



Deposited via The University of Leeds.

White Rose Research Online URL for this paper:

<https://eprints.whiterose.ac.uk/id/eprint/218216/>

Version: Accepted Version

Article:

Feng, S., Yao, R., Hess, S. et al. (2024) Deep neural networks for choice analysis: Enhancing behavioral regularity with gradient regularization. Transportation Research Part C Emerging Technologies, 166. 104767. ISSN: 0968-090X

<https://doi.org/10.1016/j.trc.2024.104767>

This is an author produced version of an article published in Transportation Research Part C: Emerging Technologies, made available under the terms of the Creative Commons Attribution License (CC BY), which permits unrestricted use, distribution and reproduction in any medium, provided the original work is properly cited.

Reuse

This article is distributed under the terms of the Creative Commons Attribution (CC BY) licence. This licence allows you to distribute, remix, tweak, and build upon the work, even commercially, as long as you credit the authors for the original work. More information and the full terms of the licence here:

<https://creativecommons.org/licenses/>

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

Deep neural networks for choice analysis: Enhancing behavioral regularity with gradient regularization

Siqi Feng^a, Rui Yao^b, Stephane Hess^c, Ricardo A. Daziano^a, Timothy Brathwaite^d, Joan Walker^d, and Shenhao Wang^{*e}

^aSchool of Civil and Environmental Engineering, Cornell University

^bSchool of Architecture, Civil and Environmental Engineering, École Polytechnique Fédérale de Lausanne

^cInstitute for Transport Studies, University of Leeds

^dDepartment of Civil and Environmental Engineering, University of California, Berkeley

^eDepartment of Urban and Regional Planning, University of Florida

Abstract

Deep neural networks (DNNs) have been increasingly applied in travel demand modeling because of their automatic feature learning, high predictive performance, and economic interpretability. Nevertheless, DNNs frequently present behaviorally irregular patterns, significantly limiting their practical potentials and theoretical validity in travel behavior modeling. This study proposes strong and weak behavioral regularities as novel metrics to evaluate the monotonicity of individual demand functions (known as the “law of demand”), and further designs a constrained optimization framework with six gradient regularizers to enhance DNNs’ behavioral regularity. The empirical benefits of this framework are illustrated by applying these regularizers to travel survey data from Chicago and London, which enables us to examine the trade-off between predictive power and behavioral regularity for large versus small sample scenarios and in-domain versus out-of-domain generalizations. The results demonstrate that, unlike models with strong behavioral foundations such as the multinomial logit, the benchmark DNNs cannot guarantee behavioral regularity. However, after applying gradient regularization, we increase DNNs’ behavioral regularity by around 6 percentage points while retaining their relatively high predictive power. In the small sample scenario, gradient regularization is more effective than in the large sample scenario, simultaneously improving behavioral regularity by about 20 percentage points and log-likelihood by around 1.7%. Compared with the in-domain generalization of DNNs, gradient regularization works more effectively in out-of-domain generalization: it drastically improves the behavioral regularity of poorly performing benchmark DNNs by around 65 percentage points, highlighting the criticality of behavioral regularization for improving model transferability and applications in forecasting. Moreover, the proposed optimization framework is applicable to other neural network-based choice models such as TasteNets. Future studies could use behavioral regularity as a metric along with log-likelihood, prediction accuracy, and F_1 score when evaluating travel demand models, and investigate other methods to further enhance behavioral regularity when adopting complex machine learning models.

Keywords: travel demand, deep learning, behavioral regularization

*Corresponding author. E-mail: shenhaowang@ufl.edu.

1 Introduction

Deep neural networks (DNNs) have revolutionized fields such as computer vision and natural language processing, which in turn support technologies such as self-driving cars and large language models (van Dis et al., 2023; LeCun et al., 2015). DNNs have also been applied in economics (Zheng et al., 2023) to interpret and predict individual choice behavior (Wang et al., 2020a,b). It is in this latter area that DNNs offer a contrast with the conventionally used discrete choice models (DCMs), which are typically based on random utility maximization and the assumption that travelers choose the alternative with the highest random utility in the choice set (Ben-Akiva and Lerman, 1985). One drawback of this traditional modeling paradigm is the time-consuming trial-and-error process for an “optimal” specification of the model, in particular the utility function that represents economic preferences (van Cranenburgh et al., 2022). Additionally, the decisions made in this process are often subjective. By contrast, DNNs are capable of automatic feature learning, i.e., the specification of a DNN-based choice model is automatically learned from the input data, which avoids the specification search process and reduces the level of subjectivity. The high prediction accuracy of DNNs is a result of their complex model structure, which can capture intricate behavioral relationships and provide new insights beyond those of conventional DCMs.

Traditional choice modelers often see DNNs as “black-box” models, although DNNs actually contain complete economic information for choice analysis (Wang et al., 2020b). However, existing DNNs often exhibit behaviorally irregular patterns because the demand functions in DNNs are not guaranteed to decrease monotonically with generalized costs. The “law of demand” in economics indicates an inverse relationship between generalized costs and the aggregate demand. While DCMs such as random utility models (RUMs) do not impose specific prior directionality, the specification search conducted by an analyst would prevent models that lead to counter-intuitive results. However, non-monotonic patterns have been detected empirically in DNNs’ predictions, even with model ensembles (Wang et al., 2020a,b; Xia et al., 2023), as the analyst has less control. This fact might be a drawback of the nonlinear structure of DNNs, allowing them to flexibly fit the data but making it challenging to constrain the gradient’s direction with limited data samples. The issue often deteriorates in out-of-domain applications, i.e., applying a trained DNN to a test set with unseen distributions (Quiñonero-Candela et al., 2008). In fact, the out-of-domain generalizability of DNNs has attracted rising interests in several computer science subfields, including domain adaption (Wang and Deng, 2018) and transfer learning (Pan and Yang, 2009).

To improve the monotonicity of DNNs, we propose to regularize the loss function in training, which has been shown in computer science to enhance the robustness of DNNs (Lyu et al., 2015; Ross and Doshi-Velez, 2018). Nevertheless, this approach has rarely been considered in previous DNN-based choice models, or used for enhancing behavioral regularity (cf. Wang et al., 2020b; Zheng et al., 2021). In this paper, we address the issue of behavioral irregularity by first defining strong and weak behavioral regularity metrics based on the monotonicity of demand functions, and further designing and implementing a constrained optimization framework that regularizes the input gradient in order to explicitly constrain the gradient’s direction. We then design experiments to examine the performance of behaviorally regularized DNNs in terms of behavioral regularity and predictive performance, differentiating between in-domain and out-of-domain generalizations. We also consider the effect of sample size, which is crucial in practice because large samples are costly for travel surveys. Based on two travel survey datasets from Chicago and London, our experiments compare across the standard DNN and TasteNet (Han et al., 2022) architectures using five evaluation metrics, including log-likelihood, prediction accuracy, F_1 score, strong behavioral regularity, and weak behavioral regularity. The multinomial logit (MNL) is chosen as a benchmark model due to its concise expression and high behavioral regularity. For TasteNets, we also compare the performance of our gradient regularizers (as soft constraints) and the hard constraints applied in Han et al. (2022). The results show that by using appropriate gradient regularization, both DNNs and TasteNets can achieve high regularity without sacrificing their predictive power, which makes these models competitive in real-world applications and demonstrates the generality of our gradient regularization framework.

The rest of this paper is organized as follows. [Section 2](#) briefly reviews the literature about the behavioral irregularity issue of DNNs with possible solutions. [Section 3](#) introduces the theory, formulates the problem, and develops a solution framework based on gradient regularization. [Section 4](#) sets up the mode choice experiment, while [Section 5](#) illustrates and analyzes the empirical results. Finally, [Section 6](#) concludes the study and looks ahead to future research. To facilitate future research, we uploaded this work to the following GitHub repository: <https://github.com/siqi-feng/DNN-behavioral-regularity>.

2 Literature review

The economic choice behavior of humans generally follows the law of demand in economics, which states the inverse relationship between price and quantity demanded. This indicates a monotonic change in market demand due to the change in consumers’ purchasing power, including price and income changes ([Chiappori, 1985](#); [Härdle et al., 1991](#); [Hildenbrand, 1983](#); [Quah, 2000](#)). The transportation field has also observed the negative influence of travel costs on travel demand ([McFadden, 1974](#); [Souche, 2010](#); [Yao and Morikawa, 2005](#)), based on which demand management policies such as road pricing ([May, 1992](#); [Yang and Bell, 1997](#)) were developed. Although such market rationality is widely recognized, individual choice behavior might be irrational ([Becker, 1962](#); [Knez et al., 1985](#)). [Lichtenstein and Slovic \(1971\)](#) studied preference reversal in decision making, which is a typical counterexample of individual rationality. Studies in bounded rationality theory ([Simon, 1957](#); [Di and Liu, 2016](#); [Watling et al., 2018](#)) and prospect theory ([Kahneman and Tversky, 1979](#); [Tversky and Kahneman, 1992](#)) also relax the strict monotonicity assumption in demand modeling.

The law of demand is generally followed by the design of random utility models. In the MNL model, for example, an increase in the travel cost of an alternative would be expected to decrease its systematic utility, thus decreasing its choice probability by design. If the initial model includes estimates with counter-intuitive signs, such as positive cost estimates, an analyst can readily spot them and then refine the model specification or deal with data issues. Once all signs are as expected, the monotonic relationship is guaranteed. By contrast, this is not the case in DNNs because of the complex nonlinear model structure, especially when the number of hidden layers increases. For example, [Xia et al. \(2023\)](#) observed non-monotonic demand predictions with increasing generalized costs in a mode choice experiment with DNNs, which suggests the need to investigate the monotonicity of DNNs to improve their behavioral regularity. Although shallow neural networks (NNs) might reduce the risk of non-monotonic behavior of DNNs ([Alwosheel et al., 2019](#); [Zhao et al., 2020](#)), this might come at the cost of reduced modeling flexibility and universal approximation power. Alternatively, [Han et al. \(2022\)](#) and [Sifringer et al. \(2020\)](#) proposed to use DNNs only for learning latent representation in the utility function, while resorting to the DCM framework to ensure model monotonicity. For example, TasteNet, the neural-embedded DCM proposed by [Han et al. \(2022\)](#), assumes a linear model specification with parameters learned by an NN. In this paper, TasteNets are considered as a reference to the standard DNN architecture. On the other hand, depending on model design, these hybrid DNN models might still produce irregular predictions ([Wang et al., 2021](#); [Wong and Farooq, 2021](#)). Moreover, hybrid DNN models are a compromise for regularity since they again require the subjective process of model specification. To fully utilize the capability of DNNs, previous studies have attempted to migrate the non-monotonic issue through model ensemble. However, irregular patterns might still be observed after averaging over multiple trainings ([Wang et al., 2020a,b](#); [Xia et al., 2023](#)). One promising direction is to directly integrate domain-specific knowledge into the design and training of DNNs, such as incorporating monotonicity constraints in model training ([Haj-Yahia et al., 2023](#)). Nevertheless, there is no consensus on how to measure or improve the model regularity of DNNs within the choice modeling field. This paper contributes to the development of a behavioral regularity measure and a novel regularization framework.

As discussed in computer science applications, the regularity of DNNs can be improved by employing either hard or soft constraints. The first category enforces monotonicity by model construction, e.g., constraining the positiveness of weights in hidden layers ([Daniels and Velikova, 2010](#); [Dugas et al., 2009](#); [Sill, 1997](#)) through non-negativity constraints ([Lawson and Hanson, 1995](#)), restricting the derivative to be posi-

tive (Neumann et al., 2013; Wehenkel and Louppe, 2019), down-weighting samples that violate monotonicity (Archer and Wang, 1993), and incorporating deep lattice networks for learning monotonic functions (You et al., 2017). The second category achieves monotonicity by regularization, i.e., by augmenting a regularization term in the loss function to jointly improve model monotonicity. In the choice modeling domain, recent studies have also incorporated hard constraints on model monotonicity based on the modeler’s prior assumptions to improve interpretability, such as in lattice networks (Kim and Bansal, 2024) and TasteNets (Han et al., 2022). Regularization is firmly rooted in constrained optimization, including the Lagrangian method as an example (Boyd and Vandenberghe, 2004). It has been widely applied as a local method to penalize constraint violations. For example, Sill and Abu-Mostafa (1996) penalized squared deviations in monotonicity for virtual pairs of input variables, while Gupta et al. (2019) proposed a pointwise loss that embeds prior knowledge about monotonicity. Moreover, gradient regularization has also been used to enhance model robustness against adversarial examples (Lyu et al., 2015; Ross and Doshi-Velez, 2018), e.g., penalizing the squared L_2 norm of the gradient of the loss with respect to (w.r.t.) inputs (Drucker and Le Cun, 1991; Ororbia II et al., 2017), penalizing the squared Frobenius norm of the Jacobian matrix of probabilities (Sokolić et al., 2017) and utilities (Jakubovitz and Giryas, 2018) w.r.t. inputs. Note that regularizing the gradient norms might be less effective to improve monotonicity than regularizing the gradient’s direction like Haj-Yahia et al. (2023). Inspired by the aforementioned regularization methods, we will design our own approaches in the demand modeling context.

3 Methodology

3.1 DNNs for choice analysis

The discrete choice problem is cast as a supervised classification task in DNN-based choice analysis. Assuming there are in total D explanatory variables $(x_1, \dots, x_D)$ for all alternatives, the attribute vector of individual n can be written as $\mathbf{x}_n = [x_{n1}, \dots, x_{nD}]^\top$. Then, a DNN model predicts the probability of n choosing i out of J alternatives, i.e., $P_{ni} : \mathbb{R}^D \rightarrow (0, 1)$ and $\sum_{i=1}^J P_{ni} = 1$. The observed choice vector $\mathbf{y}_n \in \{0, 1\}^J$ of n is used for DNN training, where $y_{ni} = 1$ if alternative i is chosen, and $y_{ni} = 0$ otherwise. Similar to conventional RUMs, DNN models aim to find specifications with high predictive power and behavioral regularity.

In contrast to RUMs with handcrafted specifications, DNNs automatically learn specifications through their unique representation learning capability. Particularly, utility vector $\mathbf{V}_n = [V_{n1}, \dots, V_{nJ}]^\top$ is specified through a series of transformations, termed as layers $(f_1, \dots, f_H)$, where H denote the total number of layers in a DNN. Each layer f_h contains a learnable parameter matrix W_h , a bias vector $\mathbf{b}_h$, and an activation function $\varphi(\cdot)$ (e.g., the rectified linear unit, ReLU) to transform $\mathbf{x}_n$. Specifically, each layer transformation can be written as

$$f_h(\mathbf{x}_n) = \varphi(W_h \mathbf{x}_n + \mathbf{b}_h) \quad (1)$$

and the utility vector $\mathbf{V}_n$ is computed in a composite form:

$$\mathbf{V}_n = (f_H \circ f_{H-1} \circ \dots \circ f_2 \circ f_1)(\mathbf{x}_n) \quad (2)$$

Finally, a softmax classification layer (i.e., the logistic function) outputs the choice probability of i as

$$P_{ni} = \frac{e^{V_{ni}}}{\sum_{j=1}^J e^{V_{nj}}} \quad (3)$$

The DNN structure generalizes the classical linear MNL model. If an NN is specified with a single output layer (i.e., without any hidden layer) and an identity activation function, the utility function in Eq. (2) would collapse to

$$\mathbf{V}_n = f(\mathbf{x}_n) = W \mathbf{x}_n + \mathbf{b} \quad (4)$$

where $W \in \mathbb{R}^{J \times D}$ can be interpreted as parameters and $\mathbf{b} \in \mathbb{R}^{J \times 1}$ as alternative-specific constants. Although closely related to the MNL model, DNNs allow for flexible model specifications through multi-layer nonlinear transformations. We illustrate in Fig. 1 a feedforward DNN structure with four hidden layers and one classification layer for a choice modeling problem with D attributes for J alternatives.

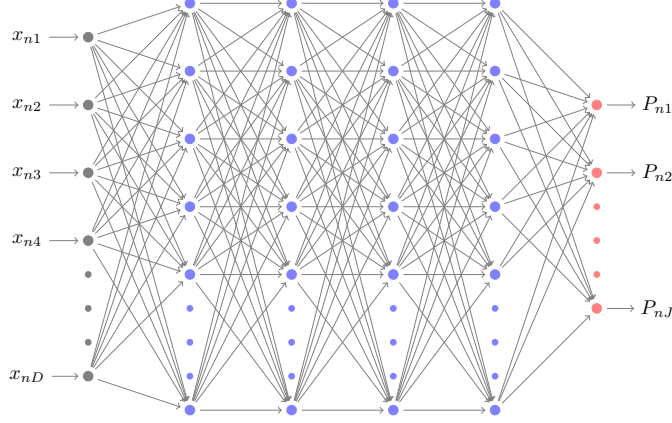


Figure 1: A feedforward DNN structure with four hidden layers and one classification layer.

3.2 Behavioral regularity metrics

In this study, we propose a novel metric to evaluate behavioral regularity, which measures the monotonicity of aggregate choice probability functions. The proposed metric essentially measures the monotonicity consistency between the model and prior knowledge on the correct signs of parameter estimates, commonly used in the subjective process of selecting plausible specifications for RUMs. The behavioral regularity metric of alternative i w.r.t. a cost variable x_d (with value x_{nd} for individual n) is defined as

$$B_{id} = \int \int \mathbb{1} \left\{ \frac{\partial P_i(\mathbf{x}_z)}{\partial x_d} < \varepsilon \right\} \rho(x_d, z) dx_d dz \quad (5)$$

where z represents the characteristic factor of a population group; $P_i(\mathbf{x}_z)$ maps the individual's attributes $\mathbf{x}_z$, including both individual-specific sociodemographics mapped from factor z , and alternative-specific cost variables, to the individual's probability of choosing i ; $\mathbb{1}\{\cdot\}$ is an indicator function that equals 1 if $\partial P_i(\mathbf{x}_z)/\partial x_d < \varepsilon$, and 0 otherwise; $\rho(x_d, z)$ denotes the joint density of attributes x_d and the population factor z ; and ε represents the modeler's prior assumptions on the monotonicity of P_i w.r.t. x_d :

- (1) $\varepsilon = 0$, termed as *strong* regularity, requires P_i to be strictly decreasing w.r.t. x_d . The formulation assumes that all individuals across population groups reduce their choice probability of alternative i with larger costs in x_d .
- (2) $\varepsilon > 0$, termed as *weak* regularity, relaxes the strict monotonicity assumption and allows P_i to be non-decreasing w.r.t. x_d . The formulation assumes that some population groups $\{z\}$ do not respond (with zero derivative) to x_d , implying that the behavioral regularity constraint becomes weaker.

As an illustration, a classical linear MNL model with a negative parameter w.r.t. x_d yields $B_{id} = 1$, which implies that all individual behaviors are consistent with the demand monotonicity assumption and well-captured by the model. We also note that Eq. (5) can be interpreted as the cumulative distribution of behavioral regularity with ε over the whole population and the domain of x_d .

The population-level behavioral regularity metric in Eq. (5) can be approximated by the mean behavioral regularity across individuals:

$$B_{id} \approx \frac{1}{N} \sum_{n=1}^N \mathbb{1} \left\{ \frac{\Delta P_{ni}}{\Delta x_{nd}} < \varepsilon \right\} \quad (6)$$

where N is the sample size, and the partial derivative is computed with finite differences. The proposed empirical regularity metric in Eq. (6) is the sample analog of the exact metric in Eq. (5). By the Glivenko–Cantelli theorem, Eq. (6) converges in probability to Eq. (5), the cumulative distribution function for the mean population behavioral regularity with ε .

Our behavioral regularity metrics can be extended to incorporate taste heterogeneity across population groups and even individuals within each group by distinguishing ε w.r.t. different groups, i.e., ε can be further specified as ε_z to reflect the group-specific thresholds. Meanwhile, our metrics only require aggregate regularity rather than individual regularity, inspired by classical economics discussions: market rationality is a fundamental law, whereas individual behaviors might present more diverse and irrational patterns (Becker, 1962).

3.3 Achieving behavioral regularity by constrained optimization

3.3.1 Unconstrained likelihood maximization

DNN-based choice models can be estimated using the likelihood maximization framework. Given a set of hyperparameters and the softmax activation function, likelihood maximization and cross-entropy minimization are mathematically equivalent, i.e., an unconstrained DNN learns parameters W through minimizing the cross-entropy L :

$$\min_W L(W) = \min_W \frac{1}{N} \sum_{n=1}^N \sum_{i=1}^J -y_{ni} \log P_i(\mathbf{x}_n; W) \quad (7)$$

The unconstrained formulation in Eq. (7) is sufficient for the estimation of conventional RUMs, since they often satisfy convexity conditions because of linear-in-parameters specifications. In a linear MNL model, for example, choice probability P_{ni} increases monotonically with utility V_{ni} according to Eq. (3). The linear specification in Eq. (4) induces monotonicity of utility w.r.t. cost variables. Therefore, if individuals indeed perceive higher utility with lower costs, the optimization would result in negative parameter estimates, and the behavioral monotonicity is clearly satisfied by evoking the chain rule. The multi-layer nonlinear transformations in DNNs allow for approximation of arbitrary functions, but these complex transformations might lead to non-monotonic choice probability functions, especially when the network is deep. In this case, unconstrained likelihood maximization can no longer guarantee a behaviorally regular model.

3.3.2 Constrained likelihood maximization

To address the irregularity issue of DNNs, we introduce a set of behavioral regularity constraints into the optimization problem, which yields

$$\min_W L(W) \quad (8)$$

$$\text{s.t. } R(\mathbf{x}_n; W) \leq 0, \quad n = 1, \dots, N \quad (9)$$

where R constrains the attributes $\mathbf{x}_n$ of an individual n , defined as $R : \mathbb{R}^{J \times D} \rightarrow \mathbb{R}$, where dimension J is the number of alternatives and D is the number of attributes. Therefore, the behavioral regularity constraints are imposed at the individual level to enhance the aggregate regularity in B_{id} . Specific behavioral regularity constraints will be designed in the next subsection.

Training DNNs with constraints is challenging. We tackle this problem by treating the *hard* constraints in Eq. (9) as *soft* constraints, motivated by the Lagrangian relaxation method. Given a hyperparameter λ , we consider the following optimization problem:

$$\min_W L(W) + \lambda \sum_{n=1}^N R(\mathbf{x}_n; W) \quad (10)$$

where λ controls the strength of the behavioral regularity constraint and can be interpreted as a Lagrangian multiplier for constrained optimization. We note that the relaxation formulation in Eq. (10) is similar to the regularization methods that are commonly applied in machine learning for model sparsity, while our motivation is to improve the behavioral regularity of the DNN choice models.

Compared with the hard constraint formulation, soft regularization can flexibly accommodate the various degrees of validity in our behavioral regularity assumptions. Similar to the motivation for the weak regularity metric, our approach allows each individual n to somewhat violate the preset constraint $R(\mathbf{x}_n)$, thus accommodating the potentially irregular behavior of certain individuals. As a result, hyperparameter λ provides insight into the consistency between behavioral regularity assumptions and the actual behavior of studied individuals. If a larger λ is required to achieve higher predictive performance, it might imply that the actual behavior is inconsistent with prior assumptions, thus providing extra insight into the validity of behavioral regularity constraints. Although the gradient regularization method does not guarantee strong behavioral regularity, it provides insights into model misspecification or the inconsistency between modeling assumptions and data.

3.4 Gradient regularization

We design gradient regularizers to improve the behavioral regularity of DNN-based choice models. Specifically, we constrain the demand feedback on generalized costs by the gradient's direction (i.e., signs of the parameter estimates) and magnitude.

For individual n , the Jacobian matrix (gradient) of demand vector $\mathbf{P} = [P_1, \dots, P_J]^\top$ w.r.t. cost variables $\{x_1, \dots, x_D\}$ can be written as

$$\nabla \mathbf{P}(\mathbf{x}_n) = \begin{bmatrix} \frac{\partial P_1}{\partial x_1}(\mathbf{x}_n) & \cdots & \frac{\partial P_1}{\partial x_D}(\mathbf{x}_n) \\ \vdots & \ddots & \vdots \\ \frac{\partial P_J}{\partial x_1}(\mathbf{x}_n) & \cdots & \frac{\partial P_J}{\partial x_D}(\mathbf{x}_n) \end{bmatrix} \quad (11)$$

which includes three types of partial derivatives:

- (1) Direct derivatives: e.g., the probability of driving w.r.t. driving cost.
- (2) Cross derivatives: e.g., the probability of driving w.r.t. train time.
- (3) Sociodemographic derivatives: e.g., the probability of driving w.r.t. the age of traveler.

The constrained likelihood maximization framework in Eq. (10) allows us to impose gradient constraints to the three types of partial derivatives in the Jacobian matrix. Since behavioral regularity is reflected by the gradient's direction, we introduce a mask matrix for individual n :

$$\Psi(\mathbf{x}_n) = \begin{bmatrix} \mathbb{1}\left\{\frac{\partial P_1}{\partial x_1}(\mathbf{x}_n) \notin \mathbb{S}_{11}\right\} & \cdots & \mathbb{1}\left\{\frac{\partial P_1}{\partial x_D}(\mathbf{x}_n) \notin \mathbb{S}_{1D}\right\} \\ \vdots & \ddots & \vdots \\ \mathbb{1}\left\{\frac{\partial P_J}{\partial x_1}(\mathbf{x}_n) \notin \mathbb{S}_{J1}\right\} & \cdots & \mathbb{1}\left\{\frac{\partial P_J}{\partial x_D}(\mathbf{x}_n) \notin \mathbb{S}_{JD}\right\} \end{bmatrix} \quad (12)$$

where $\mathbb{1}\{\cdot\}$ is an indicator function that equals 1 if $\partial P_i(\mathbf{x}_n)/\partial x_d \notin \mathbb{S}_{id}$, and 0 otherwise; and set $\mathbb{S}_{id}$ defines the expected sign of the partial derivative. Combining the mask matrix $\Psi(\mathbf{x}_n)$ and the Jacobian matrix $\nabla \mathbf{P}(\mathbf{x}_n)$, we define the *sum-based* gradient regularization using the Frobenius inner product:¹

$$R_\sigma(\mathbf{x}_n) = \langle \Psi(\mathbf{x}_n), \nabla \mathbf{P}(\mathbf{x}_n) \rangle_F \quad (13)$$

This sum-based approach flexibly accommodates different prior assumptions on the signs of the derivatives. Set $\mathbb{S}_{id}$ can take negative values ($\mathbb{S}_{id} = \mathbb{R}^-$), positive values ($\mathbb{S}_{id} = \mathbb{R}^+$), or any real values ($\mathbb{S}_{id} = \mathbb{R}$), depending on the prior assumption on attribute x_d 's effect on demand $P_i(\mathbf{x}_n)$. For example, by imposing $\mathbb{S} = \mathbb{R}^-$ on the direct derivatives, they are expected to be negative and penalized if non-negative. On the

¹For real matrices, we have $\langle A, B \rangle_F = \sum_{i,j} A_{ij}B_{ij}$.

other hand, when there is no prior assumption regarding a derivative, we allow all possible signs by taking $\mathbb{S} = \mathbb{R}$. Despite such flexibility, we only impose negative constraints on the direct derivatives throughout our empirical experiments, which is the least controversial among all possibilities.

As an alternative to the sum-based approach, we could also regularize the gradient’s magnitude, implying that demands are not expected to change drastically with small cost perturbations. Using the same notations, we define the *norm-based* regularization as

$$R_\nu(\mathbf{x}_n) = \|\nabla \mathbf{P}(\mathbf{x}_n)\|_F^2 = \langle \nabla \mathbf{P}(\mathbf{x}_n), \nabla \mathbf{P}(\mathbf{x}_n) \rangle_F \quad (14)$$

This norm-based approach is relatively common in the computer science literature (Drucker and Le Cun, 1991; Jakubovitz and Giry, 2018), thus serving as a benchmark regularization method for our empirical experiments. Although smoothness is a relatively common assumption from a pure mathematical perspective, it is not founded on strong behavioral regularity beliefs due to possible threshold effects of pricing.

The regularization terms proposed above are termed as probability gradient regularizers (PGRs) because they exploit the analytical relationship between demand monotonicity and probability gradients $\nabla \mathbf{P}(\mathbf{x}_n)$. Due to the computational chain among utilities, choice probabilities, and log-likelihoods, it is also possible to replace the probability gradients by utility and log-likelihood gradients. The two alternative regularizers are defined as:

- (1) Utility gradient regularizers (UGRs): According to the softmax function in Eq. (3), choice probability $P_i(\mathbf{x}_n)$ increases monotonically with utility $V_i(\mathbf{x}_n)$. Consequently, demand monotonicity can be retained by regularizing the utility monotonicity w.r.t. generalized costs and evoking the chain rule. Therefore, we construct UGRs by replacing $\mathbf{P}(\mathbf{x}_n)$ with $\mathbf{V}(\mathbf{x}_n)$ in derivation.
- (2) Log-likelihood gradient regularizers (LGRs): We define the individual- and alternative-specific log-likelihood as $l_i(\mathbf{x}_n) = -y_{ni} \log P_i(\mathbf{x}_n)$. Since logarithmic transformation is monotonic, demand monotonicity can be retained by regularizing the log-likelihood monotonicity w.r.t. generalized costs. Thus we construct LGRs by replacing $\mathbf{P}(\mathbf{x}_n)$ with $\mathbf{l}(\mathbf{x}_n)$ in derivation.

In brief, by combining sum- and norm-based regularization with probability, utility, and log-likelihood gradients, we have designed six gradient regularizers. They are hereafter referred to as the sum-PGR, sum-UGR, sum-LGR, norm-PGR, norm-UGR, and norm-LGR, all of which will be tested thoroughly in our empirical experiments.

When hidden layers are not present, the norm-UGR reduces to L_2 regularization because the gradient of the utility is simply the DNN parameters W . On the other hand, the proposed gradient regularizers differ from commonly used sparsity regularizers (e.g., L_1 and L_2 norms) if any hidden layer is present. This is because the derivatives of our gradient regularizers become nonseparable and nonlinear in DNN parameters W , as opposed to L_2 norms whose derivatives would be separable and linear in W . This difference also implies that our gradient regularizers would still be effective under various averaging scheme employed in algorithms like Adam, making the proposed approaches more robust to the choice of algorithms.

4 Setup of experiments

4.1 Datasets

Our experiments use two datasets from Chicago and London, with distinguished car- and transit-dependent travel patterns, to examine the gradient regularizers and the behavioral regularity metrics. The first dataset was collected by the Chicago Metropolitan Agency for Planning (CMAP) in the My Daily Travel Survey in 2018–2019.² After preliminary cleaning, the full dataset retains 26,099 trips made by four travel modes: driving, walking, train, and cycling. Driving accounts for 70% of these trips, whereas the proportion of cycling trips is negligible. Hence the walking and cycling modes were merged into a single active mode to create a more balanced dataset. Based on the spatial information of each trip in terms of origin and destination, we

²See <https://www.cmap.illinois.gov/data/transportation/travel-survey>.

compiled level of service data by utilizing Google Directions API to collect the travel time of each mode, where active times were calculated by averaging walking and cycling times. Train costs were provided by the dataset, while driving costs were computed by summing the money paid to toll plazas en route and parking lots. The K -nearest neighbors algorithm was applied to impute the missing data, especially for driving and train costs. This study uses in total ten attributes: two continuous alternative attributes (travel time of each mode, and travel costs of driving and train), three discrete sociodemographics (age, household size, and number of cars in the household), and five sociodemographic indicators (for higher education, males, one-person households, one-car households, and high-income households, respectively). The basic statistics of the full CMAP dataset are summarized in [Table A.1](#).

The second dataset is the London Travel Demand Survey (LTDS; [Hillel et al., 2018](#); [Wang et al., 2020b](#)), which has 81,086 trips made by four travel modes: walking, cycling, public transit, and driving. The proportion of cycling trips is again negligible, hence an active mode is created with travel time defined as the average between walking and cycling times. Meanwhile, the transit time is defined as the sum of access time, in-vehicle time, and transfer time. We use in total nine attributes: two continuous alternative attributes (travel time of each mode, and travel costs of driving and transit), two discrete individual-specific variables (number of cars in the household, and number of transfers in transit), and five sociodemographic indicators (for the youth, the elderly, males, driving license, and one-car households, respectively). The basic statistics of the full LTDS dataset are summarized in [Table A.2](#).

The two datasets were further reprocessed to create three smaller datasets for each to examine the effects of large versus small sample sizes, and in-domain versus out-of-domain generalizations. The first dataset, named as *10K-Random*, incorporates 10,000 trips with 70% randomly sampled for training, 10% for validation, and 20% for testing. This dataset is considered as the benchmark for the ideal modeling scenario with a sufficiently large sample size. The second dataset, named as *1K-Random*, contains 1,000 trips with 80% randomly sampled for training and 20% for validation. To avoid random variation due to small sample sizes, another 500 trips were randomly sampled for testing. The 1K-Random dataset aims to simulate a classical choice modeling scenario where limited samples are available. By comparing the results between these two datasets, we could evaluate how predictive performance and behavioral regularity vary with sample sizes. The third dataset, named as *10K-Sorted*, employs a different strategy for data splitting, where the 10,000 trips are sorted by driving cost, while the upper 20% were used for testing, and the lower 80% were further randomly sampled for training (70%) and validation (10%). As shown in [Appendix A.2](#), the distributions of variables are quite different between the training and test sets, with significantly higher mean and standard deviations in the test set. This training-testing split scheme simulates the testing carried out on more expensive trips. Such out-of-domain generalizability is not only of theoretical interests, but also highly relevant in practice because it investigates model transferability, i.e., how the models perform in a target context distinct from their source context. One rationale of the sampling scheme for 10K-Sorted datasets is that it resembles cross-city policy learning. Local governments regularly seek to implement transportation policies (e.g., congestion charging) that originate from other cities. The out-of-domain generalization can simulate the data and modeling challenges in such cross-city policy learning.

4.2 Experimental design

Our experiments use the training set for model training, the validation set for hyperparameter searching, and the test set for model evaluation and comparison. Using the training set of 10K-Random (CMAP), we show in [Appendix B](#) that both Adam ([Kingma and Ba, 2014](#)) and AdamW ([Loshchilov and Hutter, 2017](#)) are empirically suitable for our experiments, while standard stochastic gradient descent (SGD) converges much slower and results in unreasonable individual demand functions. The difference between Adam and AdamW lies in the implementation of weight decay as another type of regularization, which is not included in the experiments, thus teasing out the effects of gradient regularization from other factors. The training set was divided into 10 batches for model training, with a learning rate of 10^{-3} . To ensure convergence, we train each model until the validation loss in consecutive iterations reaches an optimum.

Using the validation sets, we selected the optimal regularization strength λ for each of the gradient regularizers by overall model performance, thus balancing predictive power and behavioral regularity. The DNN architecture was chosen with four hidden layers and 100 neurons per layer after random search. As the range of λ depends on the dataset, model class, and gradient regularizer, we only show an example of the hyperparameter space in [Table 1](#), where we took λ values from 10^{-4} to 100 in a logarithmic scale to fully demonstrate the effects of λ and select the optimum. It is expected that the DNNs with extremely small λ 's approximate the benchmark DNN, while those with large λ 's sacrifice predictive power for behavioral regularity.

Table 1: Hyperparameter space for DNNs with sum-XGR (10K-Random, CMAP).

Hyperparameter	Values
Depth (number of hidden layers)	3, 4 , 5, 6
Width (number of neurons per layer)	50, 100 , 150
Regularization strength	10^{-4} , 10^{-3} , 0.01 , 0.1, 1, 10, 100

Lastly, all models are evaluated by their performance in the test sets. Particularly, to mitigate model randomness, we analyze the ensemble performance by averaging the results of 10 model replications. [Section 5](#) will focus on comparing model performance using the test sets, while the training and validation performance are reported in [Appendix C](#). The models are evaluated by five metrics: log-likelihood, prediction accuracy, F_1 score, strong behavioral regularity, and weak behavioral regularity. The first three metrics focus on predictive power, measuring how well a model fits the observed outputs. Among the three metrics, log-likelihood is the most important one because of its probabilistic nature, its wide adoption in the field of discrete choice analysis, and its solid theoretical foundation for model convergence. Prediction accuracy and F_1 score are also adopted because the former is the most common metric in machine learning and the latter tackles the potential evaluation problem in imbalanced datasets. In addition to these predictive metrics, we also evaluate the models using strong and weak behavioral regularities based on [Eq. \(6\)](#). To empirically compute the two behavioral metrics, we set parameter ε to a small negative number for strong regularity and a small positive number for weak regularity. Despite the theoretical threshold $\varepsilon = 0$ for strong regularity, we set the value slightly lower than zero to enhance numerical stability and distinguish between the two metrics.³ We conducted a sensitivity test in [Appendix E](#) to discuss the impact of ε on the metric values. [Section 5](#) will fully demonstrate the trade-off between predictive and behavioral metrics by adjusting the strength of regularization.

4.3 Models

Three types of models are compared, including MNL models from the DCM family, standard DNNs, and TasteNets, a hybrid model combining features of DCMs and NNs ([Han et al., 2022](#)). The proposed gradient regularizers are implemented on both DNNs and TasteNets. The MNL models are estimated with `PyLogit`, and DNNs and TasteNets are implemented with `PyTorch`. The following linear-in-parameters utility function is specified for the MNL models:

$$\text{Driving: } V_{n1} = \beta_{t1}t_{n1} + \beta_{c1}c_{n1} \quad (15)$$

$$\text{Train/Transit: } V_{n2} = \alpha_2 + \gamma_2\mathbf{z}_n + \beta_{t2}t_{n2} + \beta_{c2}c_{n2} \quad (16)$$

$$\text{Active mode: } V_{n3} = \alpha_3 + \gamma_3\mathbf{z}_n + \beta_{t3}t_{n3} \quad (17)$$

where t_{ni} is the travel time of individual n by alternative $i = \{1, 2, 3\}$, c_{ni} is the travel cost of n by $i = \{1, 2\}$, $\mathbf{z}_n$ is a set of variables specific to n , and $\mathbf{w} = \{\alpha, \beta, \gamma\}$ is the set of parameters to be estimated. Our experiments focus on evaluating the effectiveness of the proposed gradient regularization in improving DNNs'

³As a result, the empirical strong regularity of MNL presented in [Section 5](#) could be slightly lower than 100%.

behavioral regularity and prediction power, while the MNL models are only benchmarks to demonstrate their inherent behavioral regularity.

For TasteNets, we follow the model specification as in Han et al. (2022). Specifically, this implementation maps individual characteristics into individual-specific taste parameters for alternative attributes via a feedforward NN with one hidden layer:

$$V_{ni} = T(\mathbf{z}_n; W)^\top \mathbf{x}_{ni} = \boldsymbol{\beta}_{ni}^\top \mathbf{x}_{ni} \quad (18)$$

where T represents the NN architecture, W is a set of weights, and $\boldsymbol{\beta}_{ni}$ is a vector of taste parameters specific to individual n and alternative i . Han et al. (2022) suggested imposing hard constraints on taste parameters to enforce behavioral regularity, given that Eq. (18) is linear in taste parameters. For example, let b_{ni} denote the NN-embedded taste parameter for travel cost, we can enforce sign constraint by setting $-\text{ReLU}(-b_{ni})$ or $-\exp(-b_{ni})$ to ensure behaviorally regular TasteNets. We also compare the performance of soft and hard constraints on TasteNets in Section 5.

5 Results

In this section, we present the results of our empirical work in three stages. Section 5.1 compares the behaviorally regularized DNNs and TasteNets with benchmark models, including their counterparts without regularization and MNL models, regarding predictive power and behavioral regularity metrics. Specifically, we design the large sample (10K-Random), small sample (1K-Random), and out-of-domain generalization (10K-Sorted) scenarios. We note that although sharing certain similarity, out-of-sample generalization (i.e., testing on new samples, like the 10K- and 1K-Random datasets) and out-of-domain generalization (i.e., testing on new distributions, like the 10K-Sorted datasets) are two different concepts. In particular, the former assumes that the training and test sets follow the same statistical pattern, whereas the latter refers to unforeseen distribution shift such as cross-city policy transfer (Liu et al., 2021). Section 5.2 further investigates how the strength of regularization influences the trade-off between predictive power and behavioral regularity in each scenario. Finally, our empirical findings are summarized in Section 5.3.

5.1 Enhancing model performance with gradient regularization

5.1.1 Large sample scenario

Using two 10K-Random datasets from Chicago and London, we evaluate the regularized DNNs by five metrics in the test sets: log-likelihood, accuracy, and F_1 score that capture the models’ predictive power, as well as strong and weak regularities that describe their behavioral regularity. Table 2 summarizes the performance of DNNs and TasteNets with optimal regularization strengths, TasteNets with hard constraints, alongside the DNN, TasteNet, and MNL benchmarks. The training and validation performance of the models are summarized in Appendix C.1. In each row, the best and second best metrics are marked in bold and underlined, respectively. For DNNs and TasteNets, each metric is averaged across ten trained model replications, with standard deviations shown in parentheses. To illustrate demand monotonicity, we plot the individual demand functions of the three alternatives for selected models in Fig. 2, where light and dark curves represent the results of training replications and ensembles, respectively. Fig. 2 uses an “average individual” as the market representative, and varies the driving cost while keeping all other variables constant. Furthermore, Appendix D illustrates the impact of regularization strength on individual demands. By examining the large sample and in-domain scenarios (10K-Random), we have three major empirical findings.

First, the benchmark DNNs without gradient regularization outperform the MNL models in predictive power but underperform them in behavioral regularity, especially for the CMAP data. The benchmark DNN improves the MNL’s log-likelihood by 5.2% of its absolute value using the CMAP data and 5.5% using the LTDS data. The empirical results show that log-likelihood is more sensitive to predictive performance

Table 2: Model performance in the test sets of 10K-Random.

Metric: mean (SD)	DNN				TasteNet					RUM	
	No GR	PGR	UGR	LGR	No GR	PGR	UGR	LGR	ReLU	Exp	MNL
Panel 1: CMAP data, sum-XGR											
Log-likelihood	−1351.9 (4.697)	−1344.3 (5.521)	−1350.2 (5.803)	<u>−1347.4</u> (5.110)	−1438.3 (5.834)	−1438.4 (6.027)	−1439.0 (5.992)	−1438.4 (6.099)	−1463.0 (20.05)	−1439.7 (6.073)	−1426.3 (0)
Accuracy	<u>0.729</u> (0.003)	0.730 (0.002)	<u>0.729</u> (0.002)	<u>0.729</u> (0.002)	0.713 (0.003)	0.713 (0.002)	0.713 (0.003)	0.713 (0.002)	0.712 (0.002)	0.717 (0.003)	0.718 (0)
F_1 score	0.691 (0.005)	0.698 (0.004)	0.694 (0.004)	<u>0.696</u> (0.004)	0.654 (0.006)	0.654 (0.005)	0.654 (0.005)	0.654 (0.005)	0.650 (0.002)	0.664 (0.004)	0.669 (0)
Strong regularity	0.888 (0.066)	0.990 (0.003)	0.982 (0.012)	0.991 (0.003)	<u>0.998</u> (0.002)	0.999 (0.001)	0.999 (0.001)	0.999 (0.001)	0.426 (0.320)	0.999 (0.001)	<u>0.998</u> (0)
Weak regularity	0.922 (0.061)	<u>0.999</u> (0.001)	0.996 (0.006)	<u>0.999</u> (0.001)	<u>0.999</u> (0.002)	1.000 (0.001)	1.000 (0.001)	1.000 (0.001)	1.000 (0.000)	1.000 (0.000)	1.000 (0)
Panel 2: CMAP data, norm-XGR											
Log-likelihood	−1351.9 (4.697)	−1353.9 (5.018)	−1362.0 (3.110)	−1354.0 (4.451)	−1438.3 (5.834)	−1439.1 (5.836)	−1469.5 (6.287)	−1440.8 (5.804)	−1463.0 (20.05)	−1439.7 (6.073)	−1426.3 (0)
Accuracy	0.729 (0.003)	0.729 (0.004)	0.725 (0.004)	<u>0.727</u> (0.003)	0.713 (0.003)	0.713 (0.003)	0.710 (0.003)	0.712 (0.003)	0.712 (0.002)	0.717 (0.003)	0.718 (0)
F_1 score	0.691 (0.005)	<u>0.688</u> (0.007)	0.677 (0.009)	0.683 (0.007)	0.654 (0.006)	0.653 (0.005)	0.643 (0.005)	0.652 (0.005)	0.650 (0.002)	0.664 (0.004)	0.669 (0)
Strong regularity	0.888 (0.066)	0.857 (0.069)	0.706 (0.051)	0.815 (0.068)	<u>0.998</u> (0.002)	<u>0.998</u> (0.002)	0.929 (0.030)	0.997 (0.004)	0.426 (0.320)	0.999 (0.001)	<u>0.998</u> (0)
Weak regularity	0.922 (0.061)	0.893 (0.067)	0.756 (0.051)	0.851 (0.069)	<u>0.999</u> (0.002)	<u>0.999</u> (0.002)	0.941 (0.026)	0.998 (0.003)	1.000 (0.000)	1.000 (0.000)	1.000 (0)
Panel 3: LTDS data, sum-XGR											
Log-likelihood	<u>−1292.0</u> (7.894)	−1288.6 (9.668)	−1288.6 (3.765)	−1305.0 (7.316)	−1305.8 (2.226)	−1308.6 (2.795)	−1317.6 (4.280)	−1308.3 (2.670)	−1335.5 (20.45)	−1312.9 (2.706)	−1366.9 (0)
Accuracy	0.729 (0.005)	0.729 (0.004)	0.727 (0.002)	0.724 (0.004)	<u>0.732</u> (0.003)	0.730 (0.002)	0.728 (0.004)	0.730 (0.002)	0.725 (0.006)	0.736 (0.002)	0.730 (0)
F_1 score	0.727 (0.004)	<u>0.728</u> (0.004)	0.725 (0.002)	0.721 (0.006)	<u>0.728</u> (0.003)	0.726 (0.002)	0.723 (0.004)	0.726 (0.002)	0.721 (0.006)	0.733 (0.002)	0.726 (0)
Strong regularity	0.950 (0.034)	<u>0.994</u> (0.004)	<u>0.994</u> (0.009)	0.997 (0.003)	0.942 (0.021)	0.964 (0.013)	0.987 (0.008)	0.972 (0.012)	0.933 (0.043)	0.990 (0.002)	0.993 (0)
Weak regularity	0.969 (0.027)	<u>0.999</u> (0.001)	0.998 (0.004)	1.000 (0.000)	0.966 (0.019)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0)

than accuracy and F_1 score due to its probabilistic nature. Meanwhile, the difference between accuracy and F_1 score reflects whether the dataset is balanced: the two metrics are similar for the LTDS data but remain a gap for the car-dominated CMAP data, indicating that the former is more balanced in travel modes. Moreover, the benchmark DNNs are behaviorally irregular, as suggested by the relatively lower strong regularity (88.8%) and weak regularity (92.2%) using the CMAP data. This is consistent with finding illustrated in Fig. 2a: the average market share of driving is non-monotonic w.r.t. driving cost, which is consistently presented in all local DNN models. This suggests that DNNs typically have high predictive performance but low behavioral regularity, aligning with the findings from many previous studies (Wang et al., 2020a,b; Wong and Farooq, 2021; Xia et al., 2023). By contrast, the benchmark TasteNets have high behavioral regularity but relatively low predictive power, possibly due to their simpler NN architecture and linear-in-parameters specifications.

Second, the DNNs with soft constraints outperform the TasteNets with hard constraints in terms of log-likelihood. By design, both ReLU and exponential constraints guarantee 100% weak regularity, but only the latter can simultaneously guarantee high strong regularity. Despite being behaviorally regular, the TasteNets with either soft or hard constraints have lower log-likelihood than the DNNs with soft constraints. This example demonstrates the flexibility and approximation power of nonlinear DNN specifications compared with the linear-in-parameters TasteNet specifications. It might also suggest the existence of irregular behaviors for certain individuals or price ranges, highlighting the flexibility of soft constraints in capturing

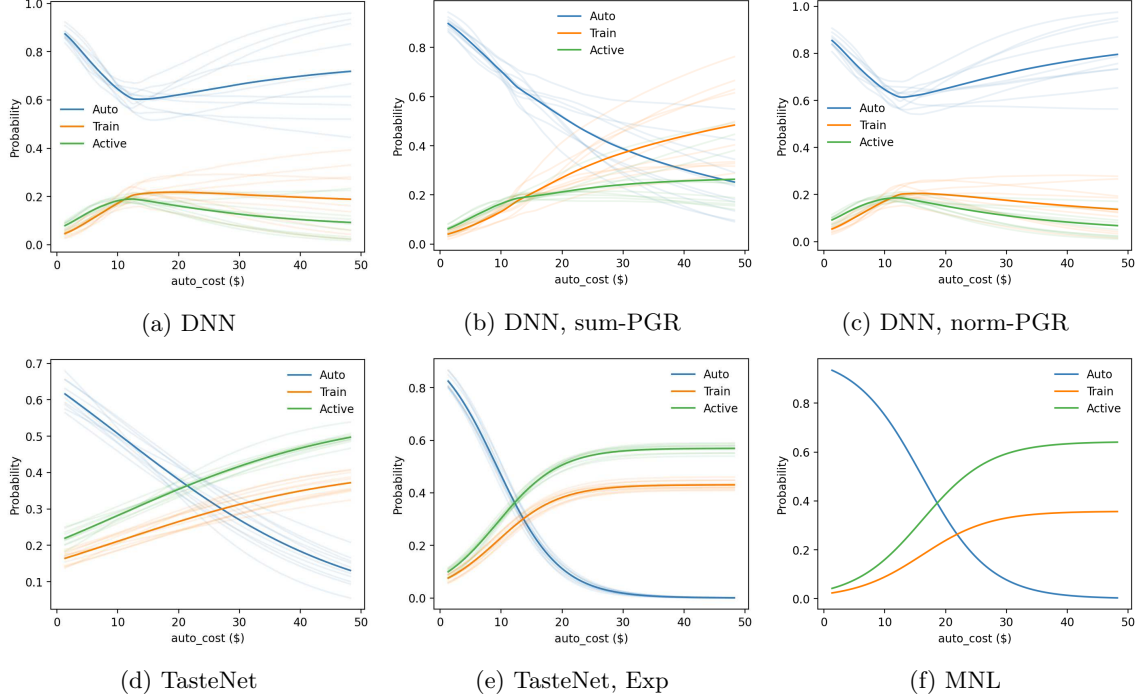


Figure 2: Individual demands as functions of driving costs (10K-Random, CMAP).

various behavioral mechanisms (Bagwell and Riordan, 1991; Stiving, 2000). As shown in Table 2 and Fig. 2, the exponentially constrained TasteNets resemble the MNL models in performance metrics and individual demands, reflecting the intrinsic connection between TasteNet and MNL models.

Third, sum-based gradient regularization can improve the behavioral regularity of DNNs and TasteNets without sacrificing their predictive power. The demand functions of the regularized DNNs become highly monotonic, as indicated by strong and weak regularities both approaching 100%. Meanwhile, sum-based gradient regularization is able to enhance DNNs’ predictive power in both datasets. Table 2 demonstrates that sum-PGR, which directly regularizes the demand functions, is slightly more effective than sum-UGR and sum-LGR, which exploit demand monotonicity through the chain rule. This finding is further elaborated by Fig. 2b, where the regularized DNN has individual demand functions more consistent with the MNL: driving is less favored due to increased costs, while train and active mobility see higher demand. The regularized demand functions are more monotonic not only in the ensemble model, but also in training replications. We note that similar behavioral regularization effects of sum-based gradient regularizers are also observed for TasteNets on the LTDS data.

Last but not least, the norm-based gradient regularization fails to enhance behavioral regularity or predictive power. For example, norm-UGR with a small λ can preserve accuracy and behavioral regularity for DNNs and TasteNets, but it would eventually lead to low predictive power and strong regularity as we increase λ . These results hold for all three norm-based approaches, potentially because they tend to flatten and smooth the demand curves. Therefore, the optimal DNNs and TasteNets in Table 2 have small λ ’s and look similar to the corresponding benchmarks (see Fig. 2c for example). With strong norm-based gradient regularization, as shown in Fig. D.1e, individual demand curves become almost flat and could not reflect the decision mechanism: travelers might not respond to cost changes at certain points, but are highly unlikely to be insensitive to all cost changes. Since smaller λ ’s lead to better performance, we might conclude that the datasets or models do not change abruptly due to cost perturbations. In brief, although regularizing the gradient norm is a common practice in computer science (Drucker and Le Cun, 1991; Jakubovitz and Giryes, 2018; Sokolić et al., 2017), it is not founded on prior beliefs in behavioral regularity and thus not effective

for our purposes. For conciseness, we will not analyze norm-based gradient regularization in the next two scenarios.

5.1.2 Small sample scenario

In this subsection, we focus on exemplifying the effectiveness of the proposed gradient regularizers in a typical choice modeling scenario where the number of available samples is limited due to resource restrictions or privacy concerns. The same analysis is applied to the 1K-Random CMAP and LTDS datasets. Table 3 illustrates the performance of DNNs, TasteNets, and benchmark models, while Fig. 3 elaborates on the individual demand functions. Furthermore, Appendix C.2 summarizes the training and validation performance of the models.

Table 3: Model performance in the test sets of 1K-Random.

Metric: mean (SD)	DNN				TasteNet						RUM
	No GR	PGR	UGR	LGR	No GR	PGR	UGR	LGR	ReLU	Exp	MNL
Panel 1: CMAP data, sum-XGR											
Log-likelihood	−375.1 (3.096)	− 367.9 (4.181)	−374.2 (3.298)	− <u>369.7</u> (3.304)	−389.3 (2.838)	−385.2 (3.351)	−383.1 (3.139)	−386.6 (3.447)	−386.3 (2.506)	−378.0 (4.123)	−380.2 (0)
Accuracy	0.705 (0.002)	0.703 (0.007)	0.676 (0.010)	0.696 (0.011)	0.700 (0.005)	0.679 (0.012)	0.686 (0.010)	0.677 (0.012)	0.696 (0.005)	<u>0.707</u> (0.004)	0.718 (0)
F_1 score	<u>0.648</u> (0.004)	0.645 (0.018)	0.559 (0.027)	0.619 (0.030)	0.619 (0.012)	0.563 (0.028)	0.580 (0.023)	0.559 (0.028)	0.610 (0.011)	0.637 (0.010)	0.665 (0)
Strong regularity	0.664 (0.173)	0.985 (0.009)	0.985 (0.011)	0.985 (0.011)	0.659 (0.129)	0.979 (0.015)	0.983 (0.013)	0.979 (0.013)	0.461 (0.195)	0.999 (0.001)	<u>0.996</u> (0)
Weak regularity	0.728 (0.162)	<u>0.999</u> (0.001)	0.997 (0.005)	<u>0.999</u> (0.002)	0.685 (0.128)	0.997 (0.004)	0.995 (0.006)	0.992 (0.009)	1.000 (0.000)	1.000 (0.000)	1.000 (0)
Panel 2: LTDS data, sum-XGR											
Log-likelihood	− <u>322.6</u> (3.455)	−328.8 (8.146)	− 317.8 (3.440)	−335.2 (11.30)	−345.1 (2.337)	−339.7 (2.023)	−343.5 (3.035)	−340.1 (1.974)	−341.0 (2.572)	−335.5 (2.951)	−331.2 (0)
Accuracy	0.737 (0.007)	0.720 (0.013)	<u>0.740</u> (0.007)	0.717 (0.021)	0.725 (0.004)	0.734 (0.006)	0.724 (0.009)	0.733 (0.006)	0.729 (0.006)	0.733 (0.008)	0.746 (0)
F_1 score	0.732 (0.007)	0.703 (0.020)	<u>0.734</u> (0.008)	0.694 (0.038)	0.711 (0.005)	0.720 (0.006)	0.705 (0.013)	0.718 (0.005)	0.715 (0.007)	0.725 (0.009)	0.740 (0)
Strong regularity	0.904 (0.069)	0.999 (0.002)	0.992 (0.014)	0.997 (0.008)	0.939 (0.023)	0.964 (0.013)	0.999 (0.002)	0.965 (0.01)	0.924 (0.046)	0.999 (0.002)	<u>0.998</u> (0)
Weak regularity	0.913 (0.067)	1.000 (0.001)	0.993 (0.014)	0.998 (0.006)	0.943 (0.024)	0.984 (0.012)	<u>0.999</u> (0.002)	0.981 (0.013)	1.000 (0.000)	1.000 (0.000)	1.000 (0)

We find that our gradient regularizers are even more effective than in the large sample scenario. First, without gradient regularization, the benchmark DNNs’ behavioral regularity becomes much worse with a small sample, e.g., the strong regularity drops by 22.4 percentage points and the weak regularity drops by 19.4 percentage points using the CMAP data. Similarly, shallower NNs like the benchmark TasteNets also have worse behavioral regularity in the small sample scenario. Meanwhile, the benchmark models fail to outperform MNL in predictive power, especially for the CMAP data. As shown in Fig. 3a, the benchmark DNN’s individual demand curves are non-monotonic and contradictory to the law of demand. Second, sum-based gradient regularization succeeds in enhancing all metrics of DNNs and most metrics of TasteNets, as shown in Table 3 and Fig. 3. In other words, the regularized DNNs and TasteNets outperform their corresponding benchmark models in both predictive power and behavioral regularity. In the CMAP case, sum-PGR improves the benchmark DNN’s log-likelihood by 1.9% of its absolute value and its strong regularity by 32.1 percentage points. Moreover, from Table 3 and Fig. 3 we can draw similar findings for the hard constraints, e.g., the exponentially constrained TasteNets resemble the MNL models in performance metrics and individual demands. In addition, ensembles of regularized models have comparable overall

performance to the MNL when using such a small sample size.

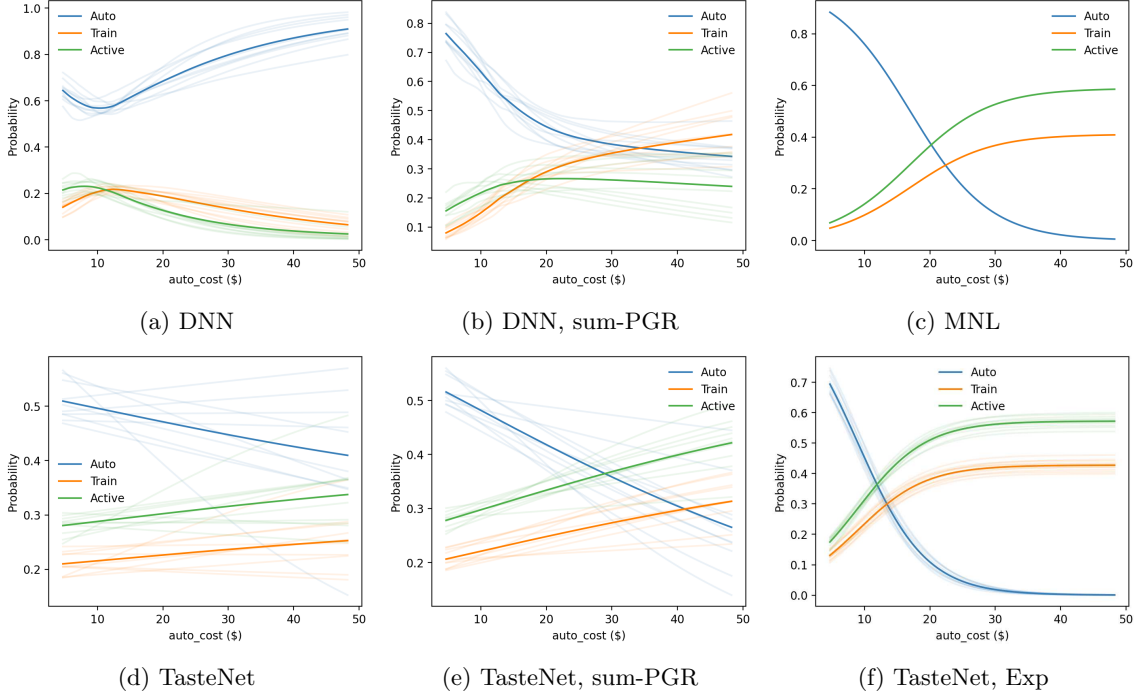


Figure 3: Individual demands as functions of driving costs (1K-Random, CMAP).

5.1.3 Out-of-domain generalization

The large and small sample scenarios above assist in examining only in-domain generalization for the random split of training and test data. Although random split is the most common practice, we are also interested in the out-of-domain generalizability of DNNs. Out-of-domain generalization is highly relevant to transportation engineering, system design, and urban planning, such as the cross-city policy transfer: when assessing whether a city should build a subway, transportation planners sometimes cite the ridership of subway systems in other cities. Our data split scheme in the 10K-Sorted datasets can be interpreted as using the patterns of individual choice behavior in one city (as in the training set) to extrapolate those in another city (as in the test set). Here we emulate such policy setting by testing the DNNs’ out-of-domain generalizability using predictive power and behavioral regularity metrics.

The results suggest that gradient regularization could drastically improve the out-of-domain generalizability of DNNs, even more effectively than improving their in-domain generalizability. Table 4 summarizes the performance of DNNs, TasteNets, and benchmark models in the test sets, while their training and validation performance are summarized in Appendix C.3. Under this setting, DNNs and TasteNets exhibit great flexibility with higher log-likelihood than the MNL model. However, behavioral regularity could be relatively low for these NN models, as reflected in their regularity metrics. Fig. 4 visualizes individual demands as functions of transit costs using the LTDS data, in which the benchmark DNN performs unreasonably in the test set, as shown to the right of the data split threshold (dashed gray line). Second, sum-based gradient regularization has the potential to simultaneously improve the predictive power and behavioral regularity of benchmark DNNs and TasteNets. For an extreme case, sum-UGR dramatically raises the strong regularity of the benchmark DNN from 0.185 to 0.979 using the LTDS data. Table 4 also reveals the case-dependent effects of hard constraints: they can significantly improve both behavioral regularity and predictive power of a poorly behaved benchmark TasteNet (the LTDS case), but are ineffective for a benchmark that is already highly regular (the CMAP case).

Table 4: Model performance in the test sets of 10K-Sorted.

Metric: mean (SD)	DNN				TasteNet						RUM
	No GR	PGR	UGR	LGR	No GR	PGR	UGR	LGR	ReLU	Exp	MNL
Panel 1: CMAP data, sum-XGR											
Log-likelihood	-1356.1 (134.1)	-1243.5 (55.15)	-1167.9 (27.64)	<u>-1232.4</u> (54.87)	-1485.9 (47.92)	-1873.7 (45.52)	-1901.4 (36.62)	-2003.9 (50.23)	-2315.4 (395.4)	-1995.0 (51.83)	-2025.7 (0)
Accuracy	0.783 (0.011)	<u>0.788</u> (0.002)	0.789 (0.003)	<u>0.788</u> (0.002)	0.750 (0.014)	0.726 (0.008)	0.725 (0.006)	0.705 (0.008)	0.578 (0.134)	0.723 (0.007)	0.722 (0)
F ₁ score	0.722 (0.010)	<u>0.726</u> (0.006)	0.727 (0.003)	0.724 (0.005)	0.707 (0.006)	0.721 (0.005)	0.720 (0.004)	0.708 (0.006)	0.603 (0.112)	0.719 (0.005)	0.721 (0)
Strong regularity	0.317 (0.240)	0.857 (0.099)	0.923 (0.087)	0.865 (0.071)	1.000 (0.000)	0.980 (0.004)	0.978 (0.004)	0.981 (0.004)	0.933 (0.169)	0.969 (0.005)	<u>0.984</u> (0)
Weak regularity	0.487 (0.230)	0.974 (0.037)	<u>0.983</u> (0.040)	0.977 (0.023)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.001)	1.000 (0.000)	1.000 (0.000)	1.000 (0)
Panel 2: LTDS data, sum-XGR (transit cost)											
Log-likelihood	-1137.9 (34.33)	<u>-1110.4</u> (23.86)	-1108.4 (21.15)	-1112.1 (26.67)	-1221.1 (33.10)	-1170.2 (27.01)	-1171.1 (28.54)	-1175.0 (30.25)	-1182.3 (54.11)	-1150.1 (29.62)	-1301.6 (0)
Accuracy	0.777 (0.010)	0.784 (0.007)	0.794 (0.007)	0.782 (0.006)	0.776 (0.009)	<u>0.786</u> (0.008)	0.785 (0.008)	0.783 (0.007)	0.782 (0.017)	0.781 (0.008)	0.780 (0)
F ₁ score	0.768 (0.011)	<u>0.776</u> (0.006)	0.778 (0.008)	<u>0.776</u> (0.006)	0.765 (0.008)	0.772 (0.008)	0.772 (0.008)	0.772 (0.007)	0.769 (0.018)	0.768 (0.008)	0.770 (0)
Strong regularity	0.185 (0.075)	0.968 (0.032)	0.979 (0.029)	0.907 (0.062)	0.313 (0.007)	0.878 (0.095)	0.872 (0.095)	0.720 (0.083)	0.248 (0.084)	0.982 (0.005)	<u>0.980</u> (0)
Weak regularity	0.207 (0.080)	0.977 (0.026)	0.984 (0.025)	0.925 (0.055)	0.343 (0.008)	<u>0.993</u> (0.010)	0.988 (0.012)	0.900 (0.045)	1.000 (0.000)	1.000 (0.000)	1.000 (0)

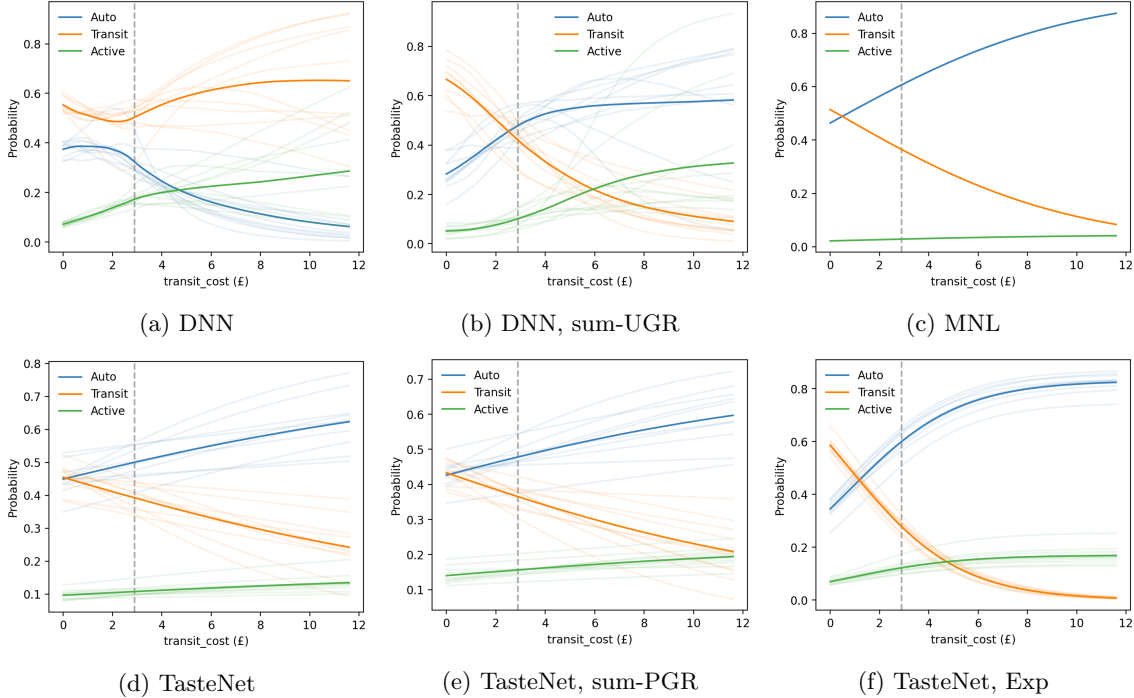


Figure 4: Individual demands as functions of transit costs (10K-Sorted, LTDS).

5.2 Trade-off between predictive power and behavioral regularity

Section 5.1 demonstrates the potentials of sum-based gradient regularization in enhancing the behavioral regularity and predictive power of benchmark DNNs and TasteNets under three settings. This subsection

will further investigate the trade-off between predictive power and behavioral regularity in these scenarios. Although the optimization objective in Eq. (10) demonstrates a clear substitution effect between predictive power and behavioral regularity in the training set, it remains an open question whether this effect persists in the test set. As detailed below, we find the same substitution effect in the test sets of 10K-Random, indicating that predictive power decreases and behavioral regularity increases with stronger sum-based gradient regularization. On the other hand, interestingly, complementary effects between predictive power and behavioral regularity are observed for in-domain generalization with 1K-Random and out-of-domain generalization with 10K-Sorted. This suggests that adequate sum-based gradient regularization can enhance both predictive power and behavioral regularity, especially in the typical choice modeling setting with limited sample sizes, and for the purpose of policy evaluations.

5.2.1 Large sample scenario: substitution effects

In a large sample scenario, higher behavioral regularity often has a trade-off with lower predictive power. The effects of regularization strength λ on the five metrics are illustrated in Fig. 5 for both 10K-Random CMAP and LTDS datasets, where the horizontal axis uses logarithmic scale $\lg(\lambda)$. As shown in Fig. 5, DNNs' model fit declines with increasing λ , especially in terms of log-likelihood, which is more sensitive to λ than accuracy and F_1 score. The decline in predictive power is particularly noticeable after a critical point, such as $\lambda = 10$ for DNNs with sum-PGR. In other words, although regularization would generally reduce predictive power, there exists a range of λ that almost preserves predictive power while enhancing behavioral regularity, such as $\lambda \leq 0.1$ for DNNs with sum-PGR, consistent with our findings in Section 5.1.1.

Fig. 5 also presents two important differences between log-likelihood and accuracy or F_1 score, as well as between sum- and norm-based gradient regularization. First, log-likelihood is much more sensitive than accuracy or F_1 score in measuring predictive power, which is theoretically valid and empirically expected.

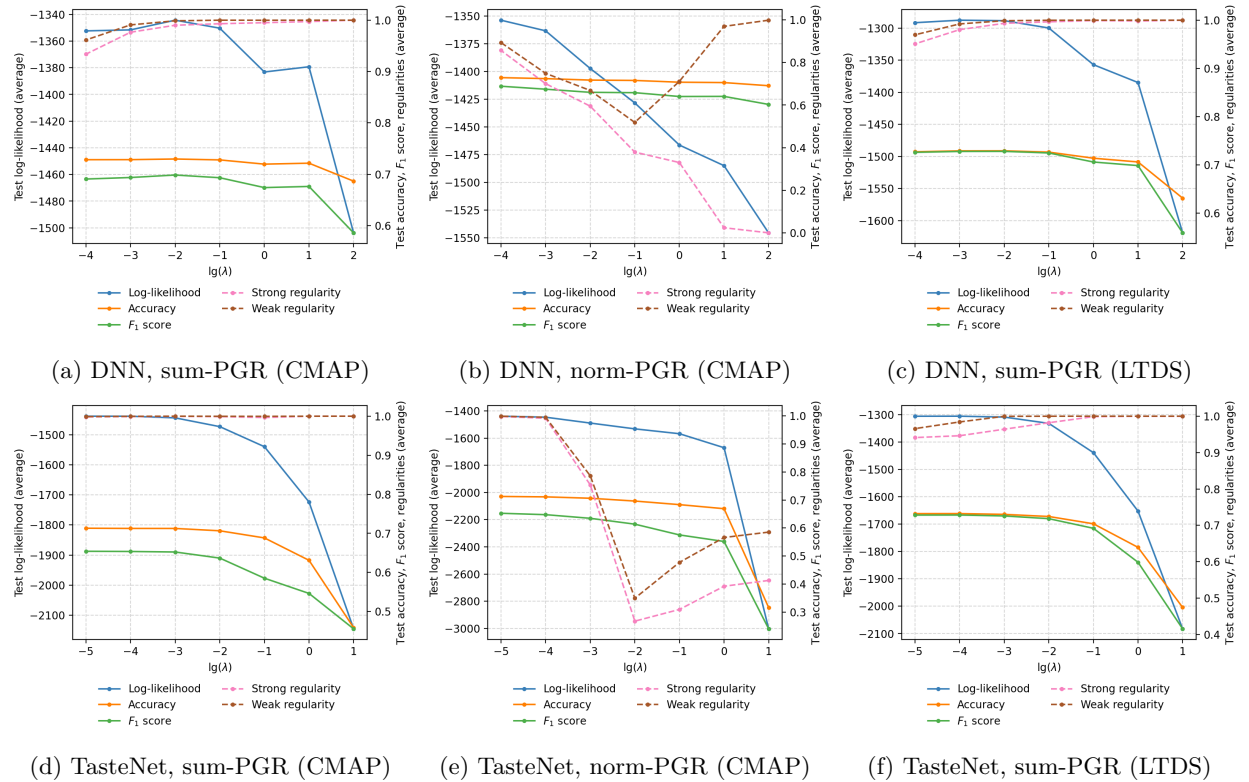


Figure 5: Effects of regularization strength (10K-Random).

Moreover, presenting accuracy and F_1 score together might be a good idea for imbalanced data. Therefore, we recommend that future studies use different metrics when comparing the performance of DCMs. Second, our behavioral regularity metrics can demonstrate the flattening effects of norm-based gradient regularization on individual demand curves, as indicated by weak regularity approaching 100% and strong regularity approaching 0 for very large λ 's (see Fig. 5b). In brief, strong regularity might be more appropriate than the weak one, at least for describing the global declining trend of demand curves, although weak behavioral regularity metric could still be important for describing local insensitivity to cost changes.

5.2.2 Small sample scenario: complementary effects

Interestingly, under a setting with relatively small sample, stronger sum-based gradient regularization can simultaneously improve predictive power and behavioral regularity, thereby advancing the Pareto frontier of different performance metrics. Fig. 6 illustrates the effects of λ on the 1K-Random datasets, where we can find a range for each gradient regularizer such that predictive power and behavioral regularity increase together. In the CMAP case, when λ increases from 10^{-4} to 0.1, sum-PGR improves the DNN's log-likelihood by 1.9% of its absolute value and strong regularity by 29.4 percentage points. This phenomenon suggests the presence of Pareto efficiency in NN models, despite the substitution effects observed in Fig. 5.

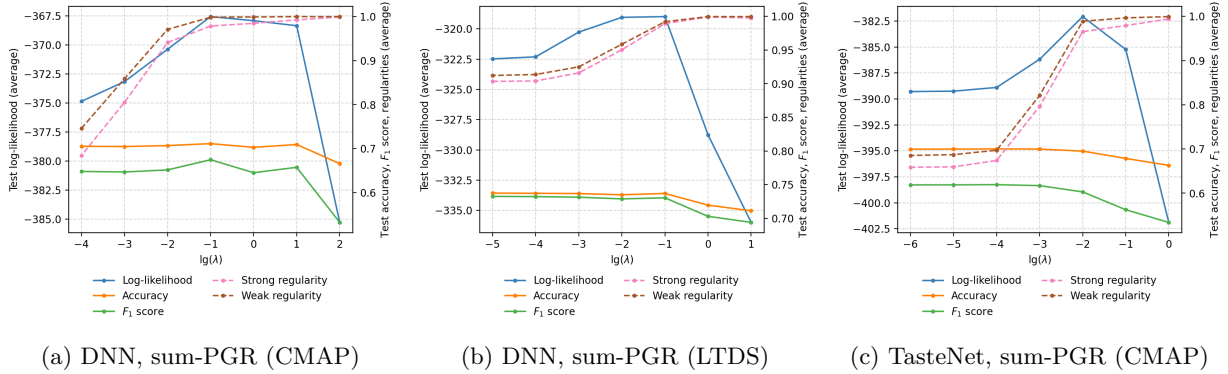


Figure 6: Effects of regularization strength (1K-Random).

On the other hand, when λ exceeds a certain critical point, we still observe substitution effects between predictive power and behavioral regularity for both DNNs and TasteNets. In the CMAP case, when λ increases from 0.1 to 100, the sum-UGR reduces the DNN's log-likelihood by 4.8% of its absolute value, but still improves the DNN's strong regularity by 2.0 percentage points. This also demonstrates the flexibility of soft constraints by identifying the optimal λ , which reflects the alignment between the data and our behavioral assumptions.

5.2.3 Out-of-domain generalization

To explore the out-of-domain generalizability of DNNs with sum-based gradient regularization, we visualize the effects of λ for the 10K-Sorted datasets in Fig. 7. Interestingly, the trend of each metric combines the characteristics of the first two scenarios: we observe both complementary and substitution effects for DNNs, whereas substitution effects are more significant for TasteNets. In addition, the log-likelihood patterns for DNNs using the CMAP and LTDS data are slightly different (see Fig. 7a and b), suggesting the data-dependency nature of NN training. On the other hand, consistent with findings from previous sections, complementary effects and Pareto efficiency are observed for the sum-based gradient regularizers. These results imply the effectiveness of gradient regularization in improving behavioral regularity and predictive power simultaneously for the NN-based choice models.

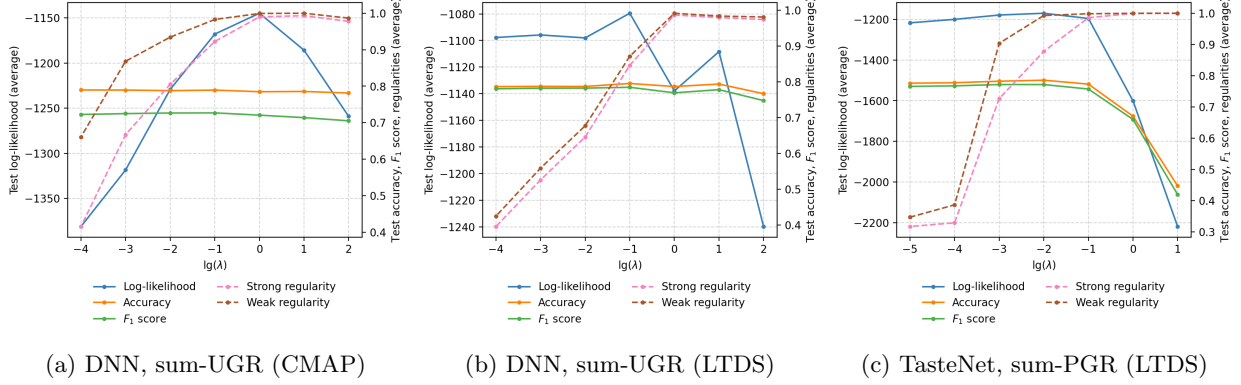


Figure 7: Effects of regularization strength (10K-Sorted).

5.3 Summary of empirical findings

As a summary, our findings demonstrate the conditions under which our sum-based gradient regularizers and behavioral regularity metrics are most effective. When the sample size is large, DNN models tend to have a high behavioral regularity. In this case, even a benchmark DNN architecture could be used for behavioral prediction and mobility policy analysis, and our behavioral regularity metrics and regularization methods might not be necessary. However, since it is always costly to collect travel survey data, the sample size in travel demand modeling is typically small (e.g., less than 10,000 samples). Under this context, sum-based gradient regularization should be broadly applied since it enables DNNs to generate reliable prediction and intuitive behavioral interpretation. Gradient regularization is also effective when the models are applied to future forecasting or cross-city transfer of mobility policies, which is emulated by the out-of-domain generalization scenario in our experiments. More generally, soft and hard constraints have their own advantages in improving behavioral regularity and predictive power, but neither is necessarily beneficial if the benchmark model already exhibits high behavioral regularity.

6 Conclusions

DNNs often present behaviorally irregular patterns that greatly limit their practical use and theoretical appeal in travel behavior analysis, especially in applications and forecasting. Nevertheless, there is no consensus on how to measure or improve the model regularity of DNNs within the field of discrete choice modeling. This paper makes contributions by developing the behavioral regularity metrics and a gradient regularization framework. Specifically, we propose the law of demand in economics as a novel measure of DNNs' behavioral regularity w.r.t. generalized costs. Using a constrained optimization framework, we design six gradient regularizers to enhance the strong and weak behavioral regularities of DNNs. Empirically, these gradient regularizers are applied to two travel survey datasets collected from Chicago and London, through which we examine the trade-off between predictive power and behavioral regularity in the small versus large sample scenarios, and in-domain versus out-of-domain generalizations. Using five evaluation metrics, we demonstrate the effectiveness of our gradient regularizers on both DNNs and TasteNets.

We find that sum-based gradient regularization can significantly improve the behavioral regularity of DNNs without sacrificing their predictive power in all the scenarios. There exists a substitution effect between predictive power and behavioral regularity in the large sample scenario, but a complementary effect in the small sample scenario. This is consistent with general understanding that, within the overparameterized regime, regularization can reduce variance (e.g., [Alpaydin, 2014](#)).⁴ Utilizing the 10K-Sorted datasets, we further find that gradient regularization is also effective for out-of-domain generalization, which is critical for

⁴To be precise, this coincides with the classical viewpoint of bias–variance trade-off. Recent studies have found evidence that over-parameterization with large data samples could potentially improve model performance (e.g., [Belkin et al., 2019](#)).

transferring knowledge across contexts. Our results also demonstrate how and why the gradient regularizers enhance predictive power and behavioral regularity, particularly for the small sample scenario and out-of-domain generalization. Comparing the soft and hard constraints, the DNNs with soft constraints can outperform the TasteNets with either soft or hard constraints, implying that TasteNets might be overly restrictive in capturing behavioral mechanisms. Besides the constrained optimization interpretation, the findings could be further understood by an analogy between gradient regularization and informative Bayesian prior. Specifically, gradient regularization can be seen as a deterministic prior imposed on the parameters estimates, that allows incorporating the modeler’s prior belief. Such prior belief can be critical especially when the sample size is limited.

This study pioneers in proposing new behavioral metrics and designing a practical regularization framework for DNNs. To address behavioral irregularity as shown in past studies (Wang et al., 2020a,b; Wong and Farooq, 2021; Xia et al., 2023), we incorporate domain-knowledge into DNNs by regularizing the gradients’ direction, in contrast to the norm-based (magnitude) regularization as in the computer science literature (Drucker and Le Cun, 1991; Jakubovitz and Gyires, 2018; Sokolić et al., 2017). With this research, strong and weak behavioral regularities could be incorporated into future behavioral analysis using deep learning, thus evaluating the consistency of models with theories in behavioral science and microeconomics. Gradient regularization is a practical framework, which can impose any prior belief of the input–output relationship on any NN architecture. The gradient regularization method provides an alternative approach to enhance behavioral regularity, in addition to designing dedicated network architectures. It integrates the computational power of deep learning with common behavioral regularity assumptions through simple gradient regularization. Hence, it provides an opportunity to develop more plausible deep learning models for transportation policy analysis. For example, researchers can use our approach to facilitate cross-city transfer of mobility policies when local governments seek to learn certain policies (e.g., congestion charging) from other areas. Our study demonstrates that deep learning with behavioral regularity could generate reliable insights for such policy transfer even when the source and the target cities have different data distributions in sociodemographics and pricing strategies.

This study mainly adopts the market-level aggregate metric for behavioral regularity and enhances it through soft constraints, despite other possible approaches. Instead of the aggregate regularity, researchers may consider individual behavioral regularity and enhance it through hard constraints (Kim and Bansal, 2024). We focus on the aggregate regularity because the metric is weaker than the individual one, thereby accommodating individual-level irrational behaviors (Bagwell and Riordan, 1991; Stiving, 2000). Empirically, the aggregate regularity and soft constraints jointly lead to higher predictive performance, implying the existence of irregular behaviors for certain individuals or price ranges. In fact, the two issues—individual versus aggregate regularity and soft versus hard constraints—have been debated for a long time in the economics and optimization fields (Becker, 1962), and we avoid taking a strong stance on either side. Nevertheless, the aggregate regularity and soft constraints form the weakest assumption among all possibilities, thus serving as an acceptable common ground for the research community.

Limitations still exist in this study. This work only regularizes direct partial derivatives, but not cross partial derivatives. Regularization on cross partial derivatives, on the other hand, could impose a stronger prior assumption, controlling the substitution and complementary patterns across alternatives. In addition, this research assumes exogenous penalty weight for gradient regularization, while future research could combine hyperparameter tuning to automatically learn these penalty weights. Lastly, as in typical deep learning models, parameter identification has been discussed (Hwang and Ding, 1997) but largely remains an open question. In traditional DCMs, parameter identification is the foundation for statistical analysis (McFadden, 1980). However, without further research into DNNs’ parameter identification, it could be challenging to quantify model uncertainty or design statistical tests, thus limiting the practicality of deep learning for discrete choice analysis. Future studies should investigate the necessary and sufficient conditions for model identification and statistical tests, especially in the over-parameterized deep learning models.

Contributions of authors

Shenhao Wang conceived of the presented idea; Shenhao Wang and Rui Yao developed the theory and reviewed previous studies; Shenhao Wang, Rui Yao, and Siqi Feng designed the experiments and discussed the results; Siqi Feng conducted the experiments and drafted the manuscript; Shenhao Wang and Rui Yao revised the manuscript; Stephane Hess, Ricardo Daziano, Timothy Brathwaite, and Joan Walker provided comments, Shenhao Wang supervised this work. All authors contributed to the final manuscript.

Acknowledgements

The authors thank the three anonymous reviewers for their constructive and insightful comments. Stephane Hess acknowledges support from the European Research Council through the advanced grant 101020940-SYNERGY.

Appendix A Summary statistics of datasets

A.1 Full datasets

Table A.1: Summary statistics of the full CMAP dataset.

Variable	Mean	SD	Min.	25%	50%	75%	Max.
Age (year)	38.95	13.30	6	29	37	47	84
Household size	2.50	1.36	1	1	2	3	12
Number of cars in the household	1.42	0.99	0	1	1	2	8
Time by car (h)	0.22	0.16	0.00	0.11	0.17	0.29	1.66
Cost by car (\$)	8.77	4.11	1.18	5.84	6.57	11.06	48.24
Time by train (h)	0.82	0.67	0.05	0.38	0.64	0.99	4.95
Cost by train (\$)	2.55	0.39	0.00	2.33	2.48	2.61	10.00
Time by active mobility (h)	1.14	1.34	0.03	0.33	0.64	1.33	21.82
Gender	11,821 (1: male);		14,278 (0: female)				
Bachelor's degree or above	18,853 (1: yes);		7,246 (0: no)				
One-person household	6,697 (1: yes);		19,402 (0: no)				
One-car household	10,004 (1: yes);		16,095 (0: no)				
Annual household income	15,520 (1: $\geq$ \$75K);		10,579 (0: $<$ \$75K)				

Table A.2: Summary statistics of the full LTDS dataset.

Variable	Mean	SD	Min.	25%	50%	75%	Max.
Number of cars in the household	0.98	0.75	0	0	1	2	2
Number of transfers in public transit	0.37	0.62	0	0	0	1	4
Time by car (h)	0.28	0.25	0.00	0.11	0.19	0.37	2.06
Cost by car (£)	1.90	3.49	0.00	0.29	0.57	1.29	17.16
Time by public transit (h)	0.47	0.31	0.01	0.23	0.39	0.64	2.73
Cost by public transit (£)	1.56	1.54	0.00	0.00	1.50	2.40	13.49
Time by active mobility (h)	0.75	0.73	0.02	0.23	0.48	1.00	5.98
Age (year)	18,917 (1: < 25); 44,957 (0: 25–55); 17,212 (2: > 55)						
Gender	38,396 (1: male); 42,690 (0: female)						
Driving license	50,035 (1: yes); 31,051 (0: no)						

A.2 10K-Sorted datasets

Table A.3: Summary statistics of the training set of 10K-Sorted CMAP.

Variable	Mean	SD	Min.	25%	50%	75%	Max.
Age (year)	38.44	13.26	13	28	36	46	83
Household size	2.49	1.36	1	1	2	3	8
Number of cars in the household	1.37	1.01	0	1	1	2	8
Time by car (h)	0.16	0.08	0.01	0.10	0.15	0.21	0.76
Cost by car (\$)	6.98	1.86	1.33	5.73	6.57	7.76	11.41
Time by train (h)	0.60	0.37	0.05	0.34	0.52	0.78	4.62
Cost by train (\$)	2.41	0.23	0.00	2.31	2.41	2.52	10.00
Time by active mobility (h)	0.60	0.40	0.04	0.28	0.51	0.84	5.93
Gender	3,104 (1: male); 3,896 (0: female)						
Bachelor’s degree or above	5,087 (1: yes); 1,913 (0: no)						
One-person household	1,830 (1: yes); 5,170 (0: no)						
One-car household	2,678 (1: yes); 4,322 (0: no)						
Annual household income (\$)	4,079 (1: $\geq$ 75K); 2,921 (0: < 75K)						

Table A.4: Summary statistics of test set of 10K-Sorted CMAP.

Variable	Mean	SD	Min.	25%	50%	75%	Max.
Age (year)	41.30	13.62	6	31	39	50	84
Household size	2.61	1.35	1	2	2	4	8
Number of cars in the household	1.67	0.96	0	1	2	2	7
Time by car (h)	0.46	0.15	0.09	0.34	0.44	0.55	1.51
Cost by car (£)	15.95	2.33	11.41	14.44	16.54	17.44	48.24
Time by train (h)	1.77	0.81	0.22	1.16	1.59	2.18	4.94
Cost by train (£)	3.10	0.41	2.00	2.83	3.15	3.33	6.65
Time by active mobility (h)	3.22	1.58	0.16	2.03	2.76	4.03	18.37
Gender	963 (1: male);		1,037 (0: female)				
Bachelor's degree or above	1,439 (1: yes);		561 (0: no)				
One-person household	436 (1: yes);		1,564 (0: no)				
One-car household	669 (1: yes);		1,331 (0: no)				
Annual household income (£)	1,309 (1: $\geq$ 75K);		691 (0: $<$ 75K)				

Table A.5: Summary statistics of the training set of 10K-Sorted LTDS.

Variable	Mean	SD	Min.	25%	50%	75%	Max.
Number of cars in the household	0.98	0.75	0	0	1	2	2
Number of transfers in public transit	0.16	0.43	0	0	0	0	3
Time by car (h)	0.21	0.18	0.01	0.09	0.15	0.27	1.45
Cost by car (£)	1.44	3.00	0.03	0.25	0.44	0.87	16.11
Time by public transit (h)	0.38	0.25	0.02	0.20	0.32	0.50	2.00
Cost by public transit (£)	0.98	0.85	0.00	0.00	1.50	1.50	2.90
Time by active mobility	0.54	0.52	0.02	0.20	0.36	0.69	5.67
Age (year)	1,746 (1: $<$ 25); 3,525 (0: 25–55); 1,729 (2: $>$ 55)						
Gender	3,296 (1: male);		3,704 (0: female)				
Driving license	4,140 (1: yes);		2,860 (0: no)				

Table A.6: Summary statistics of the test set of 10K-Sorted LTDS.

Variable	Mean	SD	Min.	25%	50%	75%	Max.
Number of cars in the household	1.04	0.72	0	1	1	2	2
Number of transfers in public transit	1.20	0.57	0	1	1	1	4
Time by car (h)	0.55	0.30	0.07	0.31	0.49	0.75	1.58
Cost by car (£)	3.89	4.64	0.21	0.97	1.63	3.47	16.28
Time by public transit (h)	0.82	0.29	0.22	0.61	0.78	0.98	2.73
Cost by public transit (£)	3.97	1.28	2.90	3.00	3.40	4.50	11.60
Time by active mobility	1.55	0.86	0.22	0.87	1.36	2.05	4.87
Age (year)	212 (1: $<$ 25); 1,636 (0: 25–55); 152 (2: $>$ 55)						
Gender	1,028 (1: male);		972 (0: female)				
Driving license	1,561 (1: yes);		439 (0: no)				

Appendix B Impacts of optimization algorithms

Using the 10K-Random CMAP dataset, we visualize the training and validation losses (Fig. B.1) as well as the corresponding individual demand functions (Fig. B.2) for the Adam, AdamW, and SGD algorithms. Based on default settings of all three algorithms, both Adam and AdamW lead to fast convergence and reasonable demand functions, whereas SGD does not converge within 100 epochs and leads to less reasonable demand functions. It is worth noting that weight decay of AdamW is set as zero to avoid further complexity in regularization, thus identifying the direct impacts of our gradient regularization on behavioral regularity.

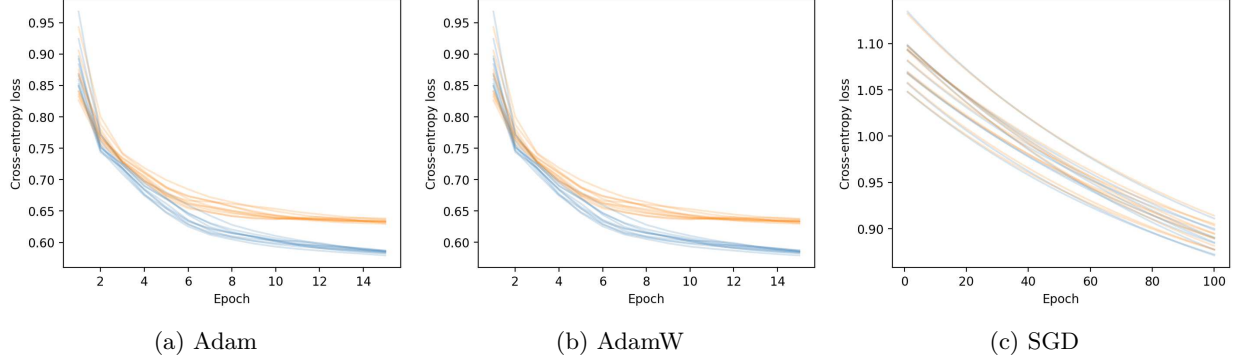


Figure B.1: Training loss (blue) and validation loss (orange) per epoch for different algorithms.

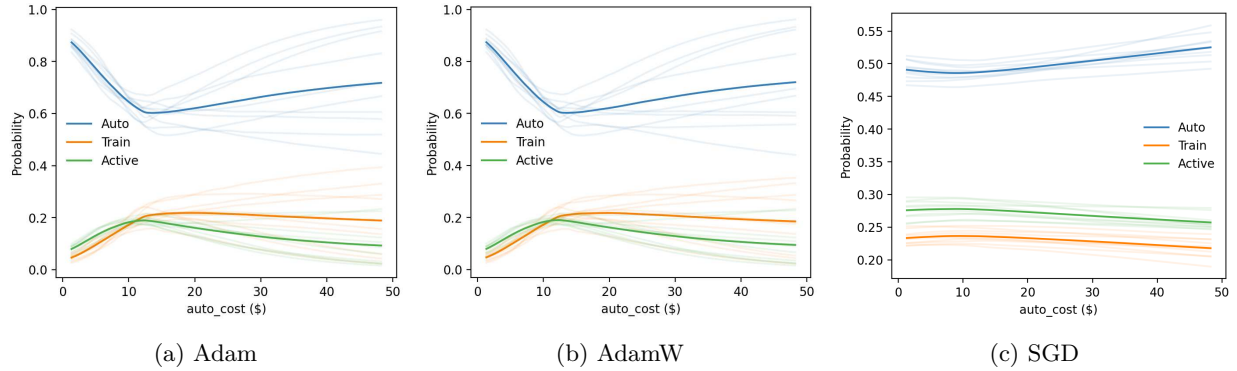


Figure B.2: Individual demands as functions of driving costs for different algorithms.

Appendix C Training and validation performance

C.1 Large sample scenario

Table C.1: Model performance in the training sets of 10K-Random.

Metric: mean (SD)	DNN				TasteNet					RUM	
	No GR	PGR	UGR	LGR	No GR	PGR	UGR	LGR	ReLU	Exp	MNL
Panel 1: CMAP data, sum-XGR											
Log-likelihood	−4272.7 (12.97)	− 4216.5 (11.74)	−4240.1 (15.40)	−4221.9 (17.58)	−4677.8 (19.89)	−4675.8 (19.38)	−4676.0 (19.57)	−4675.7 (19.24)	−4737.2 (51.18)	−4675.7 (18.97)	−4710.2 (0)
Accuracy	0.751 (0.002)	0.753 (0.002)	0.752 (0.002)	0.753 (0.002)	0.732 (0.002)	0.732 (0.002)	0.733 (0.002)	0.732 (0.002)	0.729 (0.002)	0.733 (0.002)	0.731 (0)
F ₁ score	0.717 (0.004)	0.726 (0.004)	0.721 (0.003)	0.724 (0.004)	0.675 (0.004)	0.675 (0.004)	0.675 (0.004)	0.675 (0.004)	0.671 (0.004)	0.681 (0.003)	0.681 (0)
Strong regularity	0.889 (0.061)	0.987 (0.003)	0.980 (0.011)	0.988 (0.003)	0.997 (0.003)	0.998 (0.002)	0.998 (0.002)	0.998 (0.002)	0.423 (0.315)	0.999 (0.000)	0.998 (0)
Weak regularity	0.927 (0.058)	0.999 (0.001)	0.997 (0.004)	0.999 (0.001)	0.999 (0.002)	0.999 (0.001)	0.999 (0.002)	0.999 (0.001)	1.000 (0.000)	1.000 (0.000)	1.000 (0)
Panel 2: CMAP data, norm-XGR											
Log-likelihood	− 4272.7 (13.22)	−4282.9 (12.23)	−4333.2 (14.33)	−4303.7 (11.96)	−4677.8 (19.89)	−4679.7 (19.88)	−4763.8 (17.51)	−4684.8 (19.68)	−4737.2 (51.18)	−4675.7 (18.97)	−4710.2 (0)
Accuracy	0.751 (0.002)	0.750 (0.002)	0.745 (0.002)	0.748 (0.001)	0.732 (0.002)	0.732 (0.002)	0.729 (0.002)	0.732 (0.002)	0.729 (0.002)	0.733 (0.002)	0.731 (0)
F ₁ score	0.717 (0.004)	0.713 (0.004)	0.700 (0.005)	0.706 (0.005)	0.675 (0.004)	0.674 (0.004)	0.665 (0.004)	0.673 (0.004)	0.671 (0.004)	0.681 (0.003)	0.681 (0)
Strong regularity	0.889 (0.061)	0.856 (0.063)	0.704 (0.044)	0.813 (0.063)	0.997 (0.003)	0.997 (0.003)	0.922 (0.032)	0.996 (0.005)	0.423 (0.315)	0.999 (0.000)	0.998 (0)
Weak regularity	0.927 (0.058)	0.898 (0.062)	0.761 (0.046)	0.855 (0.065)	0.999 (0.002)	0.999 (0.002)	0.935 (0.028)	0.998 (0.003)	1.000 (0.000)	1.000 (0.000)	1.000 (0)
Panel 3: LTDS data, sum-XGR											
Log-likelihood	−4427.9 (20.61)	− 4416.1 (39.09)	−4434.8 (24.41)	−4499.3 (16.00)	−4500.2 (10.52)	−4517.9 (12.03)	−4567.1 (24.14)	−4518.7 (11.93)	−4589.4 (68.97)	−4498.1 (9.146)	−4834.7 (0)
Accuracy	0.732 (0.003)	0.729 (0.004)	0.731 (0.005)	0.729 (0.004)	0.736 (0.001)	0.734 (0.002)	0.728 (0.004)	0.734 (0.002)	0.728 (0.005)	0.738 (0.001)	0.718 (0)
F ₁ score	0.731 (0.002)	0.728 (0.003)	0.730 (0.004)	0.726 (0.004)	0.732 (0.001)	0.730 (0.002)	0.723 (0.004)	0.730 (0.002)	0.724 (0.005)	0.734 (0.001)	0.714 (0)
Strong regularity	0.947 (0.032)	0.993 (0.004)	0.994 (0.009)	0.997 (0.003)	0.936 (0.023)	0.964 (0.012)	0.988 (0.009)	0.975 (0.011)	0.927 (0.046)	0.990 (0.002)	0.996 (0)
Weak regularity	0.968 (0.025)	0.999 (0.001)	0.998 (0.005)	1.000 (0.000)	0.960 (0.019)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0)

Table C.2: Model performance in the validation sets of 10K-Random.

Metric: mean (SD)	DNN				TasteNet						RUM
	No GR	PGR	UGR	LGR	No GR	PGR	UGR	LGR	ReLU	Exp	MNL
Panel 1: CMAP data, sum-XGR											
Log-likelihood	<u>−633.6</u> (2.507)	−633.0 (2.474)	−634.1 (3.594)	−634.2 (2.794)	−683.0 (2.442)	−683.1 (2.470)	−683.1 (2.482)	−683.1 (2.467)	−689.2 (6.598)	−687.1 (2.722)	−693.8 (0)
Accuracy	0.739 (0.004)	0.739 (0.004)	0.739 (0.003)	0.739 (0.003)	0.726 (0.004)	0.727 (0.003)	0.727 (0.003)	0.727 (0.003)	0.724 (0.003)	0.728 (0.004)	<u>0.737</u> (0)
F_1 score	0.702 (0.005)	0.710 (0.004)	0.705 (0.005)	<u>0.708</u> (0.003)	0.668 (0.007)	0.669 (0.006)	0.670 (0.006)	0.669 (0.006)	0.664 (0.003)	0.679 (0.004)	0.690 (0)
Strong regularity	0.879 (0.058)	0.981 (0.005)	0.973 (0.014)	0.982 (0.005)	0.997 (0.005)	<u>0.999</u> (0.002)	0.998 (0.003)	0.998 (0.002)	0.427 (0.319)	1.000 (0.001)	1.000 (0)
Weak regularity	0.928 (0.057)	<u>0.999</u> (0.001)	0.997 (0.004)	<u>0.999</u> (0.001)	0.998 (0.003)	<u>0.999</u> (0.002)	<u>0.999</u> (0.002)	<u>0.999</u> (0.002)	1.000 (0.000)	1.000 (0.000)	1.000 (0)
Panel 2: CMAP data, norm-XGR											
Log-likelihood	<u>−633.6</u> (2.507)	−633.3 (2.414)	−635.9 (2.676)	−633.3 (2.989)	−683.0 (2.442)	−683.0 (2.421)	−687.4 (2.236)	−682.9 (2.366)	−689.2 (6.598)	−687.1 (2.722)	−693.8 (0)
Accuracy	<u>0.739</u> (0.004)	0.741 (0.004)	0.741 (0.003)	0.741 (0.003)	0.726 (0.004)	0.725 (0.004)	0.720 (0.001)	0.725 (0.004)	0.724 (0.003)	0.728 (0.004)	0.737 (0)
F_1 score	0.702 (0.005)	0.702 (0.005)	0.696 (0.006)	<u>0.699</u> (0.005)	0.668 (0.007)	0.666 (0.007)	0.655 (0.003)	0.665 (0.006)	0.664 (0.003)	0.679 (0.004)	0.690 (0)
Strong regularity	0.879 (0.058)	0.846 (0.058)	0.701 (0.044)	0.803 (0.061)	<u>0.997</u> (0.005)	0.996 (0.005)	0.924 (0.03)	0.995 (0.007)	0.427 (0.319)	1.000 (0.001)	1.000 (0)
Weak regularity	0.928 (0.057)	0.900 (0.057)	0.766 (0.046)	0.858 (0.062)	<u>0.998</u> (0.003)	<u>0.998</u> (0.003)	0.935 (0.026)	0.997 (0.006)	1.000 (0.000)	1.000 (0.000)	1.000 (0)
Panel 3: LTDS data, sum-XGR											
Log-likelihood	−645.5 (4.399)	−642.8 (5.709)	−644.2 (4.345)	−647.8 (3.915)	−638.8 (1.949)	<u>−639.9</u> (2.182)	−645.3 (3.254)	<u>−639.9</u> (1.985)	−654.1 (11.75)	−640.6 (2.157)	−670.0 (0)
Accuracy	0.721 (0.002)	0.722 (0.003)	0.725 (0.002)	0.723 (0.005)	<u>0.729</u> (0.003)	0.726 (0.002)	0.726 (0.002)	0.726 (0.002)	0.720 (0.006)	0.726 (0.003)	0.735 (0)
F_1 score	0.719 (0.003)	0.721 (0.003)	0.722 (0.002)	0.718 (0.004)	<u>0.723</u> (0.003)	0.720 (0.002)	0.719 (0.003)	0.720 (0.002)	0.714 (0.007)	0.721 (0.003)	0.730 (0)
Strong regularity	0.947 (0.032)	<u>0.992</u> (0.005)	0.991 (0.012)	0.996 (0.004)	0.924 (0.026)	0.953 (0.014)	0.982 (0.011)	0.964 (0.014)	0.916 (0.050)	0.982 (0.003)	0.991 (0)
Weak regularity	0.971 (0.025)	<u>0.999</u> (0.001)	0.997 (0.007)	1.000 (0.001)	0.957 (0.023)	1.000 (0.001)	1.000 (0.000)	1.000 (0.001)	1.000 (0.000)	1.000 (0.000)	1.000 (0)

C.2 Small sample scenario

Table C.3: Model performance in the training sets of 1K-Random.

Metric: mean (SD)	DNN				TasteNet						RUM
	No GR	PGR	UGR	LGR	No GR	PGR	UGR	LGR	ReLU	Exp	MNL
Panel 1: CMAP data, sum-XGR											
Log-likelihood	−494.8 (6.074)	<u>−511.0</u> (8.266)	−546.4 (9.391)	−526.2 (8.871)	−555.6 (4.071)	−580.2 (6.789)	−570.9 (5.316)	−583.1 (9.196)	−563.0 (4.542)	−558.1 (6.232)	−548.1 (0)
Accuracy	0.740 (0.005)	<u>0.732</u> (0.011)	0.691 (0.008)	0.718 (0.01)	0.711 (0.006)	0.691 (0.007)	0.696 (0.008)	0.690 (0.007)	0.708 (0.007)	0.718 (0.006)	0.715 (0)
F_1 score	0.690 (0.005)	<u>0.682</u> (0.019)	0.578 (0.024)	0.648 (0.029)	0.634 (0.013)	0.575 (0.018)	0.588 (0.020)	0.573 (0.018)	0.623 (0.012)	0.652 (0.008)	0.663 (0)
Strong regularity	0.649 (0.175)	0.983 (0.006)	0.986 (0.013)	0.984 (0.008)	0.624 (0.120)	0.981 (0.015)	0.981 (0.013)	0.984 (0.011)	0.450 (0.185)	1.000 (0.000)	<u>0.999</u> (0)
Weak regularity	0.718 (0.165)	<u>0.999</u> (0.002)	0.995 (0.007)	0.998 (0.003)	0.657 (0.125)	<u>0.999</u> (0.001)	0.997 (0.003)	0.996 (0.004)	1.000 (0.000)	1.000 (0.000)	1.000 (0)
Panel 2: LTDS data, sum-XGR											
Log-likelihood	−506.5 (4.002)	−542.9 (11.91)	<u>−517.8</u> (6.636)	−560.6 (13.342)	−548.0 (3.359)	−553.0 (3.389)	−565.9 (5.310)	−553.5 (3.395)	−562.1 (10.07)	−541.0 (3.794)	−570.2 (0)
Accuracy	0.732 (0.005)	0.707 (0.010)	<u>0.726</u> (0.006)	0.693 (0.02)	0.705 (0.005)	0.703 (0.006)	0.691 (0.007)	0.702 (0.005)	0.703 (0.007)	0.720 (0.007)	0.709 (0)
F_1 score	0.727 (0.005)	0.694 (0.014)	0.720 (0.007)	0.673 (0.032)	0.691 (0.006)	0.689 (0.007)	0.672 (0.008)	0.688 (0.006)	0.690 (0.007)	<u>0.714</u> (0.007)	0.701 (0)
Strong regularity	0.898 (0.072)	<u>0.999</u> (0.002)	0.991 (0.015)	<u>0.999</u> (0.003)	0.946 (0.023)	0.968 (0.013)	0.998 (0.002)	0.970 (0.009)	0.931 (0.047)	0.995 (0.003)	1.000 (0)
Weak regularity	0.906 (0.072)	1.000 (0.001)	0.994 (0.012)	<u>0.999</u> (0.003)	0.950 (0.021)	0.988 (0.011)	<u>0.999</u> (0.001)	0.986 (0.01)	1.000 (0.000)	1.000 (0.000)	1.000 (0)

Table C.4: Model performance in the validation sets of 1K-Random.

Metric: mean (SD)	DNN				TasteNet						RUM
	No GR	PGR	UGR	LGR	No GR	PGR	UGR	LGR	ReLU	Exp	MNL
Panel 1: CMAP data, sum-XGR											
Log-likelihood	−151.9 (1.033)	−147.1 (1.993)	−149.2 (1.876)	<u>−147.6</u> (2.548)	−159.2 (1.725)	−155.3 (2.055)	−155.5 (1.920)	−155.6 (2.188)	−157.8 (2.613)	−160.4 (2.203)	−163.0 (0)
Accuracy	0.680 (0.006)	0.692 (0.008)	0.677 (0.008)	0.686 (0.008)	0.680 (0.011)	0.686 (0.012)	0.682 (0.012)	0.684 (0.014)	0.692 (0.010)	<u>0.697</u> (0.012)	0.700 (0)
F_1 score	0.628 (0.006)	<u>0.638</u> (0.018)	0.561 (0.016)	0.610 (0.024)	0.599 (0.019)	0.574 (0.027)	0.579 (0.026)	0.571 (0.030)	0.606 (0.017)	0.637 (0.017)	0.654 (0)
Strong regularity	0.671 (0.169)	0.984 (0.009)	0.990 (0.015)	0.988 (0.015)	0.692 (0.119)	0.982 (0.018)	0.980 (0.010)	0.985 (0.012)	0.468 (0.208)	<u>0.994</u> (0.003)	0.995 (0)
Weak regularity	0.746 (0.154)	1.000 (0.002)	0.996 (0.007)	<u>0.999</u> (0.003)	0.721 (0.119)	0.997 (0.003)	0.996 (0.005)	0.996 (0.004)	1.000 (0.000)	1.000 (0.000)	1.000 (0)
Panel 2: LTDS data, sum-XGR											
Log-likelihood	−143.3 (2.016)	<u>−138.9</u> (1.676)	−139.2 (1.338)	−140.5 (3.175)	−144.5 (0.741)	−141.7 (0.958)	−143.4 (1.105)	−141.9 (1.085)	−145.3 (2.642)	−139.1 (1.596)	−138.7 (0)
Accuracy	0.688 (0.008)	0.681 (0.015)	0.693 (0.015)	0.680 (0.019)	0.697 (0.009)	<u>0.705</u> (0.009)	0.698 (0.010)	<u>0.705</u> (0.009)	0.694 (0.013)	0.727 (0.014)	0.700 (0)
F_1 score	0.683 (0.009)	0.663 (0.019)	0.685 (0.015)	0.653 (0.031)	0.676 (0.009)	0.684 (0.010)	0.676 (0.011)	0.684 (0.01)	0.674 (0.013)	0.719 (0.014)	<u>0.689</u> (0)
Strong regularity	0.890 (0.073)	<u>0.998</u> (0.003)	0.988 (0.021)	0.997 (0.007)	0.949 (0.027)	0.970 (0.015)	<u>0.998</u> (0.003)	0.972 (0.007)	0.937 (0.044)	1.000 (0.000)	1.000 (0)
Weak regularity	0.896 (0.073)	<u>0.999</u> (0.002)	0.992 (0.016)	0.997 (0.006)	0.951 (0.025)	0.984 (0.012)	0.998 (0.003)	0.982 (0.012)	1.000 (0.000)	1.000 (0.000)	1.000 (0)

C.3 Out-of-domain generalization

Table C.5: Model performance in the training sets of 10K-Sorted.

Metric: mean (SD)	DNN				TasteNet					RUM	
	No GR	PGR	UGR	LGR	No GR	PGR	UGR	LGR	ReLU	Exp	MNL
Panel 1: CMAP data, sum-XGR											
Log-likelihood	-4292.7 (21.52)	<u>-4325.0</u> (17.68)	-4398.3 (13.50)	-4350.1 (20.77)	-4847.9 (28.64)	-4600.0 (9.302)	-4595.5 (9.634)	-4590.9 (11.26)	-4664.2 (60.61)	-4572.7 (8.131)	-4720.0 (0)
Accuracy	0.745 (0.002)	<u>0.742</u> (0.002)	0.738 (0.002)	0.741 (0.002)	0.715 (0.001)	0.726 (0.001)	0.727 (0.001)	0.727 (0.001)	0.723 (0.003)	0.730 (0.001)	0.726 (0)
F_1 score	0.717 (0.003)	<u>0.714</u> (0.003)	0.707 (0.002)	0.713 (0.003)	0.661 (0.002)	0.691 (0.002)	0.692 (0.002)	0.692 (0.001)	0.684 (0.006)	0.698 (0.001)	0.693 (0)
Strong regularity	0.881 (0.045)	0.998 (0.001)	0.998 (0.001)	0.998 (0.001)	1.000 (0.000)	1.000 (0.001)	<u>0.999</u> (0.001)	<u>0.999</u> (0.001)	0.952 (0.141)	1.000 (0.000)	1.000 (0)
Weak regularity	0.898 (0.042)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.001)	1.000 (0.001)	<u>0.999</u> (0.001)	1.000 (0.000)	1.000 (0.000)	1.000 (0)
Panel 2: LTDS data, sum-XGR											
Log-likelihood	<u>-4554.5</u> (11.10)	-4589.4 (25.82)	-4806.0 (71.16)	-4530.7 (6.477)	-4661.6 (7.997)	-4785.8 (41.15)	-4724.6 (21.08)	-4676.5 (8.192)	-4752.8 (75.76)	-4663.7 (8.416)	-4919.2 (0)
Accuracy	0.725 (0.002)	0.722 (0.002)	0.710 (0.005)	0.725 (0.002)	<u>0.724</u> (0.001)	0.712 (0.005)	0.717 (0.003)	0.721 (0.001)	0.716 (0.004)	<u>0.724</u> (0.001)	0.712 (0)
F_1 score	0.722 (0.002)	0.719 (0.002)	0.706 (0.006)	0.722 (0.002)	<u>0.720</u> (0.001)	0.706 (0.006)	0.713 (0.003)	0.717 (0.001)	0.712 (0.005)	<u>0.720</u> (0.001)	0.708 (0)
Strong regularity	0.736 (0.06)	0.999 (0.001)	0.991 (0.019)	0.991 (0.005)	0.645 (0.07)	0.991 (0.008)	0.991 (0.006)	0.968 (0.023)	0.430 (0.172)	<u>0.998</u> (0.000)	0.997 (0)
Weak regularity	0.751 (0.054)	<u>0.999</u> (0.0)	0.993 (0.016)	0.994 (0.002)	0.676 (0.067)	1.000 (0.000)	<u>0.999</u> (0.001)	0.989 (0.005)	1.000 (0.000)	1.000 (0.000)	1.000 (0)

Table C.6: Model performance in the validation sets of 10K-Sorted.

Metric: mean (SD)	DNN				TasteNet						RUM
	No GR	PGR	UGR	LGR	No GR	PGR	UGR	LGR	ReLU	Exp	MNL
Panel 1: CMAP data, sum-XGR											
Log-likelihood	−656.6 (4.636)	−647.9 (3.430)	−648.5 (2.176)	− 646.9 (2.397)	−683.0 (3.344)	−660.1 (1.171)	−660.0 (1.316)	−660.2 (1.320)	−665.1 (5.144)	−671.2 (1.098)	−673.2 (0)
Accuracy	0.737 (0.003)	0.739 (0.002)	0.737 (0.004)	0.739 (0.003)	0.729 (0.004)	<u>0.745</u> (0.003)	0.746 (0.002)	<u>0.745</u> (0.002)	0.744 (0.003)	0.741 (0.004)	<u>0.745</u> (0)
F ₁ score	0.703 (0.004)	0.707 (0.002)	0.704 (0.005)	0.707 (0.005)	0.675 (0.005)	0.709 (0.004)	<u>0.710</u> (0.003)	0.709 (0.003)	0.704 (0.004)	0.707 (0.005)	0.711 (0)
Strong regularity	0.883 (0.045)	0.998 (0.001)	0.998 (0.001)	0.998 (0.001)	1.000 (0.000)	1.000 (0.001)	1.000 (0.001)	1.000 (0.001)	0.952 (0.142)	<u>0.999</u> (0.000)	1.000 (0)
Weak regularity	0.898 (0.042)	<u>0.999</u> (0.001)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.001)	1.000 (0.001)	1.000 (0.001)	1.000 (0.000)	1.000 (0.000)	1.000 (0)
Panel 2: LTDS data, sum-XGR											
Log-likelihood	−698.5 (2.201)	−699.7 (2.208)	−713.6 (4.791)	− 696.3 (2.877)	−720.3 (2.160)	−726.6 (4.988)	−722.7 (3.051)	−720.4 (2.274)	−721.9 (2.608)	−719.4 (1.861)	−730.2 (0)
Accuracy	0.691 (0.004)	0.693 (0.005)	0.691 (0.004)	0.694 (0.004)	0.701 (0.003)	0.686 (0.008)	0.694 (0.005)	<u>0.699</u> (0.003)	0.698 (0.004)	0.697 (0.002)	0.687 (0)
F1 score	0.688 (0.004)	0.690 (0.005)	0.688 (0.004)	0.692 (0.004)	0.697 (0.003)	0.680 (0.009)	0.689 (0.005)	<u>0.695</u> (0.003)	0.693 (0.004)	0.693 (0.002)	0.683 (0)
Strong regularity	0.724 (0.062)	0.998 (0.001)	0.991 (0.020)	0.990 (0.004)	0.637 (0.070)	0.992 (0.009)	0.992 (0.006)	0.967 (0.026)	0.420 (0.168)	<u>0.997</u> (0.001)	0.996 (0)
Weak regularity	0.739 (0.054)	<u>0.999</u> (0.001)	0.992 (0.018)	0.992 (0.003)	0.668 (0.066)	1.000 (0.001)	<u>0.999</u> (0.001)	0.988 (0.007)	1.000 (0.000)	1.000 (0.000)	1.000 (0)

Appendix D Effects of regularization strength

Fig. D.1 visualizes the effects of regularization strength λ on individual demand functions.

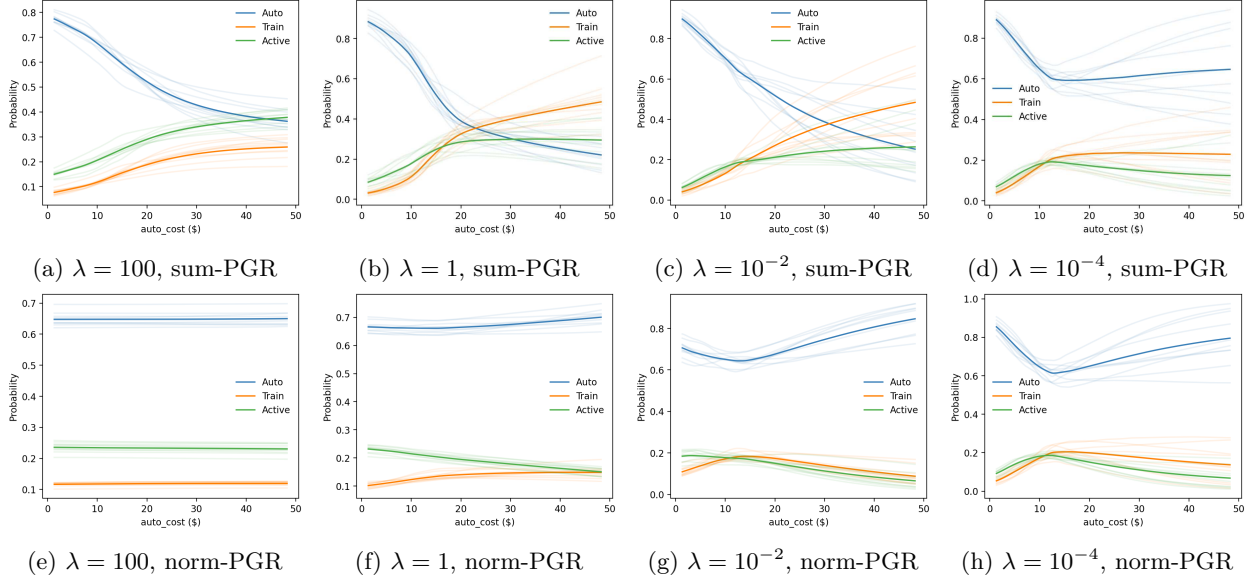


Figure D.1: Individual demands as functions of driving costs (10K-Random, CMAP).

Appendix E Sensitivity test of the behavioral regularity metric

The trend of behavioral regularity remains stable when ε changes. As shown in Fig. E.1, a sensitivity test shows that the regularity metric increases with ε . This is because a smaller ε represents a tighter upper bound of being behaviorally regular, which leads to lower behavioral regularity. The trend of the metric with λ is generally not affected by the value of ε , unless we choose a negative number with a relatively large absolute value, indicating that the modeler's assumption on behavioral regularity is too strong. Since we are more concerned about the metric trend, we can manually set suitable ε 's to distinguish between strong and weak regularities.

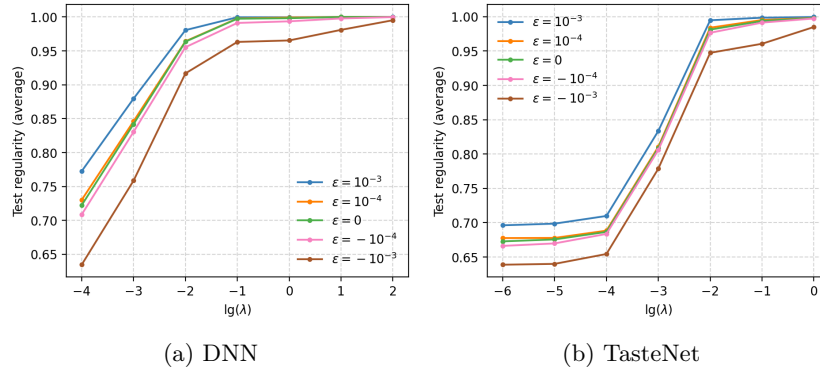


Figure E.1: Behavioral regularity as a function of ε (1K-Random, CMAP).

References

Alpaydin, E., 2014. Introduction to Machine Learning. MIT Press.

- Alwosheel, A., Van Cranenburgh, S., Chorus, C.G., 2019. ‘Computer says no’ is not enough: Using prototypical examples to diagnose artificial neural networks for discrete choice analysis. *Journal of Choice Modelling* 33, 100186.
- Archer, N.P., Wang, S., 1993. Application of the back propagation neural network algorithm with monotonicity constraints for two-group classification problems. *Decision Sciences* 24, 60–75.
- Bagwell, K., Riordan, M.H., 1991. High and declining prices signal product quality. *The American Economic Review* 81, 224–239.
- Becker, G.S., 1962. Irrational behavior and economic theory. *Journal of Political Economy* 70, 1–13.
- Belkin, M., Hsu, D., Ma, S., Mandal, S., 2019. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences* 116, 15849–15854.
- Ben-Akiva, M.E., Lerman, S.R., 1985. *Discrete choice analysis: Theory and application to travel demand*. MIT Press.
- Boyd, S.P., Vandenberghe, L., 2004. *Convex optimization*. Cambridge University Press.
- Chiappori, P.A., 1985. Distribution of income and the “law of demand”. *Econometrica* 53, 109–127.
- van Cranenburgh, S., Wang, S., Vij, A., Pereira, F., Walker, J., 2022. Choice modelling in the age of machine learning - Discussion paper. *Journal of Choice Modelling* 42, 100340.
- Daniels, H., Velikova, M., 2010. Monotone and partially monotone neural networks. *IEEE Transactions on Neural Networks* 21, 906–917.
- Di, X., Liu, H.X., 2016. Boundedly rational route choice behavior: A review of models and methodologies. *Transportation Research Part B: Methodological* 85, 142–179.
- van Dis, E.A., Bollen, J., Zuidema, W., van Rooij, R., Bockting, C.L., 2023. ChatGPT: Five priorities for research. *Nature* 614, 224–226.
- Drucker, H., Le Cun, Y., 1991. Double backpropagation increasing generalization performance, in: *IJCNN-91-Seattle International Joint Conference on Neural Networks*, IEEE. pp. 145–150.
- Dugas, C., Bengio, Y., Bélisle, F., Nadeau, C., Garcia, R., 2009. Incorporating functional knowledge in neural networks. *Journal of Machine Learning Research* 10, 1239–1262.
- Gupta, A., Shukla, N., Marla, L., Kolbeinsson, A., Yellepeddi, K., 2019. How to incorporate monotonicity in deep networks while preserving flexibility? *arXiv preprint arXiv:1909.10662*.
- Haj-Yahia, S., Mansour, O., Toledo, T., 2023. Incorporating domain knowledge in deep neural networks for discrete choice models. *arXiv preprint arXiv:2306.00016*.
- Han, Y., Pereira, F.C., Ben-Akiva, M., Zengras, C., 2022. A neural-embedded discrete choice model: Learning taste representation with strengthened interpretability. *Transportation Research Part B: Methodological* 163, 166–186.
- Härdle, W., Hildenbrand, W., Jerison, M., 1991. Empirical evidence on the law of demand. *Econometrica* 59, 1525–1549.
- Hildenbrand, W., 1983. On the “law of demand”. *Econometrica* 51, 997–1019.
- Hillel, T., Elshafie, M.Z., Jin, Y., 2018. Recreating passenger mode choice-sets for transport simulation: A case study of London, UK. *Proceedings of the Institution of Civil Engineers – Smart Infrastructure and Construction* 171, 29–42.
- Hwang, J.G., Ding, A.A., 1997. Prediction intervals for artificial neural networks. *Journal of the American Statistical Association* 92, 748–757.
- Jakubovitz, D., Giryas, R., 2018. Improving DNN robustness to adversarial attacks using Jacobian regularization, in: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 514–529.
- Kahneman, D., Tversky, A., 1979. Prospect theory: An analysis of decision under risk. *Econometrica* 47, 263–292.
- Kim, E.J., Bansal, P., 2024. A new flexible and partially monotonic discrete choice model. *Transportation Research Part B: Methodological* 183, 102947.
- Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Knez, P., Smith, V.L., Williams, A.W., 1985. Individual rationality, market rationality, and value estimation. *The American Economic Review* 75, 397–402.
- Lawson, C.L., Hanson, R.J., 1995. *Solving least squares problems*. Society for Industrial and Applied Mathematics.
- LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. *nature* 521, 436–444.
- Lichtenstein, S., Slovic, P., 1971. Reversals of preference between bids and choices in gambling decisions. *Journal of Experimental Psychology* 89, 46–55.
- Liu, J., Shen, Z., He, Y., Zhang, X., Xu, R., Yu, H., Cui, P., 2021. Towards out-of-distribution generalization: A

- survey. arXiv preprint arXiv:2108.13624.
- Loshchilov, I., Hutter, F., 2017. Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101.
- Lyu, C., Huang, K., Liang, H.N., 2015. A unified gradient regularization family for adversarial examples, in: 2015 IEEE International Conference on Data Mining, IEEE. pp. 301–309.
- May, A., 1992. Road pricing: An international perspective. *Transportation* 19, 313–333.
- McFadden, D., 1974. The measurement of urban travel demand. *Journal of Public Economics* 3, 303–328.
- McFadden, D., 1980. Econometric models for probabilistic choice among products. *Journal of Business* 53, S13–S29.
- Neumann, K., Rolf, M., Steil, J.J., 2013. Reliable integration of continuous constraints into extreme learning machines. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 21, 35–50.
- Ororbia II, A.G., Kifer, D., Giles, C.L., 2017. Unifying adversarial training algorithms with data gradient regularization. *Neural computation* 29, 867–887.
- Pan, S.J., Yang, Q., 2009. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering* 22, 1345–1359.
- Quah, J.K.H., 2000. The monotonicity of individual and market demand. *Econometrica* 68, 911–930.
- Quiñonero-Candela, J., Sugiyama, M., Schwaighofer, A., Lawrence, N.D., 2008. Dataset shift in machine learning. MIT Press.
- Ross, A., Doshi-Velez, F., 2018. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 1660–1669.
- Siffringer, B., Lurkin, V., Alahi, A., 2020. Enhancing discrete choice models with representation learning. *Transportation Research Part B: Methodological* 140, 236–261.
- Sill, J., 1997. Monotonic networks. *Advances in Neural Information Processing Systems* 10, 661–667.
- Sill, J., Abu-Mostafa, Y., 1996. Monotonicity hints. *Advances in neural information processing systems* 9, 634–640.
- Simon, H.A., 1957. *Models of man; Social and rational*. Wiley.
- Sokolić, J., Giryes, R., Sapiro, G., Rodrigues, M.R., 2017. Robust large margin deep neural networks. *IEEE Transactions on Signal Processing* 65, 4265–4280.
- Souche, S., 2010. Measuring the structural determinants of urban travel demand. *Transport Policy* 17, 127–134.
- Stiving, M., 2000. Price-endings when prices signal quality. *Management Science* 46, 1617–1629.
- Tversky, A., Kahneman, D., 1992. Advances in prospect theory: Cumulative representation of uncertainty. *Journal of Risk and Uncertainty* 5, 297–323.
- Wang, M., Deng, W., 2018. Deep visual domain adaptation: A survey. *Neurocomputing* 312, 135–153.
- Wang, S., Mo, B., Zhao, J., 2020a. Deep neural networks for choice analysis: Architecture design with alternative-specific utility functions. *Transportation Research Part C: Emerging Technologies* 112, 234–251.
- Wang, S., Mo, B., Zhao, J., 2021. Theory-based residual neural networks: A synergy of discrete choice models and deep neural networks. *Transportation Research Part B: Methodological* 146, 333–358.
- Wang, S., Wang, Q., Zhao, J., 2020b. Deep neural networks for choice analysis: Extracting complete economic information for interpretation. *Transportation Research Part C: Emerging Technologies* 118, 102701.
- Watling, D.P., Rasmussen, T.K., Prato, C.G., Nielsen, O.A., 2018. Stochastic user equilibrium with a bounded choice model. *Transportation Research Part B: Methodological* 114, 254–280.
- Wehenkel, A., Louppe, G., 2019. Unconstrained monotonic neural networks. *Advances in Neural Information Processing Systems* 32, 1545–1555.
- Wong, M., Farooq, B., 2021. Reslogit: A residual neural network logit model for data-driven choice modelling. *Transportation Research Part C: Emerging Technologies* 126, 103050.
- Xia, Y., Chen, H., Zimmermann, R., 2023. A random effect bayesian neural network (RE-BNN) for travel mode choice analysis across multiple regions. *Travel Behaviour and Society* 30, 118–134.
- Yang, H., Bell, M.G., 1997. Traffic restraint, road pricing and network equilibrium. *Transportation Research Part B: Methodological* 31, 303–314.
- Yao, E., Morikawa, T., 2005. A study of on integrated intercity travel demand model. *Transportation Research Part A: Policy and Practice* 39, 367–381.
- You, S., Ding, D., Canini, K., Pfeifer, J., Gupta, M., 2017. Deep lattice networks and partial monotonic functions. *Advances in Neural Information Processing Systems* 30.
- Zhao, X., Yan, X., Yu, A., Van Hentenryck, P., 2020. Prediction and behavioral analysis of travel mode choice: A comparison of machine learning and logit models. *Travel Behaviour and Society* 20, 22–35.

- Zheng, Y., Wang, S., Zhao, J., 2021. Equality of opportunity in travel behavior prediction with deep neural networks and discrete choice models. *Transportation Research Part C: Emerging Technologies* 132, 103410.
- Zheng, Y., Xu, Z., Xiao, A., 2023. Deep learning in economics: A systematic and critical review. *Artificial Intelligence Review* 56, 9497–9539.