Report for Discrete Choice Model and Credit Card Offers

Yufei Zhao
yuffiezhao@hotmail.com, https://github.com/yuffiezhaohk/yuffiezhao

This report details the implementation and critical analysis of Zhang et al. (2025). The rest is organized as follows. In Section 1, I review the literature of potential methods for solving the problem. Next, I propose my method implemented in Section 2. In Section 3, I present the answers to the questions in Part 1. Finally, I provide the testing plans and testing results in Section 4.

1 Literature Review

In recent years, there has been a significant body of work utilizing neural networks to approximate customers' choice probabilities, primarily aiming to mitigate restrictive assumptions like the Independence from Irrelevant Alternatives (IIA) property of the Multinomial Logit (MNL) model. For instance, Rosenfeld et al. (2020) adopt a permutation-invariant neural network, which can be viewed as an edge-free counterpart of a Graph Convolutional Network (GCN) to extract features dependent on the provided product set, thereby allowing resulting choice probabilities to also depend on that set. Following this methodology, Aouad and Désir (2025) incorporate a sample-average approximation scheme to ensure the resulting approximation remains consistent with random utility theory. Further advancements focus on capturing cross-product interactions: Wang et al. (2023) introduce the attention mechanism, while Liu et al. (2025) adopt a variant of GCN. Li et al. (2025a) provide a general framework based on GCN to use neural networks to represent various choice models, but focus on estimating assortment optimization problems rather than the choice probabilities. Li et al. (2025b) propose a model-free reinforcement learning approach based on Deep Neural Network (DNN) for online assortment customization.

2 DeepHalo

To replicate the numerical study of synthetic data with high-order effect, I use the DeepHalo under the featureless setting in Zhang et al. (2025). Specifically, I use the framework in Section 4.2. Let e_S be the indicator vector of choice set S. Let $\Theta^1 \in \mathbb{R}^{J' \times J}$, $\Theta^l \in \mathbb{R}^{J' \times J'}$ for $l = 2, \ldots, L$ and $W^l \in \mathbb{R}^{J \times J'}$. Then with $y^0 = e_S \in \mathbb{R}^J$, the recursion of quadratic activation is

$$\begin{cases} y^1 = \Theta^1 e_S \\ y^l = y^{l-1} + \Theta^l (y^{l-1} \odot y^{l-1}), \quad l = 2, \ldots, L, \end{cases}$$

and the utility is given by $u_j(S) = y^L_j$.

3 Answers to Questions in Part 1

a. Are there any errors or unclear parts in the paper? If there are, please point them out and explain the unclear parts in the report.

The following notations in the paper are unclear or wrong.

1. In Eq. (3), $u_j(x_j, X_R)$ should be written as $u_j\left(X_{R\cup\{j\}}\right)$.
2. Eq. (9) holds only for $j \in S$. Then for the featureless setting, $z_j^0 = 1_j^S$.
3. In A.2.1, the proof should be

$$y^1 = \sum_{j=1}^{J} 1_j^S + \sum_{j=1}^{J}\sum_{k\in S}\Theta_{jk}^1 e_j = \sum_{j=1}^{J} z_j^0 + \sum_{k\in S}\left(\sum_{j=1}^{J}\Theta_{jk}^1 e_j\right) = y^0 + \sum_{k\in S}\Theta^1 e_k = y^0 + \Theta^1 e_S = y^0 + \Theta^1(y^0 \odot e_S).$$

b. Do you get the same results as their synthetic data test?

I cannot get the exact same numerical results due to the limit of the computation. A simplified version for fast testing and stability checks, running a total of 100 epochs with validation performed every 50 epochs in Table 1.

Table 1 Complete Experimental Results Across Budgets and Depths

Budget	Depth	Width	Epoch 50 RMSE	Epoch 100 RMSE	Final RMSE
200,000	3	306	0.05138	0.04102	0.04102
	4	251	0.04458	0.03607	0.03607
	5	218	0.04139	0.03530	0.03530
	6	196	0.04022	0.03581	0.03581
	7	179	0.25862	0.25862	0.25862
500,000	3	490	0.04489	0.03333	0.03333
	4	401	0.03710	0.03346	0.03346
	5	348	0.03606	0.04095	0.04095
	6	312	0.03826	0.05059	0.05059
	7	285	0.03961	0.05086	0.05086

1. Impact of Depth Under Constrained Budget ($\mathcal{B} = 200k$): Under the constrained parameter budget, model performance demonstrates a clear convex relationship with network depth.
   - Optimal Range: Performance improves consistently as depth increases from 3 to 5 layers. The model achieves its global minimum RMSE of 0.03530 at Depth 5, suggesting that moderate depth is essential for capturing higher-order interaction effects when parameter capacity is limited.
   - Convergence Failure: A significant degradation is observed at Depth 7 (RMSE: 0.25862), indicating a "convergence cliff." This suggests that distributing a fixed budget across too many layers results in layers that are too narrow to propagate gradients effectively, leading to training instability.

2. Impact of Budget Scaling ($\mathcal{B} = 500k$): Scaling the parameter budget to 500,000 reveals diminishing returns for deeper architectures.
   - Efficiency in Shallow Networks: Shallower networks ($D = 3, 4$) benefit most significantly from the increased budget, with Depth 3 achieving the best overall performance in this bracket (RMSE: 0.03333).
   - Overfitting in Deep Networks: In contrast to the constrained scenario, deeper networks ($D = 5, 6, 7$) exhibit higher error rates compared to their shallower counterparts. This pattern is indicative of overfitting, where the increased model capacity captures noise rather than the underlying signal in the synthetic data.

c. What additional tests could you come up with to verify the results?

Here are three additional tests that could further strengthen or challenge the paper's conclusions.

1. Interpretability Verification Test: The paper claims the model is interpretable via the $\alpha_{jk}(T)$ metric in Section 4.3. I could design a synthetic dataset with known, pre-programmed context effects. For example, create data where item A always benefits from the presence of a specific decoy C, and item B is always harmed by a similar item D. After training DeepHalo on this data, I would calculate the α values. This would be a powerful validation of its claimed interpretability.
2. Scalability Test: The synthetic test was limited to a universe of J=20. A crucial real-world test would be to scale this up significantly, for example, to J=50. This would test how the model's training time, memory usage, and predictive performance scale as the universe of items grows, which is vital for applications.
3. Variable Choice Set Size Test: The experiment used a fixed choice set size of K=15. In reality, customers face choice sets of varying sizes (e.g., from 2 to 30 items). An additional test would be to train the model on a dataset with varying K. This would verify the model's robustness and ability to generalize across different context complexities.

d. Comparing to the reproducing kernel Hilbert Space Choice model of Yang(25), what are the pros and cons of Zhang(25)?

Deep Context-Dependent Choice Models (DeepHalo) and Reproducing Kernel Hilbert Space Choice Models (RKHS-CM) present two distinct modern approaches to choice modeling, each with unique advantages. DeepHalo offers superior flexibility and scalability, making it highly effective for modeling complex, high-dimensional data with intricate feature interactions. It leverages neural architectures to automatically capture hierarchical context effects, eliminating the need for explicit kernel specification. Conversely, RKHS-CM provides stronger theoretical guarantees and convex optimization properties, which are ideal for applications demanding rigorous mathematical soundness and interpretability. However, RKHS-CM requires careful kernel selection and often struggles with computational efficiency in large-scale settings. Ultimately, DeepHalo excels in practical applications where predictive accuracy and adaptability to complex, evolving feature spaces are the paramount concerns. RKHS-CM, however, remains valuable for methodologically rigorous scenarios where theoretical soundness and guaranteed convergence are critical requirements.

e. Do you think these models are suitable for the task of demand estimation of credit card offers? What are the potential problems or challenges?

While advanced Discrete Choice Models (DCMs) like DeepHalo introduce critical mechanisms to address the halo effect and cross-product dependency, the introduction of the model into a real-world credit card market reveals significantly greater challenges rooted in feature and user complexity, which may represent larger sources of approximation error. Firstly, the item features are inherently complex, like APRs, rewards structures and transfer partners, necessitating a highly sophisticated feature embedding component. Secondly, the "outside option", the user choosing none of the offered cards, constitutes a crucial and often majority segment that must be explicitly modeled, with its utility heavily influenced by individual user characteristics rather than card competition. This leads directly to the third point: consumer heterogeneity is extreme, meaning preferences for specific credit card features vary dramatically across the population, requiring the utility calculation to be conditioned not merely on item properties but on a deep embedding of user features (e.g., credit score, income, spending habits). Furthermore, the decision-making process is influenced by dynamic and unobserved context, including fluctuating economic climates, uncaptured life stage changes, and ephemeral marketing campaigns. Finally, the user's choice often involves an irrational weighting of benefits, such as prioritizing short-term sign-up bonuses over superior long-term utility (like better rewards rates), adding a temporal and psychological complexity that transcends simple competitive interaction. Consequently, addressing these multi-layered challenges of feature definition, user heterogeneity, and dynamic context is arguably more critical for predictive accuracy than solely optimizing the modeling of cross-product dependence.

f. What are other models, traditional non-neural network based, or ML models that you think are more suitable or at least worth trying? Why?

Given the analysis that consumer heterogeneity and feature complexity may outweigh the "halo effect" (cross-product interaction) in driving credit card choices, I recommend exploring the following alternative model, which is Mixed MNL model. The MMNL model serves as the most rigorous alternative to DeepHalo, particularly for addressing the "consumer heterogeneity" challenge. Unlike the standard MNL or basic DeepHalo which might assume fixed feature sensitivities, MMNL allows coefficients (e.g., sensitivity to Annual Fee or APR) to follow a distribution across the population. This directly addresses the observation that a frequent traveler values "transfer partners" differently than a student values "no annual fee. Also, MMNL relaxes the IIA assumption without requiring the complex "set-dependent" architecture of DeepHalo. If the primary goal is to capture that users substitute between similar cards (e.g., two high-fee travel cards) rather than distinct ones, MMNL captures this correlation naturally through error components.

4 Testing and Validation

4.1 Testing Plan and Objectives

The project's validation strategy was rigorously structured to ensure the architectural fidelity of the DeepHalo model replication code, targeting the implementation goals defined by Zhang et al. (2025). The primary objectives of the testing plan are:

- **Correctness:** Verification of core functions, including parameter constraint solving and data pipeline integrity.
- **Robustness:** Confirmation that the code handles input extremes (e.g., $D = 1$ or insufficient budget) and I/O operations reliably.
- **Validity:** Strict confirmation that the implemented DeepHalo architecture contains the critical layers defined in the original paper.

4.2 Testing Scope and Methodology

Unit tests are implemented using the standard `unittest` framework to isolate and verify critical components before integrated training.

1. **Mathematical Constraints (Cell 2):** Validation of the `calculate_layer_width` function, which correctly enforces the parameter budget by solving the required quadratic equation.
2. **Data Pipeline Integrity (Cell 3):** Testing of the `generate_synthetic_choice_data` function to ensure correct combinatorial completeness, output shapes, and caching functionality.
3. **Model Architecture Validity (Cell 4):** Explicit testing of the `build_featureless_deep_halo_model` to confirm the mandatory presence of the quadratic activation layer and the availability masking layer.

4.3 Testing Results and Conclusion

All validation steps were executed on the Google Colab (T4 GPU) environment. Following the successful completion of the unit test suite and integration tests, the implementation was confirmed to be sound.

Table 2 Unit Testing Results Summary

Test Case Domain	Outcome	Runtime (s)
Mathematical Constraints	PASS (OK)	
Data Pipeline Integrity	PASS (OK)	7.657
Model Architecture Validity	PASS (OK)	

All three implemented unit tests passed with zero failures, proving the structural integrity of the code base. Furthermore, integration tests confirmed the stability of the full training flow. The implementation is deemed correct and robust and is ready for full-scale replication (500 epochs).

References

Aouad, Ali, Antoine Désir. 2025. Representing random utility choice models with neural networks. *Management Science*, .

Li, Guokai, Pin Gao, Stefanus Jasin, Zizhuo Wang. 2025a. From small to large: A graph convolutional network approach for solving assortment optimization problems. *arXiv preprint arXiv:2507.10834*, .

Li, Tao, Chenhao Wang, Yao Wang, Shaojie Tang, Ningyuan Chen. 2025b. Deep reinforcement learning for online assortment customization: A data-driven approach. *Production and Operations Management*, 10591478251351737.

Liu, Jing, Gang Wang, Huimin Zhao, Mingfeng Lu, Lihua Huang, Gang Chen. 2025. Beyond complements and substitutes: A graph neural network approach for collaborative retail sales forecasting. *Information Systems Research*, .

Rosenfeld, Nir, Kojin Oshiba, Yaron Singer. 2020. Predicting choice with set-dependent aggregation. *International Conference on Machine Learning*. PMLR, 8220-8229.

Wang, Hanzhao, Xiaocheng Li, Kalyan Talluri. 2023. Transformer choice net: A transformer neural network for choice prediction. *arXiv preprint arXiv:2310.08716*, .

Zhang, Shuhan, Zhi Wang, Rui Gao, Shuang Li. 2025. Deep context-dependent choice model. *2nd Workshop on Models of Human Feedback for AI Alignment*.