNN_1_100_P	0.5474
10	REPTree_W	0.4808	10	LogisticRegression_l1_P	0.7312	10	DNN_5_200 AdaBoost_P	0.5474
36	nl_B	0.4674	71	mxl_B	0.6980	34	nl_B	0.5270
41	mnl_B	0.4657	76	nl_B	0.6933	35	mnl_B	0.5262
94	mxl_B	0.4250	77	mnl_B	0.6917	44	mxl_B	0.5200

Panel 2. Performance across outputs

	MC (NHTS2017+LTDS2015+SGP2017)			NHTS2017-TP			NHTS2017-CO	
Rank	Model	Accuracy	Rank	Model	Accuracy	Rank	Model	Accuracy
1	LogitBoost_R	0.6369	1	LogitBoost_R	0.6613	1	LogitBoost_R	0.6615
2	Gradient Boosting_P	0.6098	2	Gradient Boosting_P	0.5901	2	rpart2_R	0.6464
3	SimpleLogistic_W	0.5953	3	LogisticRegression_l1_P	0.5859	3	AdaBoostM1_R	0.6464
4	avNNet_R	0.5926	4	ctree_R	0.5825	4	Attribute Selected_W	0.6459
5	nnet_R	0.5921	5	mlp_R	0.5809	5	DecisionTable_W	0.6450
6	Logistic_W	0.5920	6	SimpleLogistic_W	0.5802	6	Gradient Boosting_P	0.6440
7	LogisticRegression_l1_P	0.5894	7	avNNet_R	0.5774	7	ctree_R	0.6400
8	monmlp_R	0.5882	8	Logistic_W	0.5771	8	REPTree_W	0.6380
9	AdaBoostM1_R	0.5834	9	LogisticRegression_l2_P	0.5768	9	SimpleLogistic_W	0.6315
10	LinearSVC_P	0.5832	10	Voting_P	0.5762	10	nnet_R	0.6301
52	nl_B	0.5670	52	nl_B	0.5473	81	mnl_B	0.5124
57	mnl_B	0.5656	69	mnl_B	0.5228	82	nl_B	0.5078
73	mxl_B	0.5477	74	MXL_B	0.5020	86	mxl_B	0.4840

Panel 3. Performance across sample sizes

	1k			10k			100k	
Rank	Model	Accuracy	Rank	Model	Accuracy	Rank	Model	Accuracy
1	LogitBoost_R	0.6244	1	LogitBoost_R	0.6476	1	LogitBoost_R	0.6760
2	Gradient Boosting_P	0.5874	2	Gradient Boosting_P	0.6205	2	Gradient Boosting_P	0.6353
3	AdaBoostM1_R	0.5772	3	avNNet_R	0.6082	3	C5.0Rules_R	0.6303
4	SimpleLogistic_W	0.5752	4	monmlp_R	0.6070	4	ctree_R	0.6295
5	svmPoly_R	0.5741	5	nnet_R	0.6068	5	DNN_5_200_P	0.6261
6	nnet_R	0.5724	6	Logistic_W	0.6054	6	monmlp_R	0.6257
7	avNNet_R	0.5708	7	SimpleLogistic_W	0.6049	7	DNN_3_200_P	0.6253
8	lssvmRadial_R	0.5666	8	DNN_3_200_P	0.6001	8	Logistic_W	0.6248
9	LogisticRegression_l1_P	0.5664	9	DNN_3_100_P	0.5974	9	REPTree_W	0.6244
10	svmLinear_R	0.5658	10	ctree_R	0.5972	10	DNN_5_100_P	0.6244
50	mnl_B	0.5270	14	DNN_5_200_P	0.5943	70	nl_B	0.5683
52	nl_B	0.5256	64	nl_B	0.5600	73	mnl_B	0.5584
57	DNN_5_200_P	0.5144	67	mnl_B	0.5523	-	-	-

Table 5 demonstrates that the relative ranking of individual models remains relatively stable across different contexts, while their absolute prediction accuracy could vary. When considering specific data sets (Panel 1), ensemble methods and DNNs still achieve the highest performance. LogitBoost and GradientBoosting demonstrate dominant performance across all three data sets, and DNNs like avNNet, nnet, and DNN_200_1 often rank among the top 5 models in LTDS2015 and SGP2017 datasets. This observation holds true in Panels 2 and 3, where LogitBoost and Gra-

dientBoosting achieve the highest predictive performance across sample sizes and outputs. DNNs and DTs frequently emerge as top-10 models across all three panels. Meanwhile, the DCMs (MNL, MXL, and NL models) consistently perform at the medium to lower end of the performance distributions in all nine contexts. When conditioning on sample sizes or outputs, the MNL, NL, and MXL models fail to achieve high-quality performance, ranking around the 50-70th position for each sample size.

Apart from the general patterns, the authors find two intriguing observations. Firstly, as the sample size increases, the relative ranking of DCMs and DNNs exhibits an opposite trend: DNNs' ranking improves while DCMs' ranking declines. In Panel 3, when the sample size grows from 1,000 to 100,000, the relative ranking of MNL and NL drops from 50th to 73rd. Conversely, the DNN_200_5 model achieves a prediction accuracy of approximately 51.4% at a sample size of 1,000, but its ranking jumps significantly from 57th to 5th when the sample size reaches 100K. Secondly, the relative ranking of MNL, NL, and MXL is not consistent across contexts. In the NHTS2017-MC data, MXL performs worse than MNL and NL, while it outperforms them in the LTDS2015-MC data. Occasionally, the simplest model (e.g., MNL) attains the highest performance (Panel 2, NHTS2017-CO). This discrepancy may arise because performance evaluation occurs in the testing set, where more complex models do not necessarily outperform simpler ones. Despite the specifics, these two observations support the main finding regarding significant contextual randomness in model performance. Although rankings already offer more stability than absolute prediction accuracy, the ranking of individual models still varies depending on contextual factors such as sample sizes and data sources.

6. Limitations

This benchmark study has important limitations regarding data sources, experiment design, evaluation metrics, and research scope. While our literature and experiment data are relatively comprehensive, they are not exhaustive. The literature data excludes previous studies focusing solely on ML or DCM and those centered on spatiotemporal travel data [49, 54, 78]. Our experiments exclude certain dimensions, such as training algorithms, hyper-parameters, feature selection, and feature transformation, which could influence model performance and change the benchmark results in our tournament model. Additionally, this empirical benchmark study uses prediction accuracy and model ranking as the evaluation criteria, neglecting other metrics like log-likelihood or Brier score [1, 27]. Probabilistic metrics could be more appropriate for model evaluation from a theoretical perspective. However, since log-likelihood as a probabilistic metric does not exist in many deterministic ML models, prediction accuracy is the only metric that enables our comprehensive comparison of both deterministic and probabilistic models. The authors compared only DNNs and DCMs using log-likelihood and Brier scores in Appendix V, with a more comprehensive evaluation with probabilistic metrics left for future studies. Furthermore, model comparison should not solely focus on predictive performance. Recent studies have explored innovations in deep learning

architectures [63, 29, 77], investigated the potential of images and natural language [15, 71], and examined interpretability and robustness as critical dimensions for model evaluation [44, 73, 76]. Lastly, although we find the potential publication biases in favor of ML models' higher performance by comparing literature and experiment data, our own experiment data are not devoid of publication biases. Publication bias appears an inherently challenging topic because any manuscript, including the unpublished ones, contains the authors' intention of being published, and the existing tests for publication biases are only statistical heuristics, which are largely inapplicable to the ML models [67]. Unfortunately, these limitations cannot be addressed in this empirical benchmark work and need be investigated by future researchers.

7. Conclusions

Recent studies comparing DCMs and ML models have often overlooked the randomness in model performance and have been limited to specific contexts, featuring a small number of models and data sets. To overcome these challenges, this study presents an empirical benchmark that introduces a tournament model, examining both literature and experiment data. This approach allows for formal statistical modeling to capture the inherent randomness in model comparisons and summarize results from a large number of experiments. Our data sources are extensive, comprising 6,970 experiments from 105 models and 12 model families in our experiment data, and summarizing 136 experiments from 35 previous studies in the literature data. Our benchmark study yields two findings.

On the one hand, our benchmark study confirms that ML models generally outperform DCMs. The tournament models reveal positive and statistically significant coefficients for ML model families, indicating the superiority of ML models in predictive performance. Ensemble methods such as boosting, bagging, and random forests, along with DNNs, consistently demonstrate the highest predictive performance. In contrast, the DCM family consistently falls within the medium to lower range of the predictive distribution. This trend holds across both literature and experiment data, with or without contextual factors as controls. This consistent pattern applies to both aggregate model families and individual models, as individual model performance aligns closely with the average of their respective model families. The ML models, such as LogitBoost and Gradient-Boosting, consistently achieve dominating performance across all data sets, sample sizes, and choice categories. Overall, our findings reinforce and further validate previous conclusions using formal statistical analysis.

However on the other hand, our benchmark study emphasizes the influence of contextual factors and the residual randomness inherent in model comparisons. It becomes evident that contextual factors play a more significant role in explaining the outcomes of model comparisons than the choice of models themselves. Statistical significance is observed for all contextual factors in our models, whereas the significance of model families is comparatively weaker. Even after considering all model and contextual variables, a considerable amount of residual randomness persists. This residual randomness may arise from incomplete input dimensions or inherent complexity in model comparisons.

Furthermore, our descriptive analysis also reveals significant randomness, as demonstrated by the wide variation in predictive performance among model families and individual models. These findings suggest that the outcomes of performance comparisons are primarily shaped by contextual factors and intrinsic randomness, highlighting the limited perspective of exclusively focusing on model choice.

This empirical benchmark study improves upon the research frontier of the deterministic model comparisons through innovations in methodology, findings, and implications. Regarding methodology, the tournament model emulates and yet generalizes the past deterministic comparisons. It resembles the comparative studies because modelers also seek to identify the model of higher performance through pairwise comparison. Yet, this tournament model generalizes the deterministic comparison by expanding the sample size, examining the impacts of contextual complexity, and adopting formal statistical methods to investigate the effects of model families for both literature and experiment data points. From a modeler's perspective, such a probabilistic approach captures the multiple sources of randomness in model comparison, including data collection, selection of training and testing data, stochastic algorithms, and inherent randomness in every predictive task [18]. In fact, statistical meta-modeling approaches have been widely adopted in the machine learning community for comparing classifiers across data sets, with methods including pairwise comparison [8], block-wise tests [17], or recently Bayesian approaches [69]. From a mathematical perspective, the past deterministic comparisons are only specific cases of our tournament models. The tournament model degenerates into a deterministic comparative study when the variance of ϵ_m approaches zero, the contextual factors are removed, and the sample size of pairwise comparisons is reduced to only one observation. In fact, if a deterministic process represents the underlying data generating process, our probabilistic tournament model should yield very high predictive power (e.g., 100% ρ^2 when no omitted variable exists). However, our empirical finding suggests otherwise, reaching a ρ^2 of only 0.4. Such an empirical result suggests that the performance comparison is much more random than a pure deterministic process.

Regarding findings, the tournament model reveals the generalizable patterns between the literature and the experiment data. By compiling two comprehensive data sets, the authors identify the shared trends across a wide range of experiments, surpassing the limitations of past context-specific analyses. ML and DCMs exhibit relatively consistent rankings across different contexts, and the overall impacts of contextual factors and intrinsic randomness outweigh the influence of model choice in determining model performance. Such findings are corroborated in both data sets. Regarding implications, this work can be continuously updated and widely adopted to evaluate the performance of a large number of models. Researchers could use the tournament models to evaluate model performance, thus partially avoiding the strenuous efforts of training models and collecting data. The coefficients for model families and contextual factors can be used as the prior beliefs in a Bayesian or a Frequentist framework. By augmenting new literature or experiment results, future researchers can track the changes of the model-family or contextual effects in the upcoming years. To facilitate future researchers to replicate our work, we uploaded the scripts and data sets to a

Github repository[4].

Our study highlights two important directions for future research: model transferability and uncertainty quantification. While previous deterministic comparisons have generally yielded correct conclusions, they may overlook the significance of contextual factors when fixed within a specific dataset. To address this, future studies could focus on model transferability across contexts by testing models across diverse datasets or by examining the influence of varying inputs and outputs within the same dataset. Additionally, the presence of significant residual randomness in model comparisons underscores the need to unravel the uncertainty structures in ML models. Unlike statistical methods, current ML models in practice provide deterministic predictions without statistical properties that characterize data or model uncertainty. This deterministic approach often leads to overly simplified conclusion that ML models outperform DCMs without providing statistical confidence in such a statement. Therefore, future research should explore methods for quantifying the uncertainty and testing the transferability for ML models, thus generating robust and generalizable findings regarding the predictive power of ML and DCMs in travel demand analysis.

Acknowledgement

CRediT Author Statement

The authors confirm contribution to the paper as follows. **Shenhao Wang:** Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Project administration, Validation, Writing (original draft), and Writing (review and editing). **Baichuan Mo:** Data curation, Formal analysis, Investigation, Methodology, Software, and Visualization. **Yunhan Zheng:** Formal analysis, Investigation, and Software. **Stephane Hess:** Writing (review and editing). **Jinhua Zhao:** Funding acquisition and Resources.

References

[1]   Moloud Abdar et al. "A review of uncertainty quantification in deep learning: Techniques, applications and challenges". In: *Information Fusion* 76 (2021), pp. 243–297. ISSN: 1566-2535.

[2]   David W Aha, Dennis Kibler, and Marc K Albert. "Instance-based learning algorithms". In: *Machine learning* 6.1 (1991), pp. 37–66.

---

[4]https://github.com/mbc96325/Comparing-machine-learning-and-discrete-choice-model

[3] Mahdieh Allahviranloo and Will Recker. "Daily activity pattern recognition by using support vector machines with multiple classes". In: *Transportation Research Part B: Methodological* 58 (2013), pp. 16–43.

[4] Moshe Ben-Akiva, John L Bowman, and Dinesh Gopinath. "Travel demand model system for the information era". In: *Transportation* 23.3 (1996), pp. 241–266.

[5] James P Biagioni et al. "Tour-based mode choice modeling: using an ensemble of (un-) conditional data-mining classifiers". In: *88th Annual Meeting of the Transportation Research Board, Washington, DC.* 2008.

[6] Charles Bouveyron, Stéphane Girard, and Cordelia Schmid. "High-dimensional discriminant analysis". In: *Communications in Statistics—Theory and Methods* 36.14 (2007), pp. 2607–2623.

[7] Stephen Boyd and Lieven Vandenberghe. *Convex optimization.* Cambridge university press, 2004.

[8] Ralph Allan Bradley and Milton E Terry. "Rank analysis of incomplete block designs: I. The method of paired comparisons". In: *Biometrika* 39.3/4 (1952), pp. 324–345. ISSN: 0006-3444.

[9] Giulio Erberto Cantarella and Stefano de Luca. "Modelling mode choice behaviour for extra-urban journeys". In: *Proceedings of European Transport Conference.* 2002.

[10] Giulio Erberto Cantarella and Stefano de Luca. "Multilayer feedforward networks for transportation mode choice analysis: An analysis and a comparison with random utility models". In: *Transportation Research Part C: Emerging Technologies* 13.2 (2005), pp. 121–155.

[11] Hilmi Berk Celikoglu. "Application of radial basis function and generalized regression neural networks in non-linear utility function specification for travel mode choice modelling". In: *Mathematical and Computer Modelling* 44.7 (2006), pp. 640–658.

[12] Long Cheng et al. "Applying a random forest method approach to model travel mode choice behavior". In: *Travel behaviour and society* 14 (2019), pp. 1–10.

[13] Gregory F Cooper and Edward Herskovits. "A Bayesian method for the induction of probabilistic networks from data". In: *Machine learning* 9.4 (1992), pp. 309–347.

[14] Koby Crammer et al. "Online passive-aggressive algorithms". In: *Journal of Machine Learning Research* 7.Mar (2006), pp. 551–585.

[15] Sander van Cranenburgh et al. "Choice modelling in the age of machine learning-discussion paper". In: *Journal of Choice Modelling* 42 (2022), p. 100340. ISSN: 1755-5345.

[16] Juan De Dios Ortuzar and Luis G Willumsen. *Modelling transport.* John Wiley and Sons, 2011.

[17] Janez Demšar. "Statistical comparisons of classifiers over multiple data sets". In: *The Journal of Machine learning research* 7 (2006), pp. 1–30. ISSN: 1532-4435.

[18] Thomas G Dietterich. "Approximate statistical tests for comparing supervised classification learning algorithms". In: *Neural computation* 10.7 (1998), pp. 1895–1923. ISSN: 0899-7667.

[19] Sean T Doherty and Abolfazl Mohammadian. "Application of artificial neural network models to activity scheduling time horizon". In: *Transportation research record* 1854.1 (2003), pp. 43–49. ISSN: 0361-1981.

[20] Alireza Ermagun, Taha Hossein Rashidi, and Zahra Ansari Lari. "Mode choice for school trips: long-term planning and impact of modal specification on policy assessments". In: *Transportation research record* 2513.1 (2015), pp. 97–105. ISSN: 0361-1981.

[21] Reid Ewing and Robert Cervero. "Travel and the built environment: a meta-analysis". In: *Journal of the American planning association* 76.3 (2010), pp. 265–294. ISSN: 0194-4363.

[22] Yoav Freund and Robert E Schapire. "A decision-theoretic generalization of on-line learning and an application to boosting". In: *Journal of computer and system sciences* 55.1 (1997), pp. 119–139.

[23] Jerome H Friedman. "Greedy function approximation: a gradient boosting machine". In: *Annals of statistics* (2001), pp. 1189–1232.

[24] Jerome H Friedman. "Regularized discriminant analysis". In: *Journal of the American statistical association* 84.405 (1989), pp. 165–175.

[25] Pierre Geurts, Damien Ernst, and Louis Wehenkel. "Extremely randomized trees". In: *Machine learning* 63.1 (2006), pp. 3–42.

[26] Nima Golshani et al. "Modeling travel mode and timing decisions: Comparison of artificial neural networks and copula-based joint model". In: *Travel Behaviour and Society* 10 (2018), pp. 21–32. ISSN: 2214-367X.

[27] Sebastian Gruber and Florian Buettner. "Better Uncertainty Calibration via Proper Scores for Classification and Beyond". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 8618–8632.

[28] Julian Hagenauer and Marco Helbich. "A comparative study of machine learning classifiers for modeling travel mode choice". In: *Expert Systems with Applications* 78 (2017), pp. 273–282.

[29] Yafei Han et al. "A neural-embedded discrete choice model: Learning taste representation with strengthened interpretability". In: *Transportation Research Part B: Methodological* 163 (2022), pp. 166–186. ISSN: 0191-2615.

[30] Trevor Hastie, Andreas Buja, and Robert Tibshirani. "Penalized discriminant analysis". In: *The Annals of Statistics* (1995), pp. 73–102.

[31] Trevor Hastie and Robert Tibshirani. "Discriminant analysis by Gaussian mixtures". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 58.1 (1996), pp. 155–176.

[32]   David A Hensher and Tu T Ton. "A comparison of the predictive potential of artificial neural networks and nested logit models for commuter mode choice". In: *Transportation Research Part E: Logistics and Transportation Review* 36.3 (2000), pp. 155–172.

[33]   Tim Hillel, Mohammed ZEB Elshafie, and Ying Jin. "Recreating passenger mode choice-sets for transport simulation: A case study of London, UK". In: *Proceedings of the Institution of Civil Engineers-Smart Infrastructure and Construction* 171.1 (2018), pp. 29–42. ISSN: 2397-8759.

[34]   Tim Hillel et al. "A new framework for assessing classification algorithms for mode choice prediction". In: *hEART 2018, 7th Symposium of the European Association for Research in Transportation*.

[35]   Johan Holmgren. "Meta-analysis of public transport demand". In: *Transportation Research Part A: Policy and Practice* 41.10 (2007), pp. 1021–1035. ISSN: 0965-8564.

[36]   Kurt Hornik, Maxwell Stinchcombe, and Halbert White. "Multilayer feedforward networks are universal approximators". In: *Neural networks* 2.5 (1989), pp. 359–366.

[37]   Torsten Hothorn, Kurt Hornik, and Achim Zeileis. "Unbiased recursive partitioning: A conditional inference framework". In: *Journal of Computational and Graphical statistics* 15.3 (2006), pp. 651–674.

[38]   Geoff Hulten, Laurie Spencer, and Pedro Domingos. "Mining time-changing data streams". In: *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 2001, pp. 97–106.

[39]   Arash Jahangiri and Hesham A Rakha. "Applying machine learning techniques to transportation mode recognition using mobile phone sensor data". In: *IEEE transactions on intelligent transportation systems* 16.5 (2015), pp. 2406–2417. ISSN: 1524-9050.

[40]   Matthew G Karlaftis and Eleni I Vlahogianni. "Statistical methods versus neural networks in transportation research: Differences, similarities and some insights". In: *Transportation Research Part C: Emerging Technologies* 19.3 (2011), pp. 387–399.

[41]   Ron Kohavi. "The power of decision tables". In: *European conference on machine learning.* Springer, 1995, pp. 174–189.

[42]   Niels Landwehr, Mark Hall, and Eibe Frank. "Logistic model trees". In: *Machine learning* 59.1-2 (2005), pp. 161–205.

[43]   Saskia Le Cessie and Johannes C Van Houwelingen. "Ridge estimators in logistic regression". In: *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 41.1 (1992), pp. 191–201.

[44]   Zachary C Lipton. "The mythos of model interpretability". In: *arXiv preprint arXiv:1606.03490* (2016).

[45]   Jordan J Louviere, David A Hensher, and Joffre D Swait. *Stated choice methods: analysis and applications.* Cambridge university press, 2000.

[46] Patricia C Melo, Daniel J Graham, and Robert B Noland. "A meta-analysis of estimates of urban agglomeration economies". In: *Regional science and urban Economics* 39.3 (2009), pp. 332–342. ISSN: 0166-0462.

[47] Baichuan Mo et al. *Impacts of subjective evaluations and use of existing travel modes on adoption of autonomous mobility-on-demand.* Tech. rep. 2020.

[48] Abolfazl Mohammadian and Eric J Miller. "Nested logit models and artificial neural networks for predicting household automobile choices: comparison of performance". In: *Transportation Research Record* 1807.1 (2002), pp. 92–100. ISSN: 0361-1981.

[49] Mikhail Mozolin, J-C Thill, and E Lynn Usery. "Trip distribution forecasting with multi-layer perceptron neural networks: A critical evaluation". In: *Transportation Research Part B: Methodological* 34.1 (2000), pp. 53–73.

[50] Peter Nijkamp, Aura Reggiani, and Tommaso Tritapepe. "Modelling inter-urban transport flows in Italy: A comparison between neural network analysis and logit analysis". In: *Transportation Research Part C: Emerging Technologies* 4.6 (1996), pp. 323–338.

[51] Hichem Omrani. "Predicting travel mode of individuals by machine learning". In: *Transportation Research Procedia* 10 (2015), pp. 840–849.

[52] Hichem Omrani et al. "Prediction of individual travel mode with evidential neural network model". In: *Transportation research record* 2399.1 (2013), pp. 1–8. ISSN: 0361-1981.

[53] Miguel Paredes et al. "Machine learning or discrete choice models for car ownership demand estimation and prediction?" In: *Models and Technologies for Intelligent Transportation Systems (MT-ITS), 2017 5th IEEE International Conference on.* IEEE, 2017, pp. 780–785.

[54] Nicholas G Polson and Vadim O Sokolov. "Deep learning for short-term traffic flow prediction". In: *Transportation Research Part C: Emerging Technologies* 79 (2017), pp. 1–17.

[55] Ross Quinlan. "C4. 5". In: *Programs for machine learning* (1993).

[56] PV Subba Rao et al. "Another insight into artificial neural networks through behavioural analysis of access mode choice". In: *Computers, environment and urban systems* 22.5 (1998), pp. 485–496.

[57] Brian D Ripley and NL Hjort. *Pattern recognition and neural networks.* Cambridge university press, 1996.

[58] Tarek Sayed and Abdolmehdi Razavi. "Comparison of neural and conventional approaches to mode choice analysis". In: *Journal of Computing in Civil Engineering* 14.1 (2000), pp. 23–30. ISSN: 0887-3801.

[59] Ch Ravi Sekhar and E Madhu. "Mode Choice Analysis Using Random Forrest Decision Trees". In: *Transportation Research Procedia* 17 (2016), pp. 644–652.

[60] Muhammad Awais Shafique and Eiji Hato. "Use of acceleration data for transportation mode prediction". In: *Transportation* 42.1 (2015), pp. 163–188. ISSN: 0049-4488.

[61]   Yu Shen et al. *Built environment and autonomous vehicle mode choice: A first-mile scenario in singapore.* Tech. rep. 2019.

[62]   Nagesh Shukla et al. "Tour-based travel mode choice estimation based on data mining and fuzzy techniques". In: (2015).

[63]   Brian Sifringer, Virginie Lurkin, and Alexandre Alahi. "Enhancing discrete choice models with representation learning". In: *Transportation Research Part B: Methodological* 140 (2020), pp. 236–261. ISSN: 0191-2615.

[64]   Kenneth A Small, Erik T Verhoef, and Robin Lindsey. "Travel Demand". In: *The economics of urban transportation.* Vol. 2. Routledge, 2007.

[65]   Nitish Srivastava et al. "Dropout: a simple way to prevent neural networks from overfitting". In: *Journal of machine learning research* 15.1 (2014), pp. 1929–1958.

[66]   Liang Tang, Chenfeng Xiong, and Lei Zhang. "Decision tree method for modeling travel mode switching in a dynamic behavioral process". In: *Transportation Planning and Technology* 38.8 (2015), pp. 833–850.

[67]   Alison Thornton and Peter Lee. "Publication bias in meta-analysis: its causes and consequences". In: *Journal of clinical epidemiology* 53.2 (2000), pp. 207–216. ISSN: 0895-4356.

[68]   Ahmet Tortum, Nadir Yayla, and Mahir G kda. "The modeling of mode choices of intercity freight transportation with the artificial neural networks and adaptive neuro-fuzzy inference system". In: *Expert Systems with Applications* 36.3 (2009), pp. 6199–6217. ISSN: 0957-4174.

[69]   Jacques Wainer. "A Bayesian Bradley-Terry model to compare multiple ML algorithms on multiple data sets". In: *Journal of Machine Learning Research* 24.341 (2023), pp. 1–34. ISSN: 1533-7928.

[70]   Fangru Wang and Catherine L Ross. "Machine learning travel mode choices: Comparing the performance of an extreme gradient boosting model with a multinomial logit model". In: *Transportation Research Record* 2672.47 (2018), pp. 35–45.

[71]   Qingyi Wang et al. "Deep hybrid model with satellite imagery: how to combine demand modeling and computer vision for behavior analysis?" In: *arXiv preprint arXiv:2303.04204* (2023).

[72]   Shenhao Wang, Baichuan Mo, and Jinhua Zhao. "Deep neural networks for choice analysis: Architecture design with alternative-specific utility functions". In: *Transportation Research Part C: Emerging Technologies* 112 (2020), pp. 234–251. ISSN: 0968-090X.

[73]   Shenhao Wang, Qingyi Wang, and Jinhua Zhao. "Deep neural networks for choice analysis: Extracting complete economic information for interpretation". In: *Transportation Research Part C: Emerging Technologies* 118 (2020), p. 102701. ISSN: 0968-090X.

[74]   Shenhao Wang, Qingyi Wang, and Jinhua Zhao. "Multitask learning deep neural networks to combine revealed and stated preference data". In: *Journal of Choice Modelling* (2020), p. 100236. ISSN: 1755-5345.

[75] Shenhao Wang and Jinhua Zhao. "Risk preference and adoption of autonomous vehicles". In: *Transportation Research Part A: Policy and Practice* 126 (2019), pp. 215–229. ISSN: 0965-8564.

[76] Shenhao Wang et al. "Deep neural networks for choice analysis: A statistical learning theory perspective". In: *Transportation Research Part B: Methodological* 148 (2021), pp. 60–81. ISSN: 0191-2615.

[77] Melvin Wong and Bilal Farooq. "ResLogit: A residual neural network logit model for data-driven choice modelling". In: *Transportation Research Part C: Emerging Technologies* 126 (2021), p. 103050. ISSN: 0968-090X.

[78] Yuankai Wu et al. "A hybrid deep learning based traffic flow prediction method and its understanding". In: *Transportation Research Part C: Emerging Technologies* 90 (2018), pp. 166–180.

[79] Jian-Chuan Xian-Yu. "Travel mode choice analysis using support vector machines". In: *ICCTP 2011: Towards Sustainable Transportation Systems*. 2011, pp. 360–371.

[80] Chi Xie, Jinyang Lu, and Emily Parkany. "Work travel mode choice modeling with data mining: decision trees and neural networks". In: *Transportation Research Record: Journal of the Transportation Research Board* 1854 (2003), pp. 50–61.

[81] Hong Zhang and Zhen Zhang. "Feedforward networks with monotone constraints". In: *IJCNN'99. International Joint Conference on Neural Networks. Proceedings (Cat. No. 99CH36339)*. Vol. 3. IEEE, 1999, pp. 1820–1823.

[82] Yunlong Zhang and Yuanchang Xie. "Travel mode choice modeling with support vector machines". In: *Transportation Research Record: Journal of the Transportation Research Board* 2076 (2008), pp. 141–150.

[83] Xiaolu Zhou, Mingshu Wang, and Dongying Li. "Bike-sharing or taxi? Modeling the choices of travel mode in Chicago using machine learning". In: *Journal of transport geography* 79 (2019), p. 102479. ISSN: 0966-6923.

Appendix I: Summary of Acronyms and Terminology

Table 6: Acronyms and other terminology in this study;

Panel 1. Acronyms	
DCM	Discrete choice models
DNN	Deep neural networks
DA	Discriminant analysis
BM	Bayesian models
SVM	Support vector machines
KNN	K nearest neighbors
DT	Decision tree
GLM	Generalized linear models
GP	Gaussian process
BAGGING	Bagging methods
RF	Random forests
BOOSTING	Boosting methods
MNL	Multinomial logit
NL	Nested logit
MXL	Mixed logit
NHTS2017	National household travel survey 2017
LTDS2015	London travel demand survey 2015
SGP2017	Singapore stated preference survey 2017
MC	Mode choice
CO	Car ownership
TP	Trip purposes
Panel 2. Other terminology	
Classifiers	Models using discrete values as outputs
Hyperparameters	A parameter used to control the learning process
0-1 loss	An indicator function that returns 1 when the target and output are not equal and zero otherwise

Appendix II: Further Implementation Details

In our experiments, the inputs were selected and normalized before modeling. In the NHTS2017 data, the most important 50 features were chosen from the initial 115 by using χ^2 feature selection, which examines the dependence between inputs and outputs with condensed information. In the LTDS2015 data, the best 40 features were chosen from the initial 48 features using the same method. The SGP2017 data has only 22 independent variables, so it is unnecessary to limit the number of input features. After the feature selection, all the input variables were normalized to Gaussian distributions centered on zero with unit variance.

The number of experiments is smaller than the theoretically maximum, because of the limitations on data availability, data storage, and computational difficulty. Although the theoretical maximum of experiment points is $105 \times 3 \times 3 \times 3 = 2,835$, our experiments only examined 1,394 experiment points because of computational difficulties. Specifically, trip purposes and car ownership are not available in the LTDS2015 and SGP2017 datasets. For the three GP classifiers, the complexity for storage space is $\mathcal{O}(N^2)$, where N is the sample size, so the GP classifiers with larger than 1K sample sizes can lead to memory errors in a personal computer (PC) with 32G RAM. The MXL model can only be implemented for the 1K sample size, because its training time can exceed 10 hours when the sample size reaches 10K. Five-fold cross-validation was used to compute the average prediction accuracy in the testing sets for each classifier, leading to a total number of $1,394 \times 5 = 6,970$ experiments. For each experiment, the data was split into the training and testing sets with a ratio of $4:1$.

Appendix III: Linear Model Results

Linear regressions can also be used with prediction accuracy as outputs, and model and contextual factors as inputs. The results are summarized in Table 7. The results demonstrate similar conclusions from our tournament model. The ML models outperform DCMs in both literature and experiment data, although the literature typically have higher confidence in the outperformance than the experiment data. The contextual factors and inherent randomness still dominate, as shown by the statistical significance and the high explanatory power of the contextual factors, and the large residual randomness. Due to the high similarity to the tournament model, we skip detailed discussions here.

Table 7: Results from Linear Models

Parameters	Linear Models			
	Literature (NC)	Literature (C)	Experiment (NC)	Experiment (C)
Constant	0.701 (0.020) ***	0.915 (0.034) ***	0.545 (0.017) ***	0.165 (0.003) ***
Model family effects				
BAGGING	0.205 (0.110)	0.267 (0.094) **	0.010 (0.023)	0.006 (0.014)
BM	0.000 (0.066)	0.037 (0.054)	-0.068 (0.020) ***	-0.071 (0.012) ***
BOOSTING	0.139 (0.034) ***	0.124 (0.029) ***	0.026 (0.018)	0.022 (0.011) *
DA	NA	NA	-0.004 (0.019)	-0.005 (0.011)
DNN	0.063 (0.024) *	0.095 (0.021) ***	0.033 (0.018)	0.029 (0.011) **
DT	0.150 (0.030) ***	0.102 (0.026) ***	0.009 (0.018)	0.005 (0.011)
GLM	-0.221 (0.110) *	-0.022 (0.095)	0.008 (0.019)	0.004 (0.012)
GP	NA	NA	-0.061 (0.031)	-0.052 (0.019) **
KNN	0.069 (0.110)	0.100 (0.090)	-0.061 (0.022) **	-0.065 (0.013) ***
RF	0.176 (0.037) ***	0.160 (0.032) ***	0.026 (0.026)	0.022 (0.016)
SVM	0.130 (0.031) ***	0.093 (0.026) ***	-0.080 (0.020) ***	-0.081 (0.012) ***
Contextual effects				
If_Asian		-0.046 (0.022) *		0.190 (0.004) ***
If_AM		-0.185 (0.042) ***		NA
If_Europe		-0.126 (0.022) ***		0.296 (0.004) ***
If_CO		-0.105 (0.035) **		0.116 (0.004) ***
if_TP		-0.088 (0.031) **		0.084 (0.004) ***
Input dimention		-0.004 (0.001) **		0.004 (0.000) ***
Num of alternatives		-0.019 (0.006) **		0.012 (0.002) ***
Sample size		-0.347 (1.690)		3.302 (0.407) ***
Statistical summary				
Num of Observations	178	178	1388	1388
AIC	-243.879	-315.761	-2391.3	-3762.7
R^2	0.23	0.53	0.11	0.671
Adjusted R^2	0.178	0.473	0.103	0.667

Appendix IV: Collecting and Describing Literature Data

The process of literature data collection could influence our results. To enhance the replicability of this study, the authors present the detailed data collection process as follows. We identified the relevant studies by combining the results from Google, Google Scholar, Web of Science, and Scopus, thus overcoming the limit of the searching scope in an individual engine. We combined the methodological and topical keywords in our searching: the methodological keywords included "prediction", "machine learning", "discrete choice models", and "deep learning", and the topical key words include "travel behavior", "travel demand", "choice analysis", "travel mode choice", amongst others. Although such searching often yields a large number of results, the authors had to filter out an overwhelmingly large number of irrelevant studies, which typically adopted spatiotemporal methods for aggregate demand modeling, different from the cross-sectional data sets for individual behavioral analysis[5]. With the initial set of studies, we iterated through snowball sampling by searching the related studies in their sections of "Related Work", "Literature Review", "Bibliography". Such iterations were repeated until no new relevant paper can be identified. The online search leads to 35 studies that compared the predictive performance between classical discrete choice models and machine learning classifiers.

The scope of the literature data is limited to only those studies that compared machine learning and discrete choice models for individual travel behaviors in a cross-sectional context. As a result, three groups of studies were excluded in the searching process above. The first group are those using complex data structures different from classical cross-sectional data, such as the spatial, temporal, or spatiotemporal data structures for urban mobility. The second group are those that tackle aggregate travel decisions rather than the disaggregate individual travel decisions. The third group uses either only machine learning for prediction or only discrete choice models for travel behavioral analysis without comparing their performance. In fact, the three groups constitute a vast volume of studies, which are beyond the scope of this empirical benchmark work.

Figures 6 and 7 visualize the histogram of the best models and the prediction accuracy of the models in the literature data. The results reinforce our findings that the ensemble methods and the DNNs have the highest predictive performance, while the DCMs rank at the medium or lower ends of the predictive distributions. As shown in Figure 6, 16 out of the 35 studies found that DNNs achieve the highest performance and 11 of them found that the ensemble methods (RF, BOOSTING, and BAGGING) outperform others. The ensemble methods achieve the highest prediction accuracy, and the accuracy of the three ensemble families can be as high as nearly 99.86%, as shown in Figure 7. DNNs' average prediction accuracy is around 76.47%, which can be as high as 98.4%. The MNL and NL achieve about 72.49% and 68.53% on average, which are about 4-8 percentage points lower than the DNNs and about 15-20 percentage points lower than the ensemble methods. However, similar to the experiment data, the performance of the model families also exhibit quite significant randomness, as shown by the large variation of model performance in each model family. Overall,

---

[5]In the literature searching, the authors did not iterate over different spellings, e.g., "behavior" and "behaviour", thus potentially missing some relevant studies.

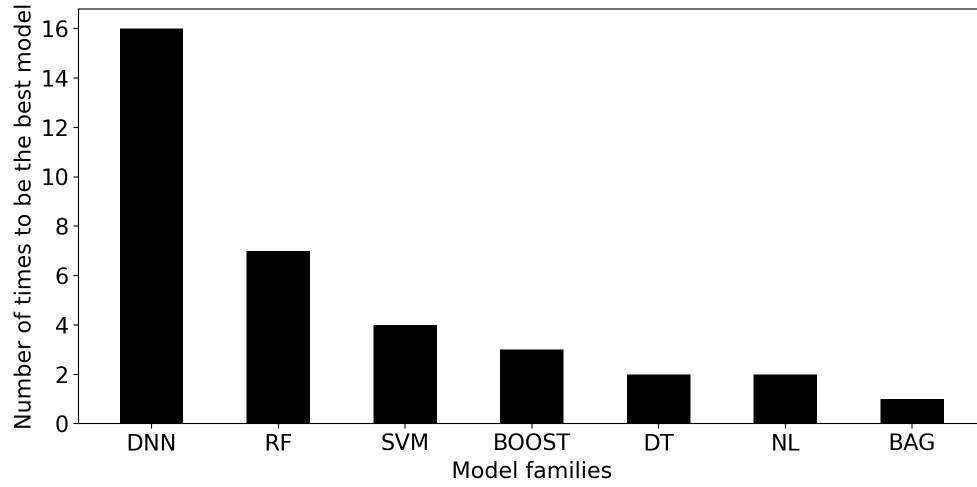the descriptive analysis reveals similar patterns between the literature and experiment data.
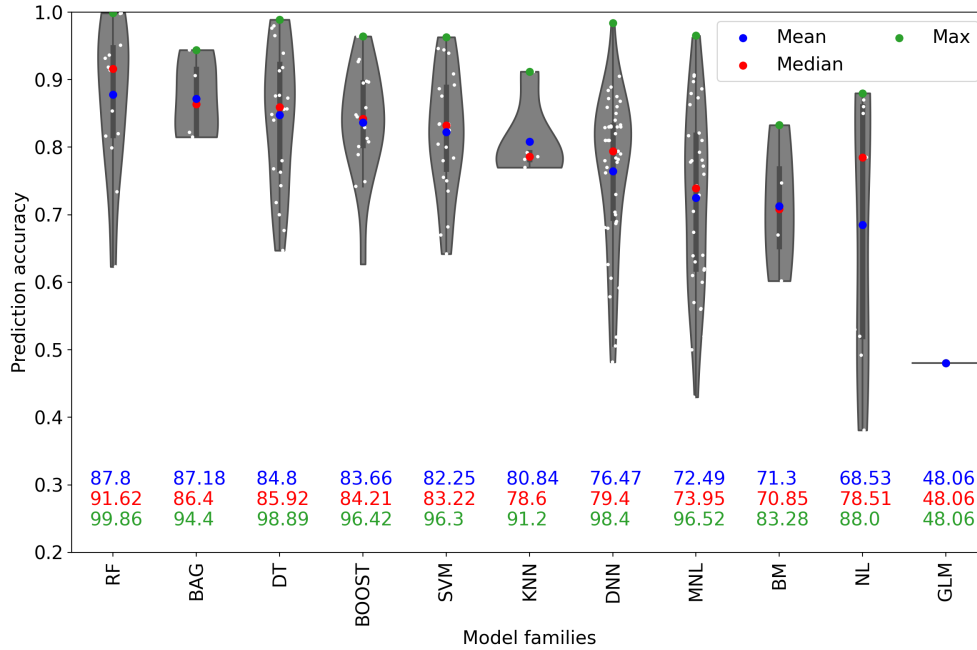

Fig. 6. Histogram of the best models

Fig. 7. Prediction accuracy of the examined models

Appendix V: Comparing DCMs and DNNs using Log-likelihood and Brier Score

To complement the findings using only the deterministic metrics, we offer a further comparison using probabilistic metrics, including log-likelihood and Brier scores[6], for the DCMs and DNNs. Table 8 summarizes the average scores across all the prediction tasks and data sets, with the best model colored in gray. Due to the computational limitations for MXL, only the data sets with 1K and 10K observations are presented. Our findings here are consistent with those in the manuscript. The models with the highest performance are typically the DNNs, regardless of the sample size or the probabilistic metric. Although the DNNs on average tend to underperform the DCMs when the sample size is small, they outperform the DCMs when the sample size reaches 10K, implying the importance of a relatively large sample size for complex model families such as DNNs. The variation of performance in the DCM family is quite limited, while that in the DNN family is much larger. This finding is also consistent with that in Figure 2.

Table 8: Comparing DNNs and DCMs using Log-likelihood and Brier score

	Log-likelihood		Brier score	
	1k	10k	1k	10k
mnl_B	-218.62	-1953.65	0.1211	0.1166
nl_B	-223.72	-1998.27	0.1192	0.1129
mxl_B	-226.58	N.A.	0.1245	N.A.
DCM (all)	-223.03	-1978.44	0.1214	0.1145
DNN_30_1_P	-208.02	-1975.75	0.1167	0.1116
DNN_30_3_P	-226.88	-1895.53	0.1189	0.1073
DNN_30_5_P	-259.98	-1946.90	0.1286	0.1096
DNN_100_1_P	-217.87	-1947.88	0.1165	0.1104
DNN_100_3_P	-256.64	-1840.05	0.1271	0.1046
DNN_100_5_P	-356.07	-1843.43	0.1454	0.1048
DNN_200_1_P	-216.54	-1839.00	0.1150	0.1111
DNN_200_3_P	-264.94	-1822.61	0.1280	0.1037
DNN_200_5_P	-391.92	-1835.39	0.1490	0.1045
DNN (all)	-267.875	-1883.95	0.1276	0.1074

---

[6]Brier score is computed as $\frac{1}{N}\sum_{i=1}^{N}(y_i - P_i)^2$

Appendix VI: Investigating Input Dimension and Performance

Here we investigate how input dimensions influence model performance. Figure 8 visualizes the relationship between prediction accuracy and input dimensions using all the literature data (Figure 8a) and those with input dimensions higher than five (Figure 8b). In Figure 8a, prediction accuracy decreases with input dimension, consistent with the tournament model in the manuscript. However, such a result could be influenced by the potential outliers. Therefore in Figure 8b, after removing the outliers with input dimensions lower than five, the relationship becomes flat, indicating only a weak correlation between accuracy and input dimensions. After running a regression, we obtain an estimated function as $y_i = \beta_0 + \beta_1 x_i$, in which $\beta_1 = -0.0024$ with p value equals to 0.22, implying an insignificant statistical relationship between accuracy and input dimensions after the outliers are removed.

We further track the outlier articles to understand how such studies achieve very high performance using a limited number of features [60, 66]. It is because these studies investigate research questions different from the common discrete choice modeling that uses either stated or revealed preference data. For example, Shafique (2015) infers travel mode choice using observed speed data, thus leading to high predictive performance [60]. However, since such studies predict travel behavior by comparing the ML and DCMs, they still fall into our research scope. Nonetheless, we acknowledge that the high heterogeneity and insufficient contextual controls in our two data sets can lead to unstable results, which shall be tackled by future studies.

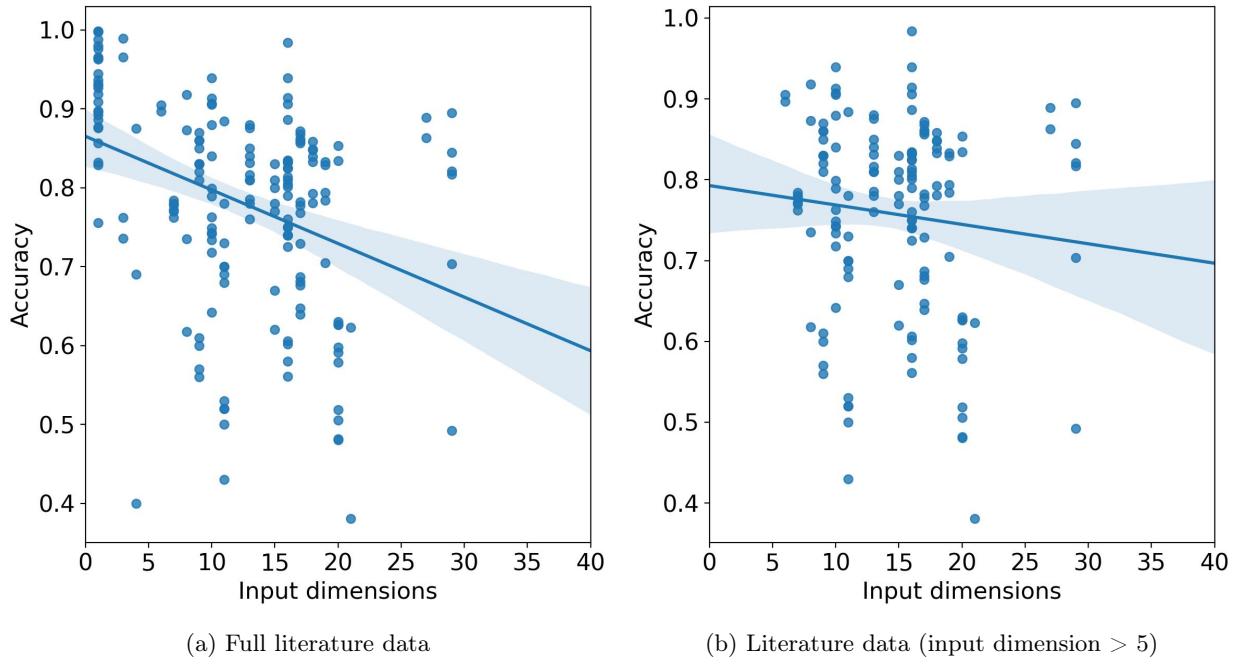(a) Full literature data   (b) Literature data (input dimension > 5)

Fig. 8. Input dimension and prediction accuracy for the literature data