BUSINESS POINT OF INTEREST RECOMMENDATION WITH REINFORCEMENT LEARNING

by
ATRA ZEYNEP BAHÇECI

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfilment of
the requirements for the degree of Master of Science

Sabancı University
December 2024

ABSTRACT

BUSINESS POINT OF INTEREST RECOMMENDATION WITH REINFORCEMENT LEARNING

ATRA ZEYNEP BAHÇECI

DATA SCIENCE M.S. THESIS, DECEMBER 2025

Thesis Supervisor: Prof. Dr. Selim Saffet Balcısoy

Keywords: business location selection, reinforcement learning, deep q-learning, location intelligence

The importance of location for business success cannot be overstated. Existing approaches to the business location selection problem often involve creating extensively tuned models specific to the geographical and economic climate being analyzed, and thus suffer from limited generalization across diverse scenarios. This thesis proposes a novel Deep Q-Learning framework for business location recommendation that can be trained in one geographic area and applied to another without requiring further training or tuning. Comprehensive experiments on real-world data demonstrate the superior generalizability of the proposed recommendation framework, outperforming the well-established Huff gravity model by 15.33% in profits, with an average profit realization of 78.02% compared to the best-case scenario. Empirical results indicate that variation in training data must be as high as the variation in the test data for the framework to be successfully applied to other locations despite discrepancies between the characteristics of the cities. The proposed approach offers a highly generalizable and easily applicable solution to the business location selection problem, providing a strong alternative to gravity-based models.

ÖZET

PEKIŞTIRMELI ÖĞRENME ILE İŞ YERI KONUM ÖNERILERI

ATRA ZEYNEP BAHÇECI

VERİ BİLİMİ YÜKSEK LİSANS TEZİ, ARALIK 2025

Tez Danışmanı: Prof. Dr. Selim Saffet Balcısoy

Anahtar Kelimeler: iş yeri konum problemi, pekiştirmeli öğrenme, derin q-öğrenme, lokasyon zekası

Bir iş yerinin başarısında konumun önemi yadsınamaz. Mevcut iş yeri konum seçimi yaklaşımları genellikle analiz edilen coğrafi ve ekonomik iklime özel oluşturulmuş ve ince ayarlanmış modellere dayanmaktadır, ve bu sebepten çeşitli senaryolar arasında sınırlı genellemeye sahiptir. Bu tez, bir coğrafi alanda eğitilebilen ve başka bir alana ek eğitim veya ince ayar gerektirmeden uygulanabilen yenilikçi bir Derin Q-Öğrenme modeli önermektedir. Gerçek dünya verileri üzerinde yapılan kapsamlı deneyler, önerilen modelin üstün genellenebilirliğini ortaya koymaktadır. Önerilen Derin Q-Öğrenme modeli seçtiği iş yeri konumları ile Huff yerçekimi modelinden %15,33 daha fazla kar elde etmekte; en iyi durum senaryosuna kıyasla ise ortalama %78,02 kar elde etmektedir. Ampirik sonuçlar, bir coğrafi alanda eğitilen modelin başka bir coğrafi alanda başarılı bir şekilde uygulanabilmesi için eğitim verilerindeki değişintinin test verilerindeki değişinti kadar yüksek olması gerektiğini göstermektedir. Modelin eğitildiği ve uygulandığı şehirlerin çeşitli özellikleri arasındaki farklılıklara rağmen, eğitildiği şehir verisindeki değişinti uygulandığı şehir verisindeki değişintiden yüksek olduğu sürece Derin Q-Öğrenme modeli uygun iş yeri konum önerileri çıkarabilmektedir. Önerilen yaklaşım, iş yeri konum seçimi sorununa son derece genellenebilir ve kolayca uygulanabilir bir çözüm sunmakta, yerçekimi tabanlı modellere güçlü bir alternatif sağlamaktadır.

ACKNOWLEDGEMENTS

I would like to express my profound gratitude to my supervisor, Prof. Selim Balcısoy, whose mentorship has been pivotal throughout my master's research. His exceptional guidance, characterized by remarkable scientific insights and consistent encouragement of innovative thinking, has significantly shaped the trajectory of this thesis. Prof. Selim Balcısoy's unwavering support extended beyond mere academic oversight; he has been an influential role model for me both as a researcher and as a person. His genuine interest in the personal growth of his students and his empathetic nature have left a lasting impression on me. Through his mentorship, I have learned not only the intricacies of academic research but also the importance of maintaining a balanced, compassionate, and enthusiastic approach to both work and life.

I would like to thank Dr. Hasan Alp Boz for his invaluable contributions during my master's research. His expertise and guidance have been instrumental in shaping this thesis. I would also like to express my gratitude for his mentorship and friendship, which have greatly enriched my master's experience.

Finally, I would like to thank my loved ones, Yasemin Satır, Yusuf Bahçeci, and Burak Yüksel for their constant support and compassion.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATONS

ADAM: Adaptive Moment Estimation

AHP: Analytic Hiearchy Process

DL: Deep Learning

DQL: Deep Q-Learning

DQN: Deep Q-Network

GDP: Gross Domestic Product

IM: Influence Maximization

LAIM: Location-Aware Influence Maximization

MCDM: Multiple Criteria Decision Making

MDP: Markov Decision Process

ML: Machine Learning

MSE: Mean Squared Error

NAICS: North American Industry Classification System

NP: Nondeterministic Polynomial-time

PRS: Profit Realization Score

ReLU Rectified Linear Unit

RL: Reinforcement Learning

TOPSIS: Technique for Order of Preference by Similarity to Ideal Solution

1. INTRODUCTION

Retail trade plays a significant role in national economies. In the United States, it accounts for over 20% of the Gross Domestic Product (GDP) and creates 15% of jobs [49]. In India, it contributes 8% to employment and 10% to GDP [20], while in the United Kingdom, it generates 9% of jobs and accounts for 5% of GDP [19, 45]. Surely, the success of retail businesses depends on a multitude of factors that are to be carefully considered by the business owner. These factors include proximity to the target customer demographics, competition, labor availability, operational costs, brand awareness and image, and regional regulations. One crucial decision that encompasses all these factors is the selection of the business location. The selection of a business location is inevitably a critical determinant of its success [31, 37]. This importance has given rise to a problem studied by economists, mathematicians, and engineers: the business site selection problem.

The business site selection problem is an algorithmic approach to selecting a location among alternatives that will maximize the success of the business. This problem is inherently complex due to two main reasons. First, the interplay between elements of a business is ever-changing and at times ambiguous. These elements, such as socioeconomic demographics, customer mobility patterns, spending habits, and geographical constraints, all interact with and affect one another. More importantly, the pattern of this interaction is neither static nor deterministic, making it difficult to analyze or foresee. Secondly, the location selection is a long-term decision [18]. This not only makes it difficult to study the outcome of the decision but also adds to the complexity of the problem by increasing the possibility of shifts in interaction patterns over time.

In this thesis, a two-step deep Q-learning (DQL) framework for the business site selection problem is proposed. In the first step, a deep-Q network (DQN) that generates a Q-value for each candidate location is trained. Daily mobility, spending, and annual census data of various locations within a city are fed into the network, which in turn returns the location with the highest Q-value as the recommended site, rewarding the agent with the consequential profits. In the second step, the trained

DQN produces Q-values for each candidate location through a forward pass of the data. After processing all the data, the Q-values for each location are accumulated and a final list of ranked recommendations based on these Q-values is generated.

Existing approaches to the business site selection problem, ranging from gravity-based models to metaheuristics, focus on understanding the relationship between the elements of a business and its success. However, since these factors exert varying degrees of influence on business success depending on the specific location, business type, and economic climate [38], it remains challenging to develop an approach that can effectively address the shifting dynamics over time, across cities, and sectors. Moreover, this ever-changing nature requires the site selection problem to be studied separately for each geographical location [38, 55], making it difficult to create an approach that can be universally applied across different geographical regions. In this thesis, it is argued that a Reinforcement Learning (RL) based approach can overcome the complexity of the problem and provide a generalizable approach for several reasons.

First, the most important distinction of RL in this context is that its direct aim is not to model the relationship between the input and output with maximum accuracy, but that it aims to maximize a pre-defined goal via its actions over the inputs. Unlike well-established models, such as the Huff gravity model, RL is tailored to fulfill the objective of business success with a focus on the respective decision-making process, not primarily on prediction. In this regard, it does not tackle modeling the relationship between the input and output, the relationship between information on candidate locations and business success. As it does not tackle modeling this relationship, which is dependent on a plethora of quantifiable and unquantifiable parameters, it is hypothesized that the proposed RL-based approach will have superior performance. Moreover, by eliminating the modeling of said relationships the proposed model will be more generalizable compared to existing methods. The relationships among the elements that affect the success of a business vary in temporal dimensions, geographical levels, locations, business sectors, etc., making a universally applicable method based on prediction elusive. The DQL model learns the actions and their consequences, which seem to not differ as much as the relationship between fore mentioned complex parameters, allowing a singular RL framework to navigate these complicated settings efficiently. Secondly, the DQL does not require any predetermined notion or guide, such as the notion that certain parameters are relevant or not (like feature subset selection), or that certain parameters follow a specific function (like the store visits in the Huff gravity model). This eliminates the need for expert opinions, a detailed preliminary analysis, or assumptions. Again, the omission of such assumptions or set structures that vary from one setting to another contributes to the generalizability of the proposed approach.

The contributions of this thesis are the following:

1.1 At the time of this study, this thesis is the first to utilize an RL based approach for the business site selection problem.

1.2 At the time of this study, this thesis is the first to perform training and testing in cities from different states for the business site selection problem.

1.3 A DQL framework for business location recommendation that can be trained with data from one city and can be used for recommendation generation in another is proposed.

1.4 Experiments demonstrate that (a) the proposed framework outperforms the Huff gravity model benchmark, (b) the framework requires the variance of the training dataset to be at least as high as the test dataset for profitable recommendations, and (c) the framework requires no hyper-parameter tuning or feature subset selection.

In Chapter 2, a literature review is conducted on approaches to the business site selection problem with a focus on the limitations of existing research. In Chapter 3, the fundamentals of RL and Q-learning are introduced. In Chapter 4, the data and the proposed recommendation framework are explained in detail. Chapter 5 introduces the experimental setup and benchmark model, presents experimental results, and discusses the findings. Lastly, in Chapter 6, the thesis is summarized, the concluding remarks are presented, and the potential directions for future research are explored.

2. RELATED WORKS

The business site selection problem has been extensively studied since the early 1900s [26]. Over the years, the problem has been studied through various research domains and formulations (besides business site selection), including market share prediction [16,38,55], business performance prediction [64,66], and footfall prediction [17,40]. This section will discuss the research on the business site selection problem, aiming to identify the scientific progress, the current state of research, and explore its limitations.

2.1 Huff Gravity Model

In 1963, David L. Huff created the Huff gravity model to determine the *"retail trade area"*, the geographical area that contains the prospective consumers for a shopping center [29]. He argued that the probability of a consumer visiting a shopping center is related to the *"utility"* of the center, which is made up of the square footage of the store and the travel times of customers to get to the store; and quantified the probability of a customer from region i visiting shopping center j as the following:

$$P(C_{ij}) = \frac{S_j}{T_{ij}^{\lambda}} \bigg/ \sum_{j=1}^{J} \frac{S_j}{T_{ij}^{\lambda}} \tag{2.1}$$

where S denotes the store sizes, T denotes the travel times, and λ denotes the empirical parameter that estimates the weight of the travel times. Over time, the Huff gravity model slightly evolved with the square footage replaced by a more general term store attractiveness (A) with an added parameter to reflect its weight. With the up-to-date notation, the Huff gravity model calculates the probability of consumer from area i visiting store j as follows:

$$P_{ij} = \frac{A_j^\alpha}{D_{ij}^\beta} \bigg/ \sum_{k=1}^{n} \frac{A_k^\alpha}{D_{ik}^\beta} \tag{2.2}$$

where D denotes the distance between the customers' homes and the store, α denotes the weight of the attractiveness, and β denotes the weight of the distances.

Through the decades, the Huff gravity model has remained one of the most well-established and widely used models for business site selection [16, 34, 50, 55]. Its simplicity, computational efficiency, and interpretability compared to other methods have made it a popular choice [12]. Nevertheless, recent research suggests that the interpretability and efficiency of the Huff gravity model may come at a cost [15, 38, 40, 54, 55].

Dock et. al demonstrated the importance of variable selection and how the attractiveness (A) is constructed affects the performance of the Huff gravity model [15]. While they managed to improve predictive accuracy by creating a carefully constructed attractiveness measure, their approach relied heavily on domain-specific knowledge and was validated for only a single county and a specific business category.

Suarez et. al focused on the parameter calibration of the Huff gravity model [54]. They hypothesized that certain parameters exhibit spatial nonstationarity and therefore require careful calibration using localized data. Their experiments validated this hypothesis, demonstrating the need for location-specific adjustments; while also suggesting that the Huff gravity model could not be effectively generalized across different geographical regions.

Researchers have analyzed the robustness of the Huff gravity model. Lu et. al examined how the model's performance varies with different sampling locations [40]. They discovered special locations within the large dataset that significantly enhance predictive accuracy. However, they found no specific social characteristics that could help identify these special locations across different applications or datasets; again hinting that the Huff gravity model requires additional analysis or domain-specific knowledge for the best results.

Another issue that researchers have focused on is the efficacy of the traditional Huff gravity model in capturing critical temporal relationships in the business setting. Liang et al. argued that the conventional Huff gravity Model is static and introduced the Time Aware Dynamic Huff (T-Huff) model [38]. They claimed that the original Huff gravity model treats mobility data as any time series where the pattern in the data is solely sequential (see Equation 2.2). In contrast, the T-Huff model was developed to also account for daily patterns and re-formulated to predict visit probabilities over a time window. Their experiments showed that the T-Huff significantly improves the accuracy of market share predictions compared to the traditional Huff gravity model, but was limited to only the top three chain-store brands.

In 2021, Suhara et al. were the first to validate the Huff gravity model for market share prediction using real-world transactional data [55]. They divided one city into 17 regions and developed a separate model for each region. Although they demonstrated a significant predictive performance across all regions, they showed that the model must be trained specifically for each region, even within the same city, hinting that the model cannot transfer its learnings from one region to another and is locally bound.

In summary, despite its performance and simplicity, the adaptability of the Huff gravity model across different geographic regions, time frames, and contexts remains limited [15,50,54,55]. Moreover, the Huff gravity model requires considerable domain knowledge [15, 40] and extensive pre-processing [40, 54]. It can be concluded that the Huff gravity model lacks generalizability for the business site selection problem.

2.2 Multiple Criteria Decision Making

Multiple criteria decision making is a methodology that facilitates the selection of an optimal decision among various alternatives based on specific criteria [3]. In the context of business site selection, researchers have applied methods such as Analytic Hiearchy Process (AHP) [2, 41, 53, 63], the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) [2, 53], and the Delphi method [41] to select business locations across various sectors.

AHP is a structured method that involves a hierarchy of a main objective, sub-objectives, and alternatives. Through matrix-wise operations, AHP enables the selection of the best alternative by evaluating each against the defined objectives [3]. Yap et al. employed AHP to determine the optimal location for utility payment branches [63]. Their criteria, determined by experts and validated by sales data, were based on the accessibility, distances, and convenciences of the alternative locations. Despite the performance of their method, their study was limited to selecting among only four sites.

Several studies have integrated AHP with TOPSIS. Similar to AHP, TOPSIS uses a matrix of criteria and alternatives, and uses matrix-wise opeartions to determine the optimal decision [3]. Shaik et al. developed a hybrid AHP-TOPSIS model to select a gas station location among 20 alternatives [53]. The criteria for their study; traffic patterns, vehicle ownership, site popularity, and competition; were based on expert judgments. Al et. al followed a similar methodology of a AHP-TOPSIS model to determine the hospital location among four alternatives, with the decision criteria defined by a committee of decision-makers [2].

There are also studies that utilize other multiple criteria decision making methods. In a 2013 study, researches have utilized step-wise weight assessment ratio analysis and weighted aggregated sum product assessment to determine the location of a shopping mall among five alternatives [67]. They first determined the decision criteria based on expert opinions, employed step-wise weight assessment ratio analysis to calculate the criteria weights, and applied weighted aggregated sum product assessment for ranking the alternatives based on these weighted criteria.

Despite the experimental success of the aforementioned studies, multiple criteria decision making for business site selection faces several limitations. Firstly, it often restricts the number of candidate locations that can be analyzed [2, 53, 63, 67]. Secondly, the candidate locations are forced to be close in proximity, as the decision criteria vary across location alternatives in different regions [2, 53, 63, 67]. Thirdly, multiple criteria decision making methodologies heavily rely on expert knowledge, which can be both challenging and costly to acquire [2, 41, 53, 63, 67]. Lastly, these methods lack generalizability across sectors and geographic regions. The expert-defined criteria tailored to a specific sector and region change across different contexts [32], making it difficult to re-apply these methodologies in new settings.

2.3 Influence Maximization Methods

In the field of Influence Maximization (IM), real-world scenarios are modeled using social networks represented as graphs. Each edge in the graph is associated with a probability of activation during the propagation process. The goal of IM is to identify a seed set of pre-defined size that maximizes the number of activated nodes, thereby maximizing the overall influence spread. In the context of business site selection, candidate locations and prospective customers are modeled in the graph.

The influence of each candidate location is measured by the number of customers it attracts, and the IM algorithm aims to identify a seed set from the social network that maximizes this influence. In essence, IM focuses on identifying a smaller subset of nodes that maximize the expected influence of customers [14].

Given that IM is a Nondeterministic Polynomial-time (NP) hard problem, it is addressed using various metaheuristic methods. As social networks grow in size, driven by the increasing availability of (mostly check-in) data, researchers focused heavily on developing efficient algorithms. Prominent metaheuristic methods for business site selection include pruning algorithms [28], which iteratively narrow down the potential set of optimal seed sets. These algorithms work by eliminating solutions in the search space that are branching out from non-optimal seed sets, reducing the computational complexity. There are also various other metaheuristic approaches such as the nearest location circle, and Voronoi diagrams [27].

While these methods are proven to be efficient, they often overlook the spatial relationships within social networks. These spatial relationships can provide critical insights into influence spread, and thus the success of a business. Ignoring such factors may lead to suboptimal solutions in real-world applications, particularly in the business setting where geographical proximity plays a key role in determining influence.

2.3.1 Location-aware influence maximization

To address the limitation of overlooking spatial relationships in IM, Location-Aware Influence Maximization (LAIM) was created by Li et al. [35]. With LAIM, the probability of activation for every edge is dependent on its region. Unlike IM, the influence spread is not modeled between nodes (node-node), but it is modeled between nodes and regions (node-region).

Similar to the original IM formulation, LAIM is NP-hard, necessitating the use of metaheuristic approaches. These approaches, ranging from pruning algorithms to randomized influence strategies, have been proven to be both efficient and effective [35, 39, 65].

Despite the predictive performance and scalability of the IM and LAIM methods, they only focus on addressing a singular problem: determining the optimal location for a business in a specific, isolated setting [27, 28, 35, 39, 65]. Since these methods model real-world scenarios through a graph-based approach, the learning and thus the decision-making are inherently confined to that specific graph. Consequently, with IM or LAIM, insights gained from one particular setting, specific to a single business category, geographical location, and time period, cannot be generalized or transferred to other contexts.

2.4 Machine Learning

In the context of business site selection, machine learning methods have been utilized as predictive methods. Most research identifies an indicator for business success, predicts it for various candidate locations, and selects the location with the highest predictions as the recommended site. Over the years, both traditional machine learning methods [7, 32, 36, 57, 59, 61, 64] and deep learning models [23, 62, 66] have been studied.

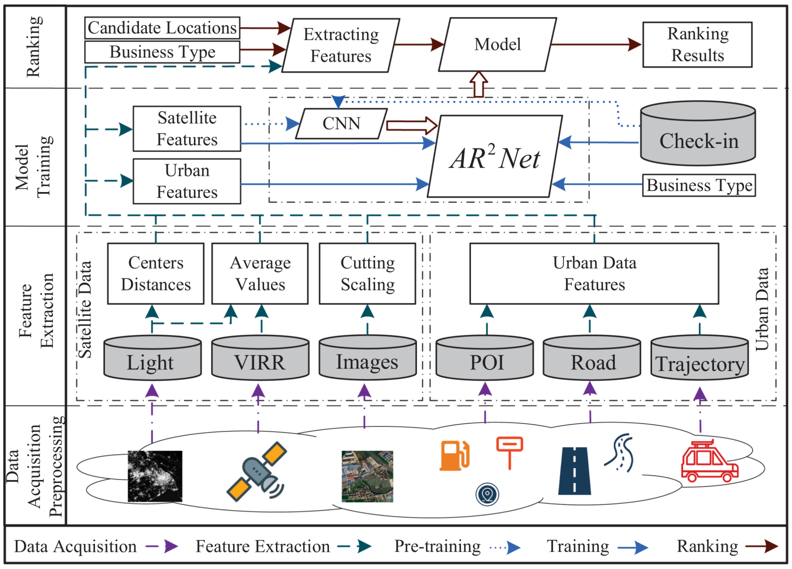
Among the traditional machine learning approaches, regression models are among the most studied methods [7, 32, 59]. In a 2016 study, Wang et al. employed ridge regression, support vector regression, and boosted regression trees to predict store visits and recommend site locations based on the predicted number of visits [59]. Despite their outstanding experimental results, it can be argued that visit counts alone do not equate to business success in all sectors. Furthermore, the relationship between predictor variables and visit counts can vastly change across settings; making it highly unlikely that such a model can be generalized to other contexts. Certain studies have integrated regression-based methods into more complex decision-making frameworks [7, 24, 32]. Karamshuk et al. developed a model that uses separate regression models to predict eight different features that signal business success and then ranks locations based on these predictions [32]. Similarly, Bilen et al. created a methodology that predicts 17 features individually and clusters the predictions to identify potential regions for business sites [7]. While such approaches demonstrate how embedding regression methods within larger frameworks increases model performance, they introduce the challenge of deciding which features to predict and, by extension, what criteria to base site selection on. Given that indicators of business success vary over time, across locations, and between sectors, these methods are also difficult to generalize and re-apply in other contexts.

In addition to regression-based approaches, various traditional machine learning methods have also been applied to the business site selection problem such as learning to rank algorithms and boosting models [36, 57, 61, 64].

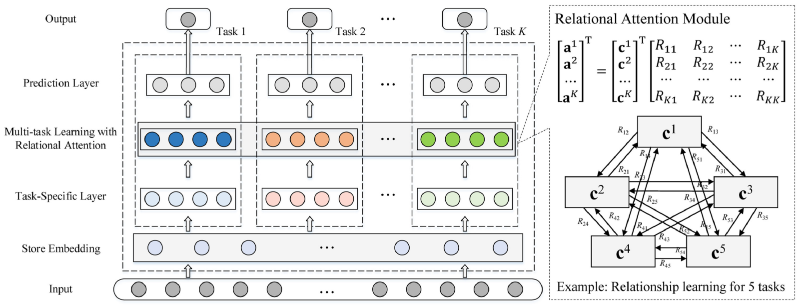
2.4.1 Deep learning models

The deep learning methods applied to business site selection significantly vary in their approaches and focus. For instance, in 2020, Xu et al. introduced AR^{2}Net, a framework that analyzes satellite images and road trajectories, learning the relationship between these analyses and business success to predict the popularity of candidate locations [62]. They argued that AR^{2}Net is capable of providing store recommendations across all sectors, making it more generalizable. Similarly, Zhao et al. developed a deep learning framework incorporating deep multi-task learning with relational attention [66]. Their model consists of a store embedding layer, a task-specific layer, a relation learning module, and a prediction layer. The task relationships are not predefined, instead, they are learned from data, making the approach more generalizable. Han et al. developed a two-step framework that applies spatial co-location pattern mining to identify candidate locations for restaurants, followed by a novel graph convolutional network, locationGCN, to select from these candidates based on specific restaurant types [23]. By designing the model to be adaptable to different sorts of restaurants, they created a methodology that can be generalized within the restaurant sector.

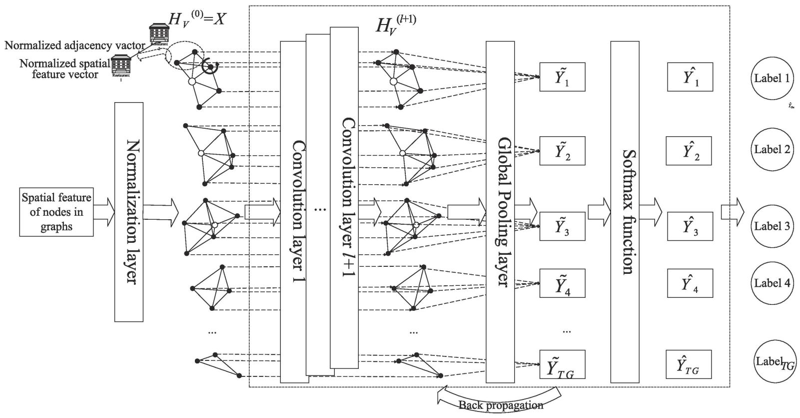
The previously mentioned deep learning approaches certainly enhance the generalizability of methods for business site selection. However, their experiments demonstrate this generalization only within a single city or a specific sector. Furthermore, as demonstrated in Figure 2.1, these methods are complex, time-consuming, and computationally expensive. In summary, while deep learning methods offer greater yet limited generalization compared to other approaches, they come at the cost of increased complexity and resource demands.



(a) Architecture of the proposed methodology by [62]



(b) Architecture of the proposed methodology by [66]



(c) Architecture of locationGCN, step 2 of 2 from the proposed methodology by [23]

Figure 2.1 Architectures of various deep learning approaches to the business site selection problem

3. REINFORCEMENT LEARNING

In this thesis, the business location selection problem is studied via a RL framework. In this section, the definitions, notations, and mathematical equations of RL that will be referenced in the subsequent chapters are detailed.

Sutton and Barto define RL as: “a machine learning technique the learns how to act in order to maximize a quantitative reward", and list its main features as the following [56]:

2.1 “It has a closed loop structure”. The model calculates possible rewards and actions by observing the environment. When action is taken, the environment changes. The environment is re-observed to determine a new action and a cycle occurs. In summary, the output of each period creates the input of the next period.

2.2 “Actions are not determined by direct instructions”. Actions are learned entirely by trials and experience. There are no instructions specifying which actions to take in which situations.

2.3 “The consequences of actions continue to affect the model over a long period of time”. Given the cyclical nature of RL, every action indirectly or directly leads to the next action, creating both a long-term and a short-term effect for each action. The aftermath of the actions can and will be observed in the future periods.

In line with these three features, RL’s place among other machine learning techniques is visualized in Figure 3.1 (figure adapted from [47]).

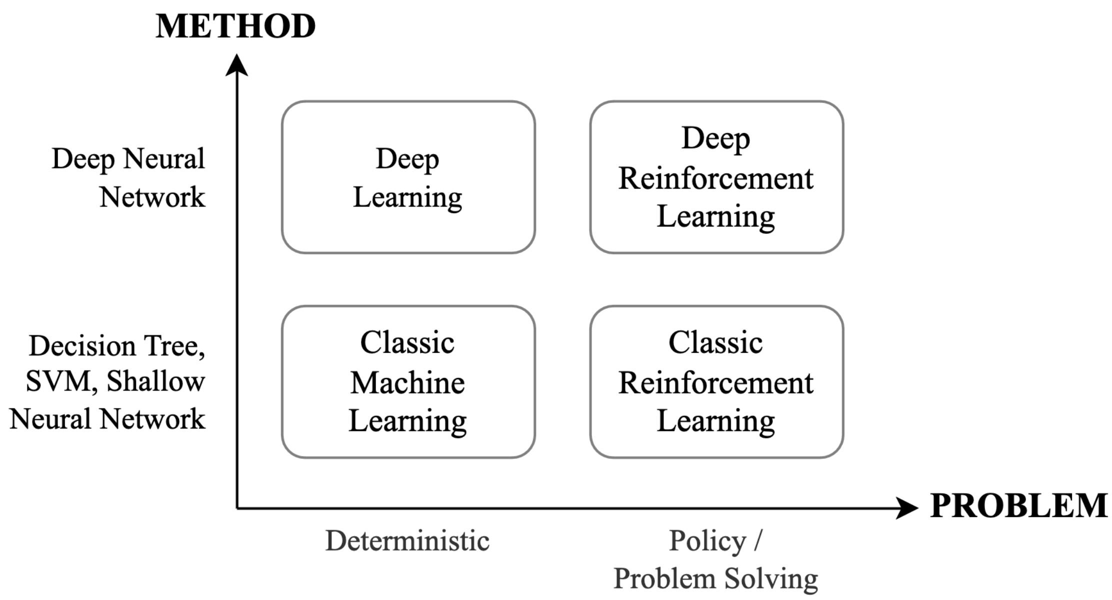


Figure 3.1 Machine learning methods according to problem type

In contrast to classic machine learning and deep learning, RL is not focused on learning a set of equations that apply to every system. Instead, RL is a flexible method that adapts to the data and the environment to create the optimal strategy.

3.1 Terms and Notation

The main components of RL are agent, environment, action, state, reward, and policy. Sutton and Barto define these terms as follows [56]:

Definition 3.1: The element that learns to achieve and take actions towards the reward is called an *agent*.

Definition 3.2: All the items that the agent interacts with and that are outside the agent are called the *environment*. Although the environment-agent boundary may seem flexible in some settings, it is generally accepted that all the elements that the agent cannot arbitrarily change are taken as the environment.

Definition 3.3: All possible decisions the agent can make are called *actions*.

Definition 3.4: All the elements that may change or affect the action in any way are called *state*.

Definition 3.5: As the agent takes action, it receives scalar signals from the environment. These signals are called *rewards*. Each reward is a real number.

Definition 3.6: The *policy* is the rules by which the agent selects its actions based on the state. These rules are stochastic due to the nature of RL.

Table 3.1 summarizes the key RL notations used in this thesis.

Table 3.1 Key RL notations

Notations	**Meanings**
a	Action
A	Set of all Actions
α	Learning Rate
ϵ	Exploration Rate
γ	Discount Parameter
M	Number of Eepochs
$Q(s,a)$	Action-value Function
Q^{Pred}	Predicted Q-value
Q^{Target}	Target Q-value
r	Reward
s	State
S	Set of all State Values
Θ	Network Parameters
Θ'	Target Network Parameters
π	Policy
V	Value Function
w	Network Weights

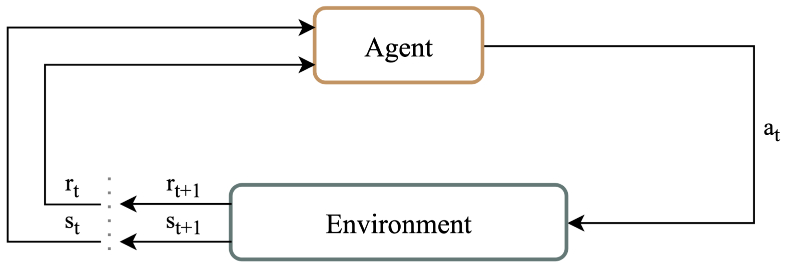


Figure 3.2 Simplified RL structure

The agent observes the environment via the state s_t, takes action a_t, and receives the consequent reward r_t. Then, the system transitions to the next state s_{t+1}, thereby creating the closed loop structure.

The following mathematical representations and explanations of RL are taken from Sutton and Barto [56].

The RL process can be expressed through the Markov Decision Process (MDP). MDP is a tuple denoted by (S, A, P_a, r_a), where $P_a(s_t, s_{t+1})$ is the probability that action a causes the transition from s_t to s_{t+1}, and $r_a(s_t, s_{t+1})$ is the reward at time t

achieved by action a and the transition from s_t to s_{t+1}. The objective of the model is to find the policy that provides the maximum reward. The objective function is as follows:

$$\arg\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_{\pi}(s_t, s_{t+1})\right] \tag{3.1}$$

where r_{π} is denotes the reward the policy earned when. The discount factor γ prevents the policy from postponing its actions over the analysis period.

Definition 3.7: During the training process, the algorithm learns through multiple policies and compares their rewards. The policy that brings the highest reward is called the *optimal policy*.

3.2 Model Based and Model Free Learning

RL is divided into 2 categories: model-based and model-free.

The aim of model-based learning is to ensure that every action taken, regardless of the situation, is optimal and will grant the maximum reward in the future [30]. The agent creates a model of the environment using the state and creates a policy based on the model of the environment. In other words, the policy is shaped solely by the data [42].

On the other hand, with model-free learning, learning is done over the maximization of the value function; and the objective is to find the optimal value function

$$V^{\star}(s) = \max_{\pi} V_{\pi}(s) \quad \text{s.t.} \quad V_{\pi}(s) = \mathbb{E}\left[r_{t+1} + \gamma r_{t+2} + \gamma r_{t+3} + \ldots\right] \tag{3.2}$$

where $r(s)$ is the short-term reward, $V_{\pi}(s)$ is the long-term reward of policy π.

In Equation 3.2, discount factor γ determines the model's tendency to explore. A discount factor closer to 0 pushes the model to prefer instant rewards over long-term rewards, and consequently inhibits the model from exploring. A discount factor closer to 1 encourages the model prefer long-term rewards and explore.

Mathematical representations and explanations of Q-learning are adapted from Carta et al. [13].

Q-learning is distinguished by the rest of the model-free learning methods with its action-value function, $Q(s,a)$, that replaces the traditional value function, $V(s)$.

$$Q(s,a) = \sum_{s'} p_{ss'}(a) \cdot [\nabla(s,s',a) + \gamma \cdot \max_{a'} Q(s',a')] \tag{3.3}$$

where s' is the state observed after performing an action, $p_{ss'}(a)$ is the probability of going from state s to state s' by performing action a, $r(s,s',a)$ is the immediate award of action a that lead the transition from s to s'.

The Q-values are updated by the Bellman Equation given below:

$$Q(s,a)_{\text{new}} \leftarrow Q(s,a) + \alpha \left[r(s,s',a) + \gamma \max_{a'} Q(s',a') - Q(s,a) \right] \tag{3.4}$$

where $Q(s,a)$ is the old Q-value at state s, and $\max_{a'} Q(s',a')$ is the maximum Q-value of the destination state s'.

The agent selects action according to the Q-values. The optimum action, $a^\star$ is selected as the following:

$$a^\star = \operatorname*{argmax}_a Q(s,a) \tag{3.5}$$

The learning starts with all Q-values $Q(s,a)$ initialized as 0. At every step t, the agent selects action a_t via Equation 3.5, receives reward r_t, and consequently transitions to state s_{t+1}. With the transition, the Q-values are updated via the Bellman Equation 3.4.

Definition 3.8: The agent begins learning with high exploration, with a high probability to take random actions. This probability is denoted with ϵ and called the *exploration rate*.

Definition 3.9: Through the iterations, the exploration rate ϵ decays, and hence the agent acts more and more based on what it has learned. This behavior is called the *greedy strategy*.

3.3.1 Deep Q-learning

In systems where the environment is observed by a multitude of state values, it becomes inefficient and time-consuming to update the Q-values every iteration. To address this, Mnih et al. developed an approach wherein the Q-values are not updated by the Bellman Equation but approximated by a neural network called Deep Q-Network (DQN), and named this architecture Deep Q-learning (DQL) [43].

In DQL, there are two identical networks called the target network and the prediction network. Both networks take the state as the input, and output Q-values for each action. The difference between the two networks is that the prediction network's outputs are the values that actually determine the action, and the target network's outputs are used to train the prediction network. The prediction network is trained with the following loss function:

$$L(\theta) = (Q^{\text{Target}} - Q^{\text{Pred}})^2 = (r_t + \gamma \max_a Q^T(s_{t+1}, a; \theta') - Q(s_t, a; \theta))^2 \tag{3.6}$$

The predicted Q-value is calculated as the following:

$$Q^{\text{Pred}} = Q(s_t, a; \theta) \tag{3.7}$$

and is simply the output of the prediction network.

The target Q-value is calculated as the following:

$$Q^{\text{Target}} = r_t + \gamma \max_a Q^T(s_{t+1}, a; \theta') \tag{3.8}$$

The prediction network is initialized with random weights, and the target network is initialized by directly copying the prediction network. The weights of the prediction network are updated following the equation:

$$w_{t+1} = w_t - \alpha \mathbb{E}[L(\theta)] = w_t - \alpha \mathbb{E}[(Q^{\text{Target}} - Q^{\text{Pred}})^2] \tag{3.9}$$

Periodically, the target network is updated by directly copying the parameters from the prediction network. Without backpropagation on the target network, the computational overhead of having an additional neural network is insignificant. The update frequency of the target network can be pre-determined or adjusted dynamically during training.

In some DQL applications, the learning is supported with experience replay. That is, the (s, a, r) tuples encountered are stored in a memory, and the prediction network is optimized using the batch of random samples obtained from the memory. The purpose of this approach is to eliminate potential correlations between consecutive states by basing updates not on sequential states but on randomly selected ones [43]. In this thesis, a memory replay mechanism is not utilized for the reasons discussed in Subsection 4.2.3.

4. METHODOLOGY

This section provides a comprehensive overview of the data and recommendation framework. First, the datasets and their enhancements through feature engineering are described. Following this, the parameter configuration and model architecture are thoroughly explained. The section concludes with a discussion of the measures taken to ensure training stability and the challenges addressed in developing the framework.

The following Table 4.1 summarizes the notations used for the recommendation framework.

Table 4.1 Notations for the recommendation framework

Notations	**Meanings**
c	Business Category Index
C	Competition
D	(Average) Distance Between Business and Visitors' Residents
E	Expenditure (in USD)
GT	Ground Truth
H	Business Diversity
i	Neighborhood Index
I	Set of all Neighborhoods
I^*	Set of Recommended Neighborhoods
MA	Median Age
DW	Median Dwell at Business (in minutes)
MHV	Median Home Value
MI	Median Income
MR	Median Rent
NF	Number of Families
P	Population
PRS	Profit Realization Score
PTA	Public Transportation Accessibility
V	Visits (count)

4.1 Data

4.1.1 Datasets

Mobility Data: Mobility data is acquired from the SafeGraph Data Consortium [52]. The data encompasses information regarding the weekly frequency of visits to businesses (V), the median time they spent at the businesses (DW), the residential origins of these visitors, the distances between their residences and businesses (D), and the North American Industry Classification System (NAICS) codes relevant to these workplaces.

Spending Data: Spending data is again sourced from the SafeGraph Data Consortium [51]. This dataset includes transactions conducted at businesses, both weekly and capturing a blend of physical and online spending.

The mobility and spending data is taken from January 2019 to March 2020, effectively preventing the model to be distorted by the effects of the Covid-19 pandemic. Furthermore, the data is sub-sampled to only include the weekdays (Monday to Friday).

American Community Surveys: To investigate the socio-demographic characteristics differentiating the neighborhoods, data is sourced from the American Population Surveys conducted by the United States Census Bureau [9]. CensusData library is used to collect 1-year estimates for 2019, for the variables population (P), median age (MA) and income (MI), median rent (MR), median home value (MHV), and the number of families (NF) via tags `'B01001_001E': 'Population'`, `'B01002_001E': 'Median Age'`, `'B19013_001E': 'Median Income'`, `'B25064_001E': 'Median Rent'`, `'B25077_001E': 'Median Home Value'`, and `'B11016_001E': 'Number of Families'`, respectively. This library enables the retrieval of American Community Surveys data through the U.S. Census Bureau's API.

A methodological change was made to resolve the observed mismatch between mobility and spending, where mobility data was collected more frequently, leading to a false representation of customer behavior. For a considerable amount of neighborhoods, it appeared that businesses that had lots of visitors had nearly no revenue, indicating a sampling bias for the spending data. The data was harmonized by calculating the expenditure (E) metric via Equation 4.1, which distributed the total spend proportionately over the number of visitors.

$$E = \frac{\text{Total Spend}}{\text{Number of Spenders}} \cdot \text{Number of Visitors} \tag{4.1}$$

4.1.2 Feature engineering

In addition to the features readily available within the datasets, three variables are constructed to enrich the agent's capability to distinguish among neighborhoods. These variables are public transportation accessibility (PTA), business diversity (H), and competition (C).

Public transportation accessibility: Public transportation accessibility, PTA_t^i for neighborhood i at time t, is added as an indicator of a neighborhood's urban quality, a proxy for a business's potential integration within mobility networks. This variable is intended to reflect the ease with which individuals can reach or depart from a given location, thereby influencing the location's attractiveness for both businesses and their prospective customers. The total number of public transportation access points in each neighborhood is used as a measure of its accessibility. These access points are retrieved from the OSMnx library using the following tags: `'public_transport':'station'`, `'highway':'bus_stop'`, `'railway':'tram_stop'`, `'railway':'subway_entrance'`, `'amenity':'ferry_terminal'`, `'public_transport':'stop_position'`.

Business Diversity: The concept of business diversity is added to hint at a neighborhood's heterogeneity in terms of various business categories and sectors. The informative aspect of this variable is that the diversity of a neighborhood's business environment potentially influences its appeal as a location, either by positioning it within existing patterns of mobility and consumer behavior or by situating it outside these established flows. Shannon's Entropy was used to calculate the business diversities in neighborhood i at time t via the following formula outlined by Bahrami et al. [5]:

$$H_t^i = -\sum_c p_c \log_2(p_c) \tag{4.2}$$

where p_c denotes the proportion of businesses belonging to the c-th category. To ensure the accuracy of the business diversity metric, calculations were performed on a monthly basis rather than daily or weekly intervals. This mitigates potential distortions resulting from intermittent data collection and reduces the possibility of false interpretations of business closures.

Competition: The competition feature, C_t^i for neighborhood i at time t, is developed to provide a quantitative assessment of the competitive landscape prevailing within each neighborhood. This measure, inspired by Tian et al. [57], was derived by calculating the ratio of the number of unique businesses within the selected business category to the total number of unique businesses within each neighborhood with the following formula:

$$C_t^i = \frac{(B_c)_t^i}{\sum_c (B_c)_t^i} \tag{4.3}$$

where $(B_c)_t^i$ denotes the number of businesses in category c in neighborhood i at time t.

4.2 Recommendation Framework

The proposed framework employs a DQL-based approach to recommend optimal neighborhoods for establishing businesses within specific categories. The framework treats the city under analysis as the environment, and the recommendation of a business location as the action. The environment (city), is observed via the data, an action (location recommendation) is taken, and the validity of the action is assessed via the dataset.

4.2.1 Parameter configuration

State. The state of the model s_t is defined as:

$$s_t = [s_t^1, s_t^2, s_t^3, ..., s_t^n] \tag{4.4}$$

where the state is actually a flattened vector of data from all neighborhoods. Data from each neighborhood, or the state vector s_t^i for neighborhood i, is expressed as:

$$s_t^i = [\mathrm{C}_t^i, \mathrm{D}_{t-1}, \mathrm{E}_{t-1}^i, \mathrm{DW}_t^i, \mathrm{H}_t^i, \mathrm{MA}_t^i, \mathrm{MHV}_t^i, \mathrm{MI}_t^i, \mathrm{NF}_t^i, \mathrm{P}_t^i, \mathrm{PTA}_t^i, \mathrm{V}_{t-1}^i] \tag{4.5}$$

Reward. The reward of the model is constructed as the following:

$$\mathrm{r}_t^i = \mathrm{E}_t^i - \mathrm{MR}_t^i \tag{4.6}$$

The reward aims to measure the profit of a business over the available data. It essentially is an approximation of the difference between the revenue (E) and the expenses (MR). Incorporating a term for the expenses allows the model to reflect the real-world setup, as it was observed that the locations with the most revenue often have the largest expenses, and thus the revenue alone would be a false indicator of business success.

4.2.2 Model training and recommendation generation

The developed framework consists of two main steps; model training and recommendation generation. The 2-step framework is summarized in Figure 4.1.

Step 1: Model Training. The training process includes a prediction network and a target network, with the goal of training the prediction network.

The architecture of the prediction and the target network is detailed in table 4.2. Rectified Linear Unit (ReLU) activations are employed for the first two layers, to capture the non-linear relationships inherent in the dataset, reflecting the complexities of the real-world environment. For the third layer, however, ReLU is replaced with a linear activation function allowing the network to output negative values. This adjustment allows the model to output negative values, which is crucial given the nature of business setting. The ability to generate both positive and negative Q-values is especially important as it enables the model to fully represent the range of outcomes in the system, where incorrect actions can have significant consequences, and correct actions can yield substantial benefits. This capacity to express a wide range of values through the Q-values gives the framework the flexibility needed to capture the dynamic characteristics of the business setting.

Table 4.2 Network architecture

Layer	# Nodes	Activation Function
Input Layer	720	Linear
Hidden Layer 1	128	ReLU
Hidden Layer 2	64	ReLU
Output Layer	60	Linear

The training process is carried out through the following steps:

Algorithm 1: Model Training

```
Initialize prediction network with random weights Θ;
Initialize target network with weights Θ' = Θ;
for epoch = 1 to M do
    for t = 1 to T do
        Observe state s_t;
        if With probability ε then
            select a random action a_t;
        else
            select a_t = max_a Q^Pred(s_t, a; Θ);
        end
        Receive reward r_t;
        Observe next state s_{t+1};
        Calculate loss L(θ) = MSE(Q^Target, Q^Pred);
        Update prediction network w_{t+1} = w_t − αE[L(θ)];
        Decay exploration ε ← max(ε · ε^decay, ε^min);
        if target network update condition is met then
            Update target network Θ' ← Θ;
        end
    end
end
```

Step 2: Recommendation Generation. The model generates recommendations of k locations to open a business in for the test period with the following steps:

Algorithm 2: Recommendation Generation

for $epoch = 1$ *to* M **do**
    for $t = 1$ *to* T **do**
        Observe state s_t;
        Calculate $Q_t^i(s_t, a; \Theta)$;
    end
    Recommend k neighborhoods $I^\star = \underset{i \in \mathcal{I}}{\operatorname{argmax}}^k \sum_t Q_t^i$;
end

For both of the steps, data exclusively from the preceding time step is fed to the model. Specifically, the action at time step t is determined based on observations from time step $t-1$, thereby avoiding look-ahead bias and creating a setup in line with the real world.

4.2.3 Considerations for training stability and model performance

Real-world applications of DQL are often highly sensitive and prone to divergence [25]. Divergence in DQL typically arises from Q-value updates [1], specifically from overestimation [44]. Overestimation occurs when small changes in the value function lead to significant variations in the policy (refer to Equation 3.4) [22]. To prevent divergence and ensure performance, the framework incorporates three key considerations.

3.1 *Parameter Scaling:* All variables in the state undergo min-max scaling, whereas the reward is scaled by percentage. Given that the DQN is initialized with random weights between 0 and 1, the initial Q-values typically are very small numbers. If any of the other variables, especially the reward, is significantly larger or smaller than these initial Q-values, the model is prone to divergence (refer to Equation 3.8). Both min-max scaling and percentage scaling produce values within the range of (0,1), aligning with the scale of the initial Q-values and thus avoiding divergence. For the reward, the proper scaling is important for the additional reason of reflecting a difference between the possible actions. If the rewards for different actions are not sufficiently differentiated, it becomes challenging for the model to learn. Through percentile scaling, this necessary differentiation is achieved, enabling the model to learn within the complex system of business success and location.

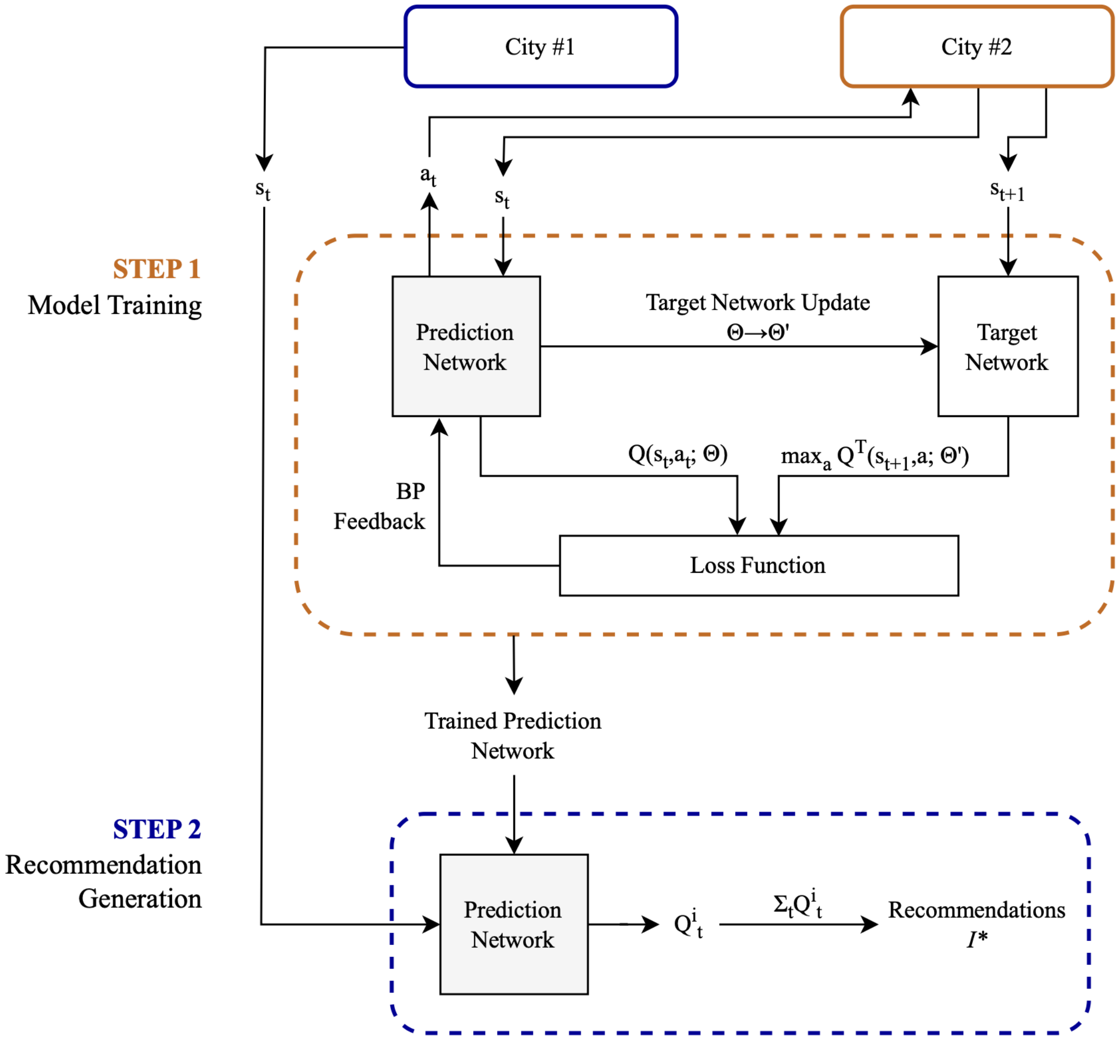


Figure 4.1 Flowchart of the recommendation framework

In Step 1, data from the first city is used to train the prediction network. In Step 2, the trained prediction network from Step 1 is used to generate the set of recommended neighborhoods, I^*, in the second city. The trained network is used as is and no further processing for the second city is done.

3.2 *Target Network:* A target network is embedded in the framework to prevent divergence [46]. Some researchers have developed various Q-learning frameworks that render the target network unnecessary for preventing divergence [1,33,48], however, such methodologies were proven efficient primarily with very large datasets. In this thesis, the dataset is much shorter than the datasets in typical RL studies. Hence, a target network is chosen over these alternative methods.

3.3 *Memory Replay:* A memory replay mechanism in DQL, where the agent randomly samples from its prior experiences during the learning process, creates off-policy learning. In off-policy learning, the agent utilizes data from earlier states of the policy. Sutton and Barto identify off-policy learning as one of the *"three deadly triads"* of DQL [56]. This is because of extrapolation errors (a phenomenon that occurs when the data sampled from the memory has a different distribution than the current state) and consequently the agent making inaccurate predictions [21], possibly leading to overestimation. In the business setting, the agent is especially susceptible to extrapolation errors. For instance, if the agent is processing data from April but randomly samples past data from the Christmas season, it will inherently disrupt the policy being learned. For this recommendation framework, employing a memory replay mechanism slightly decreases model performance. Overall, it can be concluded that memory replay should be approached with caution in settings where data exhibit seasonal patterns, such as in business contexts. In this thesis, a memory replay mechanism is omitted to improve recommendation performance and to ensure training stability.

These three considerations have successfully ensured the stability of training through consistent Q-value updates (see Figures B.1 and B.2), and the mitigation of extrapolation errors (see Figures B.3 and B.4).

4.3 Challenges in Learning and Configurational Experiments

During the model development phase of this thesis, a challenge was encountered where the model failed to learn effectively. Specifically, monitoring the loss throughout training showed that it did not exhibit the expected gradual decrease. This behavior contrasts with the loss reduction observed in the final framework, as illustrated in Figures B.3 and B.4. Consequently, the model was unable to learn and generate valid location recommendations.

Suspecting that the model was underfitting, several adjustments were explored to enhance its performance:

4.1 Increasing the learning rate and the maximum number of epochs.

4.2 Incorporating a learning rate scheduler.

4.3 Deepening the DQN via two additional hidden layers with ReLU activation functions.

4.4 Introducing a drop-out layer to the DQN.

4.5 Reducing the state space by excluding features PTA, C, and H.

None of these approaches significantly affected model performance, leading to the investigation of more substantial changes to the framework, including:

5.1 *Removing the target network from the framework.* As recent research suggests, having a target network is not always necessary for a successful DQL framework [1, 33, 48]. Although these research are in vastly different domains than the business setting, this approach was tested in light of these studies.

5.2 *Adding a memory replay mechanism.* As discussed in Subsection 4.2.3, memory replay mechanisms can lead to extrapolation errors and thus potentially cause divergence. Nonetheless, this modification was explored in an attempt to maximize the learning capacity of the agent by enabling it to store and directly utilize past experiences.

Unlike the initial modifications, these changes affected the model's behavior by reducing recommendation performance. Double DQN [58] and Dueling DQN [60] frameworks were also experimented with, hypothesizing that more complex architectures might better capture the intricacies of the business site selection problem. However, these approaches also failed to produce satisfactory results.

Lastly, outliers were removed from the dataset. Surprisingly, this led to a significant decline in recommendation performance. Further analysis revealed that exposing the agent to a diverse range of scenarios was crucial for effective learning, as later discussed in Subsection 5.4.4. Examining the variation in other parameters, it was discovered that the reward values were overly concentrated due to the use of min-max scaling. By switching to percentile scaling, the reward values became differentiated, allowing the model to experience a broader range of scenarios. This diversity in the agent's experiences enabled the model to generate successful recommendations, and the final configuration of the framework detailed in Section 4.2 was formed.

5. EXPERIMENTS

Experiments are performed on the 13 most populated U.S. cities of 2019 for the business categories restaurants and other eating places and grocery stores from January 2019 to March 2020. The cities are as follows:

- Atlanta, Georgia
- Austin, Texas
- Boston, Massachusetts
- Chicago, Illinois
- Cleveland, Ohio
- Dallas, Texas
- Houston, Texas
- Los Angeles, California
- Manhattan, New York
- Philadelphia, Pennsylvania
- Phoenix, Arizona
- Sacramento, California
- Tampa, Florida

For New York City, the experiments are conducted exclusively in the Manhattan borough to ensure that the results are more comparable to other studies that focus mostly on Manhattan rather than the entire city [8].

The cities are studied at the neighborhood level, meaning every neighborhood is analyzed as a candidate location. The visit counts and spending per neighborhood are aggregated into a single scalar value, representing the total sum of visit counts and spending across all businesses within the specified category in each neighborhood. Census data is also utilized at the neighborhood level.

5.1 Experimental Protocol

As a model-free method, DQL does not simulate the analyzed environments. Instead, it relies solely on observations of the environment. In other words, the cities studied are not simulated but are observed through the available data. The model operates under the following assumptions:

- If and when the agent decides to open a business, it is presupposed that the requisite resources and opportunities are available.
- The model's decision to open a business does not have a consequential impact on the surrounding environment.

Sixty neighborhoods with available data are selected randomly from each city with the following considerations:

- Selected neighborhoods should not include military bases and international airports. To determine the location of military bases and international airports, the OSMnx library is utilized via the tags `'landuse': 'military'` and `'aeroway': 'aerodrome'`. This exclusion mimics the real-world setup by confining the model to realistic candidate locations.
- The selected neighborhoods should have more than 250 days of data. This minimum length criteria ensures that the model will have enough data for it to properly learn.

Additionally, to ensure consistency in the model's input, the same number of neighborhoods is sampled from each city. This approach guarantees that the state, which is fed as input into the DQN, maintains a uniform shape across all cities. A standardized state shape allows the DQN trained in one city to be applied to generate recommendations in another, without the need for additional preprocessing to adapt the input size. The number of sampled neighborhoods, set at 60, is determined based on data availability. Specifically, 60 is the maximum number of neighborhoods that could be selected while ensuring at least three months of available data for all cities. Three months is determined as the threshold here as any data length below it does not allow the agent to properly explore and learn.

For the business category of restaurants and other eating places, a model is trained on data from one of the 13 cities and then tested on the remaining 12. For the business category of grocery stores, the lack of available data has limited the experiments to 10 cities; with the exclusion of Atlanta, Boston, and Cleveland. The setup of training and testing the model in different cities evaluates the framework's generalization capability and assesses whether an RL approach can transfer its learning across different socio-demographic attributes, geographical patterns, and consumer habits.

5.1.1 Recommendation performance measurement

Profit Indicators. The profit indicators of businesses are calculated by subtracting the median rent of their neighborhood from their total revenue. The median rent here is used as a proxy for costs concerning geographical considerations.

Ground Truth Locations. The model recommendations are evaluated by comparing them to the locations that have the highest profit indicators and noted by GT for ground truth locations. The ground truth neighborhoods, I^{GT}, are the k neighborhoods with the highest total profit indicators through the analysis timeframe.

Performance Metric. The recommendation performance is measured by the percentage of rewards (profit indicators) that the agent earned with its recommendations compared to the highest possible rewards, created for this study and named the Profit Realization Score (PRS). Through our analysis, we measured the PRS over the top 1, 2, 3, 4, and 5 recommendations. PRS is calculated as follows:

$$PRS@k = \frac{\sum_{i \in I^{\star}} \sum_t r_t^i}{\sum_{i \in I^{\mathrm{GT}}} \sum_t r_t^i} \tag{5.1}$$

PRS is chosen as the performance metric over more established metrics like Precision@k and Recall@k, as it better aligns with the primary objective of the model: maximizing potential business success. While Precision@k and Recall@k allow for the comparison of the model's ranking of neighborhoods against the actual rankings based on profit indicators, these metrics fail to fully capture the model's purpose. The limitation of these metrics arises from the narrow profit differentials often observed among top-ranked neighborhoods. For instance, consider three top neighborhoods (A, B, and C) with actual profit indicators of 1015, 1010, and 1000, respectively. If the model recommends neighborhoods D, E, and F with profit indicators of 995, 990, and 985, both Recall@3 and Precision@3 would yield a score of 0. This outcome suggests extremely poor model performance, even though the model successfully identified neighborhoods with high profit indicators. In contrast, PRS@3 would yield a score of 97.129%, reflecting that the model's recommendations are still valid and near-optimal in terms of profit potential. This example highlights how PRS provides a more accurate assessment of the model's performance relative to its objective. By quantifying the extent to which the model's recommendations approach the maximum achievable profit, PRS effectively evaluates the model's ability to maximize business success, making it the preferred metric for this study.

5.1.2 Hyperparameter selection

Overfitting vs Underfitting. The model is trained under Adaptive Moment Estimation (ADAM) optimization, with Mean Squared Error (MSE) loss between Q^{Target} and Q^{Pred} following Equation 3.6. The learning rate (α) is set to 0.01, and maximum number of epochs (M) to 500. The high learning rate and the low number of epochs are chosen to prevent overfitting and to ensure that the model trained in one city can be generalized to another, while still allowing the model to properly learn and not underfit.

Myopic vs. Far-Sighted Decision Making. The discount factor (γ) is set to the high value of 0.9, prohibiting the model from being myopic as per Equation 3.8. Since iterations are performed daily, it is of utmost importance that the model prioritizes long-term gains over immediate rewards. The recommendation framework aligns these long-term gains with the actual objective in the real-world setup, which is to maximize profits over several months.

Exploration vs Exploitation. The initial exploration rate (ϵ) is set to 0.9, exploration rate decay (ϵ^{decay}) to 0.9, the minimum exploration rate (ϵ^{min}) to 0.1. The high exploration rate of 0.9 allows the model to initially learn by studying random actions and their consequences. However, due to the limited length of the data, the low exploration rate decay of 0.9 allows the model to quickly shift from exploration to exploitation. Lastly, the high minimum exploration rate of 0.1 ensures that the model continues to explore, regardless of its progress in the learning process. This is crucial as the business environment does not offer a deterministic setup where every outcome roughly replicates previous patterns. At each step, the model can identify and learn new strategies.

Target Network Updates. The target network is updated every step to allow the Q-values to adapt to the rapidly changing patterns in the data. Additionally, in a setting with a relatively small dataset (compared to typical experimental RL setups), frequent updates improve model performance [46] and fasten model training [48]. The chosen update frequency ensures proper alignment between predicted and target Q-values (see Figures B.1 and B.2).

The hyper-parameters mentioned above are not tuned for every city or business category, they are kept the same. This ensures that the framework can be easily generalized by:

6.1 Having the exact same structure provide valid recommendations for a wide range of different cities and sectors.

6.2 Having a very efficient framework to re-apply both in terms of computation and time.

5.2 Benchmark Model

The Huff gravity model is selected as the benchmark given its proven effectiveness in business site selection in the current research [5, 38, 55]. The following Table 5.1 summarizes the notations used for the benchmark Huff gravity model.

Table 5.1 Notations for the Huff gravity model

Notations	**Meanings**
α	Weight of the Attractiveness
β	Weight of the Distance
A	Attractiveness
D	Distance Between Business and Visitor's Resident
e	Weight of Attractiveness Indicator
i	Attractiveness Indicator Index
I	Attractiveness Indicator
n	Number of Attractiveness Indicators

Using the same dataset and experimental protocol as the DQL framework, the Huff gravity model is used to predict the probability of visits to businesses. This probability is calculated using the distance between the business and the visitors' residents, and the business's attractiveness as detailed in Equation 2.2. Conventionally, the attractiveness of a candidate location is modeled by its store area. However, in this thesis, attractiveness is modeled as an interaction of different indicators to reflect not only the characteristics of a particular point of interest but also its home area. The indicators used, which are also the same variables incorporated in the DQL framework as part of the state, are: Competition (C), Median Dwell (DW), Expenditure (E), Business Diversity (H), Median Age (MA), Median Income (MI), Number of Families (NF), Population (P), and Public Transportation Accessibility (PTA).

The attractiveness of a location is defined as the multiplication of these indicators I with their exponents e controlling their weights and is calculated via the following equation:

$$A = \prod_{n} I_i^e \tag{5.2}$$

The distance (D) is used as is since it is readily available in the dataset. To optimize the exponents, particle swarm optimization technique is used which is an iterative optimization method in which a number of particles collectively search for the best parameter set within the search space. For this thesis, the search is performed in a 2-dimensional feature space (distance and store area) in 10 iterations with 20 particles. Cognitive and social parameters are both set to 1.5 while the inertia parameter is set to 0.9.

A Huff gravity model is trained using data from each city, and the exponents α and β are learned. The model is then tested on the remaining cities, where the probabilities of visits are estimated. These probabilities are multiplied by the actual visit counts to calculate the expected visits. Recommendations are made based on the candidate locations with the highest total expected visits. The effectiveness of these recommendations is evaluated using the same metric as the DQL framework, the PRS as detailed in Equation 5.1.

The benchmark model is developed by Hasan Alp Boz and further details can be found in [4].

5.3 Sample Experimental Case

For this thesis, a total of 246 experimental test cases were conducted. In each case, a model trained on data from one city and business category was utilized to generate location recommendations for another city, resulting in 246 distinct combinations. This section focuses on the specific case where the model was trained using data from Philadelphia, with recommendations generated in Houston for businesses in the restaurants and other eating places category. This example provides a clear illustration of the experimental process and is detailed in three steps: model training, recommendation generation, and performance measurement.

Step 1: Model Training

A total of 60 neighborhoods are randomly selected from the training city, Philadelphia. The main considerations for neighborhood selection are:

7.1 *Exclusion of Airports and Military Bases*: The neighborhoods containing Philadelphia International Airport and Northeast Philadelphia Airport are excluded to eliminate unrealistic candidate locations, as discussed in Section 5.1.

7.2 *Data Availability*: Each of the selected neighborhoods must have data available for every timestep within the analysis period. This consistency ensures that the state fed into the network is consistent in dimensions. The DQN requires a 1-dimensional state vector with 720 elements, comprising 12 features from each of the 60 neighborhoods, flattened into a single vector (see Equations 4.5 and 4.4).

The selected neighborhoods used for model training are illustrated in Figure 5.1.

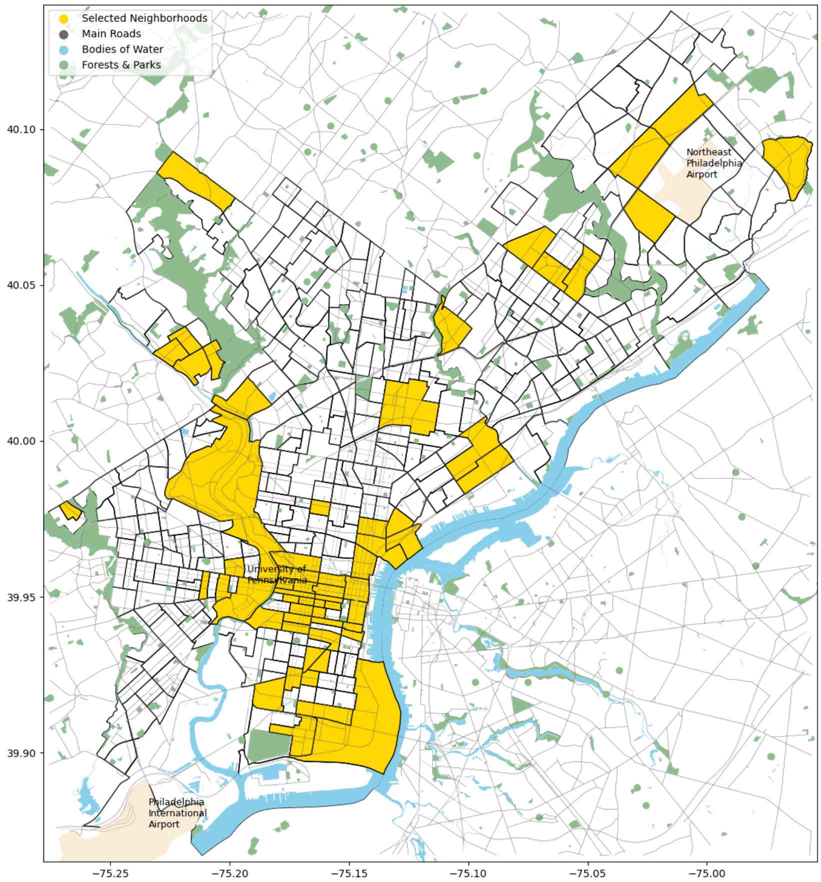


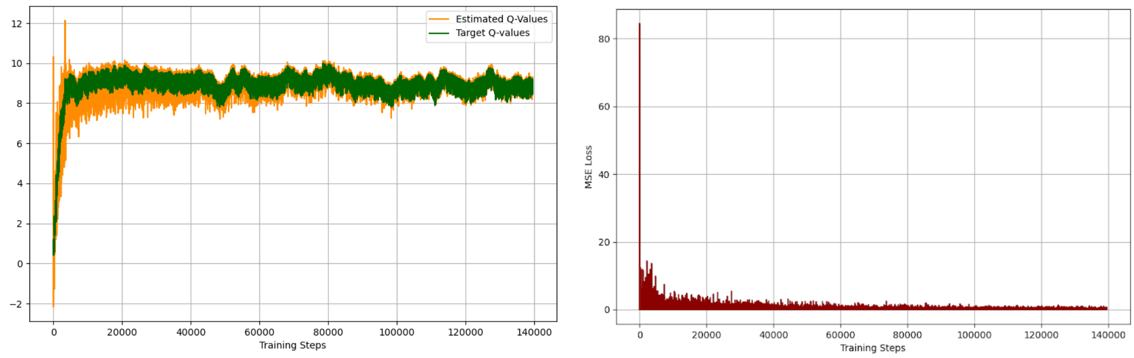
Figure 5.1 Selected neighborhoods from Philadelphia for model training

Selected neighborhoods, colored in yellow, excluding airports and military bases. The data from these neighborhoods will be used for *Model Training*.

The model is trained using the data from selected Philadephia neighborhoods, following Algorithm 1. The hyper-parameters are selected as; $\alpha = 0.01$, $\mathrm{M} = 500$, $\gamma = 0.9$, $\epsilon = 0.9$, $\epsilon^{\mathrm{min}} = 0.1$, $\epsilon^{\mathrm{decay}} = 0.9$, and Target Network Update $= 1$. These hyper-parameters are not tuned, and their selection is discussed in detail in Subsection 5.1.2.

The training is stable, as shown in Figure 5.2, demonstrating that the considerations for training stability detailed in Subsection 4.2.3 were effective.

The trained DQN is saved to be used later for recommendation generation.



(a) Estimated and target Q-values (b) Training loss

Figure 5.2 Training stability of Philadelphia for business category restaurants and other eating places

Over-estimation of Q-values or divergence is not present during training. Estimated Q-values match target Q-values, with increased precision as training progresses as observed in (a). The MSE loss decreases steadily without any unexpected spikes, as shown in (b).

Step 2: Recommendation Generation

A total of 60 neighborhoods are randomly selected from the training city, Houston. Similar to the neighborhood selection for model training, the neighborhoods containing George Bush Intercontinental Airport-Houston and William P. Hobby Airport are excluded, and the selected neighborhoods all have available data for every timestep of the analysis period.

The selected neighborhoods for recommendation generation are illustrated in Figure 5.3.

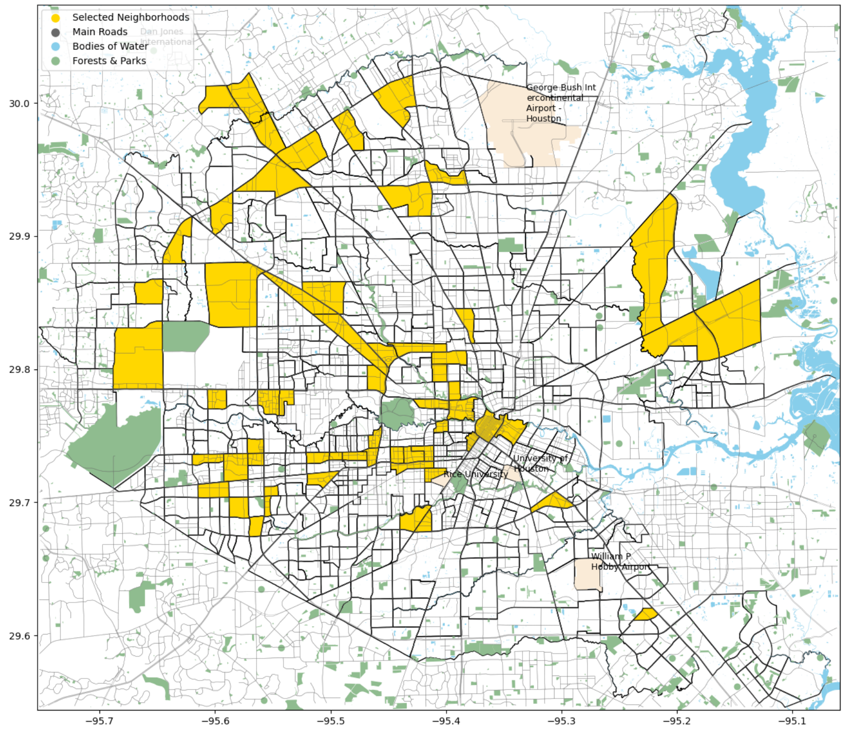


Figure 5.3 Selected neighborhoods from Houston for recommendation generation

Selected neighborhoods, colored in yellow, excluding airports and military bases. Location recommendations will be generated among these neighborhoods.

Recommended neighborhoods are selected among the selected Houston neighborhoods, using the trained DQN from *Step 1: Model Training*. The DQN is used as-is, no tuning or further learning is performed. The same hyper-parameters from the previous step are used. The recommended neighborhoods are selected following Algorithm 2, and are shown in Figure 5.4.

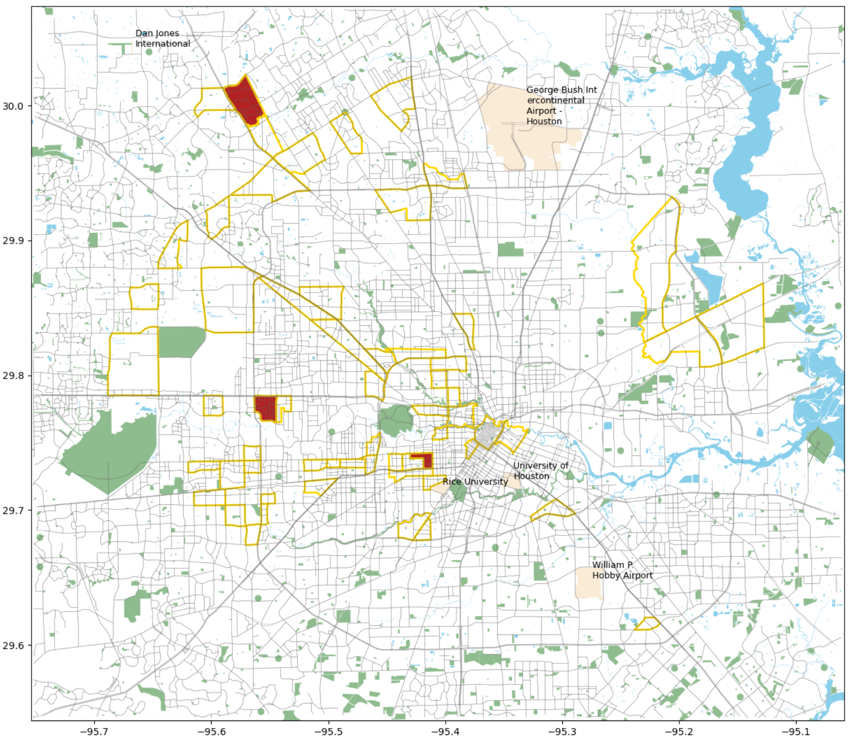


Figure 5.4 Model recommendations for Houston for the business category restaurants and other eating places from the model trained on Philadelphia

The neighborhoods outlined in yellow show the candidate locations, with neighborhoods colored in red showing the model's recommendations.

Step 3: Recommendation Performance Measurement

Recommendation performance is measured in comparison to GT locations (the locations that yield the highest aggregated profit indicators over the analysis period) using the PRS metric as detailed in Subsection 5.1.1. The model yields a 97.30% PRS@1, 97.54% PRS@2, 95.55% PRS@3, 96.37% PRS@4, and 95.98% PRS@5 with its recommendations compared to GT profit indicators depicted in 5.5. This essentially means that the top neighborhood recommended by the model would have earned 97.30% of the profits, the top two 97.54%, the top three 95.55%, the top four 96.37%, and the top five 95.98% compared to the best possible locations assuming that businesses were established in the recommended locations.

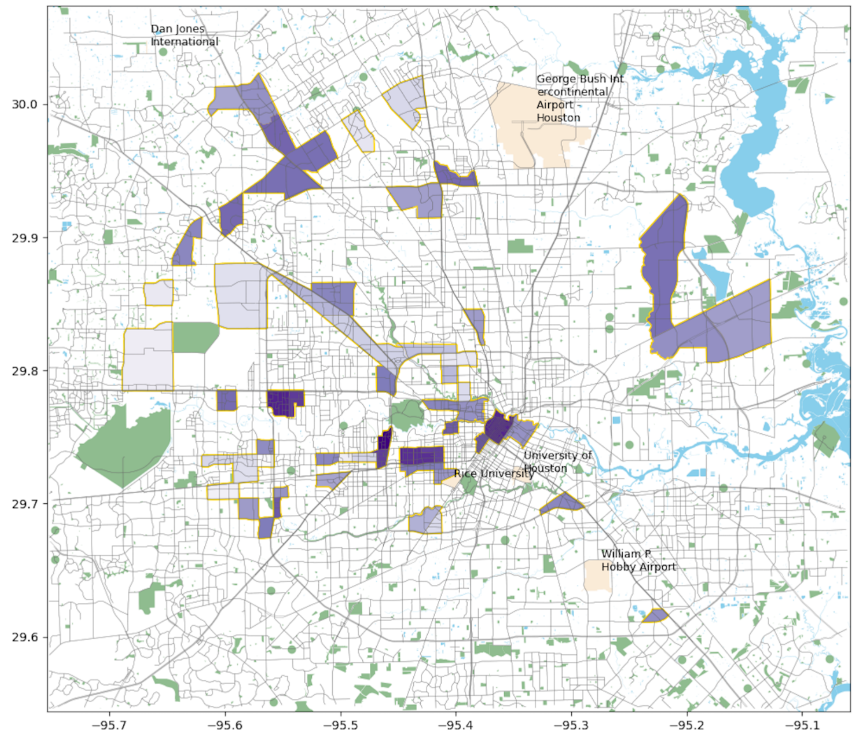


Figure 5.5 Profit indicators for Houston for the business category restaurants and other eating places

Purple denotes the profit indicators with a darker purple indicating higher profits.

The flowchart of the DQL framework from Figure 4.1 is extended for this run in Figure 5.6. In Step 1 of Figure 5.6, data from the selected Philadelphia neighborhoods are used to train the prediction network. In Step 2 of the figure, the trained prediction network from Step 1 is used to generate the set of recommended neighborhoods, I^*, in Houston. In Step 3, the performance of the recommendations is measured in comparison to GT locations via PRS.

5.4 Results

In this section, recommendation performance is investigated using the top three recommendations ($PRS@3$), for clarity purposes. Performance metrics $PRS@1$, $PRS@2$, $PRS@4$, and $PRS@5$ are also reported in Appendix A. Across both business categories, the metrics are consistent, with an expected average improvement of 1.5% in $PRS@k$ as k increases.

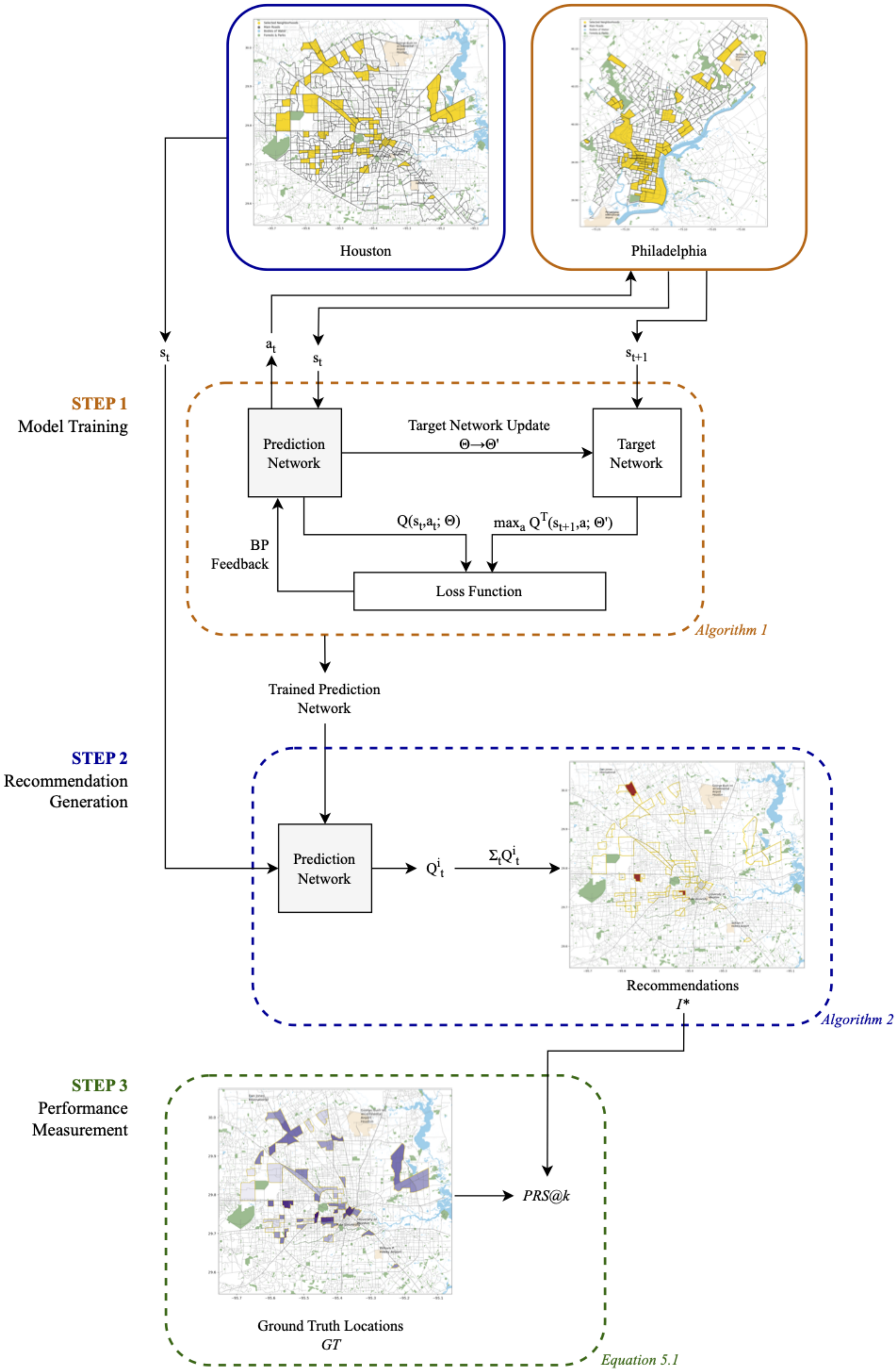


Figure 5.6 Flowchart of the sample experimental run

5.4.1 Recommendation performance

Experimental results demonstrate the generalization capacity of the proposed DQL framework, achieving an average $PRS@3$ of 81.28% for restaurants and other eating places businesses and 72.37% for grocery stores. The benchmark Huff gravity model significantly underperforms with an average $PRS@3$ of 63.99% for restaurant and other eating places businesses and 60.44% for grocery stores. The recommendation performances are summarized in tables 5.2 and 5.3 for business categories restaurant and other eating places and grocery stores, respectively (see Appendix A for detailed results).

Table 5.2 Average PRS@3 according to training cities for business category restaurants and other eating places

City	DQL Framework	Huff Gravity Model
Atlanta	0.815865	0.591196
Austin	0.786978	0.685410
Boston	0.834806	0.687403
Chicago	0.857097	0.670576
Cleveland	0.793917	0.629739
Dallas	0.831589	0.535682
Houston	0.788485	0.665810
Los Angeles	0.799311	0.590428
Manhattan	0.795311	0.683671
Philadelphia	0.805368	0.619000
Phoenix	0.804590	0.654019
Sacramento	0.820437	0.660319
Tampa	0.832082	0.645476
Average	**0.812757**	0.639902

5.4.2 Recommendation consistency

In addition to its performance, the DQL framework offers consistent recommendations, as shown in Figure 5.7. While the performance of the DQL framework slightly varies across different test cases, it reliably delivers valid and profitable recommendations in contrast to the Huff gravity model. The Huff gravity model shows greater inconsistency and its performance is greatly affected by the city pairs, the cities that the model is trained and tested on. Moreover, the generalization capacity of the DQL framework remains consistent across both business categories. Overall, the

Table 5.3 Average PRS@3 according to training cities for business category crocery stores

City	DQL Framework	Huff Gravity Model
Austin	0.751357	0.628454
Chicago	0.728124	0.611001
Dallas	0.684461	0.620690
Houston	0.723346	0.585236
Los Angeles	0.692142	0.616801
Manhattan	0.731577	0.603258
Philadelphia	0.727550	0.546023
Phoenix	0.748932	0.605991
Sacramento	0.764139	0.626143
Tampa	0.685346	0.600353
Average	**0.723697**	0.604396

analysis of recommendation consistencies demonstrates the generalization strength of the DQL framework across cities and sectors over the Huff gravity model.

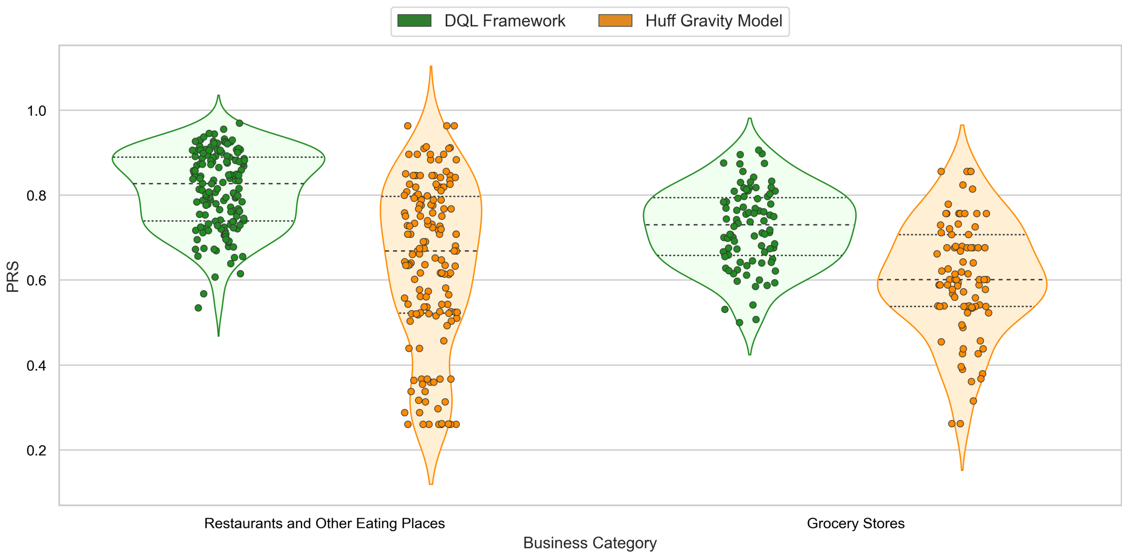


Figure 5.7 Violin plots of recommendation performances

The Huff gravity model generates recommendations with significant variability in profitability. In contrast, the DQL framework consistently delivers profitable recommendations across all test cases, demonstrating its adaptability across different cities.

5.4.3 Recommendation performance and spatial characteristics

To investigate the relationship between recommendation performance and the spatial characteristics of the city pairs (i.e.. the cities in which the model has been trained and tested), two metrics are utilized to quantify similarity: flow hierarchy [6], and orientation order indicator [8]. The analysis revealed no significant correlation between the model's performance and either the flow hierarchy or orientation order indicator of the city pairs, as shown in Table 5.4.

Table 5.4 Correlation between recommendation performance and spatial order

Business Category	Orientation Order Indicator	Flow Hierarchy
Restaurants and other eating places	-0.14	0.29
Grocery stores	-0.21	0.23
All business categories	-0.15	-0.29

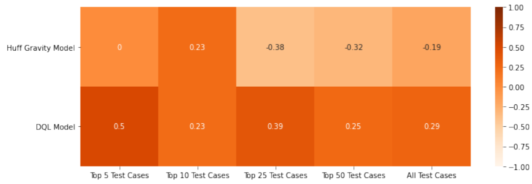
Pearson's correlation coefficients between PRS and the city pairs' absolute difference of orientation order indicator and flow hierarchy

Additionally, when the top-performing test cases were analyzed separately, no correlation was found between the model's performance and the spatial characteristics of the city pairs. As depicted in Figure 5.8, the correlation between recommendation performance and city similarities is insignificant and does not represent a pattern, indicating that the spatial similarity between city pairs is irrelevant to model performance.

5.4.4 Recommendation performance and variation in data

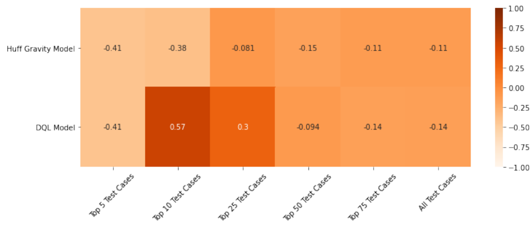
Analysis of the results reveals a single consideration that directly affects recommendation performance: The variance of the data. The variance of data influences the model performance in the following ways:

8.1 Recommendation performance improves as the variance of the training dataset increases. A training dataset with a large variance allows the model to encounter a diverse subset of (s, a, r) values, enabling it to make correct decisions based on training experiences when encountering similar scenarios in the test data.

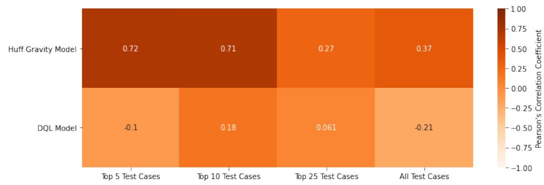


(a) Flow hiearchy - restaurants

(b) Flow hiearchy - grocery stores



(c) Orientation order indicator - restaurants



(d) Orientation order indicator - grocery stores

Figure 5.8 Pearson's correlation coefficients between PRS and the city pairs' absolute difference of orientation order indicator and flow hierarchy.

8.2 The model also provides profitable recommendations when the variance of the training dataset is small, provided that the variance of the test dataset is also small. In high-performing test cases with a training dataset exhibiting a relatively small coefficient of variation, the test dataset has a low variance as well. This similarity in variance once again allows the model to accurately predict the outcome of its decisions based on similar sequences it has experienced.

8.3 The low variance of certain features in the model's state, whether in the training or test datasets, does not affect the recommendation performance at all. This is due to the fact that like any neural network, the DQN is robust to irrelevant features. Surely, this robustness minimizes the need for business domain knowledge or extensive pre-processing by eliminating the need to filter out irrelevant features via prior knowledge or feature subset selection.

Feature Importance. Based on the hypothesis that the model performance is dependent on the variance of data, the coefficient of variance for features of the model's state for top and bottom-performing three training cities are studied. For restaurants and other eating places, the coefficient of variance for the features of distance to visitors' home, median income, and median dwell exhibits the most significant differences between the top and bottom training cities, with an average difference of 0.4597, 0.1434, and 0.1233, respectively. For grocery stores, the coefficient of variance for the features of public transportation accessibility and competition shows the greatest differences between the top and bottom training cities, with an average difference of 0.5650, and 0.1258, respectively. These findings indicate that for both business categories, geographical accessibility of the businesses significantly enhances their attractiveness.

Performance Difference between Business Categories. Lastly, the analysis of experimental results indicates that the model does not perform well if trained with less than four months of data. With experience-based learning, a data length of less than 4 months does not allow the model to explore and correct its actions properly. For the business category of grocery stores, the data is significantly more limited compared to the category of restaurants and other eating places, resulting in a performance gap between the two categories.

6. CONCLUSION

This thesis proposes a DQL framework to address the complex problem of business site selection. The main objective is to develop a generalizable framework that can be adapted to various contexts, enabling the transfer of insights across time, regional regulations, and spatial and social characteristics in the domain of business site selection.

6.1 Summary of Work

The proposed approach is a two-step framework. In step 1, named *Model Training*, a DQN is trained. The state (S) is defined as the business diversity, competition, expenditure, transportation accessibility, visits, and socio-demographic data of all candidate locations. The reward (R) is formulated as the differential between total revenue and median rent, where the latter serves as a proxy for operational costs and thus the reward mimics the profitability of a location. In each iteration, the state passes through the DQN, and Q-values of candidate locations are produced. The action (A) or namely the location with the highest Q-value is selected and the corresponding reward is received. In this step, the learning starts by taking random actions and learning from the consequences, a process called exploration in RL. Over the iterations, the agent explores less and emphasizes what it has already learned via exploitation. Overall, in this step the agent aims to create the optimum strategy for business site selection by studying the experiences, (s, a, r) tuples, it encounters. This step includes a target network and excludes a memory replay mechanism to ensure training stability. In Step 2, named *Recommendation Generation*, the trained network is used to generate Q-values for candidate locations. Each iteration, the state, consisting of data from each candidate location, is fed into the network to produce corresponding Q-values. At the end of the analysis period, the Q-values for each candidate location are summed, and the top k locations with the highest aggregated Q-values are returned as the model's recommendations. In this step, no learning is done, and the DQN is not backpropagated.

The experiments are performed on the 13 most populous US cities (Atlanta, Austin, Boston, Chicago, Cleveland, Dallas, Houston, Los Angeles, Manhattan, Philadelphia, Phoenix, Sacramento, Tampa), for business categories restaurants and other eating places and grocery stores, between dates January 2019 to March 2020 on the neighborhood level. For each business category, a model is trained using data from one city and then utilized to generate location recommendations in another. The framework and all its components, including hyper-parameters, input data features, and parameter formulations, remain the same throughout both the *Model Training* and *Recommendation Generation* phases. This experimental setup is designed to evaluate the framework's generalizability and transferability across varying environments. The Huff gravity model is employed as the benchmark. Using the same data as the DQL model, the Huff gravity model is calibrated using data from each city and then used to predict foot traffic at candidate locations in other cities. Locations are then recommended based on the highest predicted footfall. The performance of the recommendations is assessed using the Profit Realization Score $PRS@k$, a metric developed for this thesis. The PRS quantifies the percentage of profits generated by the top k recommendations relative to the maximum ground truth profits of the top k neighborhoods.

The proposed DQL framework achieved an average $PRS@3$ of 81.28% for restaurants and other eating establishments, and 72.37% for grocery stores. This performance surpassed that of the Huff gravity model by 17.29% and 11.93%, respectively. Moreover, the DQL framework consistently generated valid recommendations across all experimental cases, while the performance of the Huff gravity model exhibited considerable variability and inconsistency. In this thesis, it is hypothesized that the recommendation performance of the DQL framework is not influenced by the demographic, spatial, or proximity similarities between the city pairs used for training and testing. Instead, the performance is determined by a single factor: the variation in the data. Experimental results indicate that the performance of the proposed method's recommendations is contingent upon the variance of the distributions of state (S), action (A), and reward (R) in the training data being at least as high as in the test data. Since the model's learning is based on the (s, a, r) tuples encountered during training, it provides appropriate recommendations in testing as long as it is presented with (s, a, r) tuples similar to what it has already experienced during training.

Overall, this thesis demonstrates that RL-based models offer generalizable and readily reapplicable solutions to the business site selection problem. Furthermore, it is shown that an RL model developed in a specific context can generate valid recommendations in new settings without the need for additional processing or calibration. This thesis hypothesizes that the generalization capacity of the proposed DQL framework compared to other techniques stems from the very nature of RL. Unlike the Huff gravity model or other supervised machine learning methods, RL does not directly learn the relationships between model inputs (candidate location characteristics) and outputs (location profitability). In a business context, where these relationships are complex and continually changing, direct adaptation to different settings is extremely difficult, and at times even illogical. Instead, RL learns the optimal decision (A) to make in each situation (S) based on the resulting consequences (R), rather than directly learning the complex relationship between inputs and outputs. The strategies the model acquires, determining appropriate actions in varying contexts, appear to remain relatively consistent across different settings, thereby allowing generalization and reapplication.

6.2 Future Research

In future work, the existing model-free recommendation framework can be extended to a model-based approach. The current framework solely relies on observational data, without modeling or simulating cities. Consequently, location selections are hypothetical, and the assumption is made that opening a new business in a given location has no significant impact on the broader urban environment. A more sophisticated model-based RL framework can be developed to simulate cities, enabling the observation of the effects of business openings.

In a model-based framework, each component of the system would be represented by an agent. First, an agent to represent customers would be developed. This customer agent would have socio-demographic attributes, such as income, age, education level, gender, and ethnicity. Based on these attributes, the agent would visit and spend at various businesses over the designated analysis period. Secondly, a business agent would be implemented to represent enterprises within the system. The business agent would have attributes that represent operating costs, workforce characteristics, transportation accessibility, competition, and business diversity. The business agent would attract customers and generate revenue based on these attributes. Lastly, a regulatory agent would be developed to simulate the role of governance, enforcing regulations, imposing taxes, and supervising business operations.

Attributes for customer and business agents could be derived from real-world data sources. Specifically, the data on demographics (publicly available in American Community Surveys [10]) and the businesses (publicly available in Statistics of U.S. Businesses [11]) enables the creation of realistic attribute distributions, allowing agents to draw from these distributions to simulate real-world conditions.

The proposed agents would engage in interactions with one another, enabling a dynamic simulation of the city's retail business ecosystem. The nature of agent interactions could be determined either by learning from empirical data or by setting predefined rules. The interactions between customer and business agents, which are typically non-deterministic and volatile in the real world, would benefit from data-driven modeling, to better capture the stochastic nature of consumer behavior. Such modeling would, however, necessitate access to long-term, high-quality datasets, such as those available through the SafeGraph Data Consortium [51, 52]. The interactions involving regulatory agents would likely be more deterministic and structured. These interactions could be effectively captured by a predefined set of rules that involve tax regulations, labor law, business hours, and accessibility features of the physical business. This combination of learned and rule-based interactions would allow for an effective representation of urban business dynamics.

Overall, a model-based RL framework offers significant extensions to this thesis for addressing business site selection by simulating an urban environment in which decisions about business establishment can be tested. In this simulated environment, decisions to establish a new business could be made by virtually placing a business in a candidate location and observing the revenue it generates. In addition to location selection, a model-based framework would facilitate analysis of how a new business integrates with the existing urban dynamics, attracts foot traffic, influences neighboring businesses, and contributes to broader economic outcomes. This thesis demonstrates that RL-based approaches are well-suited for the business site selection problem. The proposed model-based extension could enhance existing research by incorporating long-term effects and improving interpretability via the comparison of simulated outcomes of various candidate locations.

BIBLIOGRAPHY

[1] Joshua Achiam, Ethan Knight, and Pieter Abbeel. Towards characterizing divergence in deep q-learning. *arXiv preprint arXiv:1903.08894*, 2019.

[2] Alaa Alden Al Mohamed, Sobhi Al Mohamed, and Moustafa Zino. Application of fuzzy multicriteria decision-making model in selecting pandemic hospital site. *Future business journal*, 9(1):14, 2023.

[3] Martin Aruldoss, T Miranda Lakshmi, and V Prasanna Venkatesan. A survey on multi criteria decision making methods and its applications. *American Journal of Information Systems*, 1(1):31–43, 2013.

[4] Atra Zeynep Bahceci, Hasan Boz, and Selim Balcisoy. Business site recommendation with deep q-learning. In *Book of Abstracts of the NetMob Conference*, pages 52–53, 2024. Abstract only.

[5] Mohsen Bahrami, Yilun Xu, Miles Tweed, Burcin Bozkaya, et al. Using gravity model to make store closing decisions: A data driven approach. *Expert systems with applications*, 205:117703, 2022.

[6] Aleix Bassolas, Hugo Barbosa-Filho, Brian Dickinson, Xerxes Dotiwalla, Paul Eastham, Riccardo Gallotti, Gourab Ghoshal, Bryant Gipson, Surendra A Hazarie, Henry Kautz, et al. Hierarchical organization of urban mobility and its connection with city livability. *Nature communications*, 10(1):4817, 2019.

[7] Tugce Bilen, Muge Erel-Özçevik, Yusuf Yaslan, and Sema F Oktug. A smart city application: Business location estimator using machine learning techniques. In *2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, pages 1314–1321. IEEE, 2018.

[8] Geoff Boeing. Urban spatial order: Street network orientation, configuration, and entropy. *Applied Network Science*, 4(1):1–19, 2019.

[9] United States Census Bureau. American community survey. https://www.census.gov/data/developers/data-sets/acs-5year.html, 2021. Accessed: 2023-11-11.

[10] United States Census Bureau. American community survey. https://www.census.gov/programs-surveys/acs/data.html, 2024. Accessed: 2024-10-10.

[11] United States Census Bureau. Statistics of u.s. business. https://www.census.gov/programs-surveys/susb/data.html, 2024. Accessed: 2024-10-10.

[12] Oriol Cabanas-Tirapu, Lluís Danús, Esteban Moro, Marta Sales-Pardo, and Roger Guimerà. Human mobility is well described by closed-form gravity-like models learned automatically from data. *arXiv preprint arXiv:2312.11281*, 2023.

[13] Salvatore Carta, Anselmo Ferreira, Alessandro Sebastian Podda, Diego Reforgiato Recupero, and Antonio Sanna. Multi-dqn: An ensemble of deep q-learning agents for stock market forecasting. *Expert systems with applications*, 164:113820, 2021.

[14] Wei Chen, Yajun Wang, and Siyu Yang. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 199–208, 2009.

[15] Joel P Dock, Wei Song, and Jia Lu. Evaluation of dine-in restaurant location and competitiveness: Applications of gravity modeling in jefferson county, kentucky. *Applied Geography*, 60:204–209, 2015.

[16] Zvi Drezner and Dawit Zerom. A refinement of the gravity model for competitive facility location. *Computational Management Science*, 21(1):2, 2024.

[17] Bahaeddin Eravci, Neslihan Bulut, Cagri Etemoglu, and Hakan Ferhatosmanoğlu. Location recommendations for new businesses using check-in data. In *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, pages 1110–1117. IEEE, 2016.

[18] Hikmet Erbıyık, Selami Özcan, and Kazım Karaboğa. Retail store location selection problem with multiple analytical hierarchy process of decision making an application in turkey. *Procedia-Social and Behavioral Sciences*, 58:1405–1414, 2012.

[19] Office for National Statistics. Employees in the uk by industry: 2019. https://www.ons.gov.uk/employmentandlabourmarket/peopleinwork/employmentandemployeetypes/bulletins/employeesintheukbyindustry/2019, 2020. Accessed: 2024-10-02.

[20] India Brand Equity Foundation. Indian retail industry analysis, 2024. Accessed: 2024-10-21.

[21] Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*, pages 2052–2062. PMLR, 2019.

[22] Yaozhong Gan, Zhe Zhang, and Xiaoyang Tan. Stabilizing q learning via soft mellowmax operator. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7501–7509, 2021.

[23] Shuihua Han, Linlin Chen, Zhaopei Su, Shivam Gupta, and Uthayasankar Sivarajah. Identifying a good business location using prescriptive analytics:

Restaurant location recommendation based on spatial data mining. *Journal of Business Research*, 179:114691, 2024.

[24] Shuihua Han, Xinyun Jia, Xinming Chen, Shivam Gupta, Ajay Kumar, and Zhibin Lin. Search well and be wise: A machine learning approach to search for a profitable location. *Journal of Business Research*, 144:416–427, 2022.

[25] Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, pages 3207–3214, 2018.

[26] Harold Hotelling. Stability in competition. *The Economic Journal*, 39(153):41–57, 1929.

[27] Jin Huang, Zeyi Wen, Mukaddim Pathan, Kerry Taylor, Yuan Xue, and Rui Zhang. Ranking locations for facility selection based on potential influences. In *IECON 2011-37th Annual Conference of the IEEE Industrial Electronics Society*, pages 2411–2416. IEEE, 2011.

[28] Jin Huang, Zeyi Wen, Jianzhong Qi, Rui Zhang, Jian Chen, and Zhen He. Top-k most influential locations selection. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 2377–2380, 2011.

[29] David L Huff. A probabilistic analysis of shopping center trade areas. *Land economics*, 39(1):81–90, 1963.

[30] Quentin JM Huys, Anthony Cruickshank, and Peggy Seriès. Reward-based learning, model-based and model-free. In *Encyclopedia of Computational Neuroscience*, pages 1–10. Springer New York, 2014.

[31] Nurul Indarti. Business location and success: the case of internet cafe business in indonesia. *Gadjah Mada International Journal of Business*, 6(2):171–192, 2004.

[32] Dmytro Karamshuk, Anastasios Noulas, Salvatore Scellato, Vincenzo Nicosia, and Cecilia Mascolo. Geo-spotting: mining online location-based services for optimal retail store placement. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 793–801, 2013.

[33] Ethan Knight and Osher Lerner. Natural gradient deep q-learning. *arXiv preprint arXiv:1803.07482*, 2018.

[34] Zhou Lei and An Ye. Research on location selection of super mall based on gis technology and huff model. In *International Conference on Economic Management and Green Development (ICEMGD 2018)*, pages 194–203, 2018.

[35] Guoliang Li, Shuo Chen, Jianhua Feng, Kian-lee Tan, and Wen-syan Li. Efficient location-aware influence maximization. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 87–98, 2014.

[36] Jing Li, Bin Guo, Zhu Wang, Mingyang Li, and Zhiwen Yu. Where to place the next outlet? harnessing cross-space urban data for multi-scale chain store recommendation. In *Proceedings of the 2016 ACM international joint conference on pervasive and ubiquitous computing: adjunct*, pages 149–152, 2016.

[37] Yingru Li and Lin Liu. Assessing the impact of retail location on store performance: A comparison of wal-mart and kmart stores in cincinnati. *Applied Geography*, 32(2):591–600, 2012.

[38] Yunlei Liang, Song Gao, Yuxin Cai, Natasha Zhang Foutz, and Lei Wu. Calibrating the dynamic huff model for business analysis using location big data. *Transactions in GIS*, 24(3):681–703, 2020.

[39] Ping Liu, Meng Wang, Jiangtao Cui, and Hui Li. Top-k competitive location selection over moving objects. *Data Science and Engineering*, 6(4):392–401, 2021.

[40] Shiwei Lu, Shih-Lung Shaw, Zhixiang Fang, Xirui Zhang, and Ling Yin. Exploring the effects of sampling locations for calibrating the huff model using mobile phone location data. *Sustainability*, 9(1):159, 2017.

[41] Sanja Marinković, Ilija Nikolić, and Jovana Rakićević. Selecting location for a new business unit in ict industry. *Zbornik radova Ekonomskog fakulteta u Rijeci/Proceedings of Rijeka Faculty of Economics*, 36(2):801–825, 2018.

[42] Amir massoud Farahmand, Azad Shademan, Martin Jagersand, and Csaba Szepesvári. Model-based and model-free reinforcement learning for visual servoing. In *2009 IEEE International Conference on Robotics and Automation*, pages 2917–2924. IEEE, 2009.

[43] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[44] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[45] Office for National Statistics. Gdp output approach – low-level aggregates. https://www.ons.gov.uk/economy/grossdomesticproductgdp/datasets/ukgdpolowlevelaggregates, 2024. Accessed: 2024-10-02.

[46] Shota Ohnishi, Eiji Uchibe, Yotaro Yamaguchi, Kosuke Nakanishi, Yuji Yasui, and Shin Ishii. Constrained deep q-learning gradually approaching ordinary q-learning. *Frontiers in neurorobotics*, 13:103, 2019.

[47] Błażej Osiński and Konrad Budek. What is reinforcement learning? deepsense.ai's complete guide, 2018. Accessed: 2024-08-12.

[48] Alexandre Piché, Joseph Marino, Gian Maria Marconi, Valentin Thomas, Christopher Pal, and Mohammad Emtiyaz Khan. Beyond target networks: Improving deep q-learning with functional regularization. https://openreview.net/

forum?id=JhTPL9BiYy, 2021. Under review as a conference paper at ICLR 2022.

[49] PwC. The economic contribution of the u.s. retail industry. Technical report, Prepared for National Retail Federation, March 2024. Accessed: 2024-10-02.

[50] Derek T Robinson and Bogdan Caradima. A multi-scale suitability analysis of home-improvement retail-store site selection for ontario, canada. *International Regional Science Review*, 46(1):69–97, 2023.

[51] SafeGraph. Spend. https://docs.safegraph.com/docs/spend, 2021. Accessed: 2023-11-11.

[52] SafeGraph. Weekly patterns. https://docs.safegraph.com/docs/weekly-patterns, 2021. Accessed: 2023-11-11.

[53] Salman Ahmed Shaikh, Mohsin Memon, and Kyoung-Sook Kim. A multi-criteria decision-making approach for ideal business location identification. *Applied sciences*, 11(11):4983, 2021.

[54] Rafael Suárez-Vega, José Luis Gutiérrez-Acuna, and Manuel Rodríguez-Díaz. Locating a supermarket using a locally calibrated huff model. *International Journal of Geographical Information Science*, 29(2):217–233, 2015.

[55] Yoshihiko Suhara, Mohsen Bahrami, Burcin Bozkaya, and Alex ‘Sandy’ Pentland. Validating gravity-based market share models using large-scale transactional data. *Big Data*, 9(3):188–202, 2021.

[56] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction.* MIT press, 2018.

[57] Miao Tian, Zhiwen Yu, Zhu Wang, and Bin Guo. Combining social media and location-based services for shop type recommendation. In *Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers*, pages 161–164, 2015.

[58] Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, pages 2094–2100, 2016.

[59] Feng Wang, Li Chen, and Weike Pan. Where to place your next restaurant? optimal restaurant placement via leveraging user-generated reviews. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 2371–2376, 2016.

[60] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Hasselt, Marc Lanctot, and Nando Freitas. Dueling network architectures for deep reinforcement learning. In *International conference on machine learning*, pages 1995–2003. PMLR, 2016.

[61] Mengwen Xu, Tianyi Wang, Zhengwei Wu, Jingbo Zhou, Jian Li, and Haishan Wu. Demand driven store site selection via multiple spatial-temporal data. In

Proceedings of the 24th acm sigspatial international conference on advances in geographic information systems, pages 1–10, 2016.

[62] Yanan Xu, Yanyan Shen, Yanmin Zhu, and Jiadi Yu. Ar2net: An attentive neural approach for business location selection with satellite data and urban data. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 14(2):1–28, 2020.

[63] Jeremy YL Yap, Chiung Ching Ho, and Choo-Yee Ting. Analytic hierarchy process (ahp) for business site selection. In *AIP Conference Proceedings*, volume 2016, 2018.

[64] Kamil Żbikowski and Piotr Antosiuk. A machine learning, bias-free approach for predicting business success using crunchbase data. *Information Processing & Management*, 58(4):102555, 2021.

[65] Qian Zeng, Ming Zhong, Yuanyuan Zhu, Tieyun Qian, and Jianxin Li. Business location planning based on a novel geo-social influence diffusion model. *Information Sciences*, 559:61–74, 2021.

[66] Jiejie Zhao, Bowen Du, Leilei Sun, Weifeng Lv, Yanchi Liu, and Hui Xiong. Deep multi-task learning with relational attention for business success prediction. *Pattern Recognition*, 110:107469, 2021.

[67] Sarfaraz Hashemkhani Zolfani, Mohammad Hasan Aghdaie, Arman Derakhti, Edmundas Kazimieras Zavadskas, and Mohammad Hossein Morshed Varzandeh. Decision making on business issues with foresight perspective; an application of new hybrid mcdm model in shopping mall locating. *Expert systems with applications*, 40(17):7111–7121, 2013.

APPENDIX A RECOMMENDATION PERFORMANCE RECOMMENDATION PERFORMANCE

Detailed Recommendation Performance

The recommendation performance in $PRS@3$, as calculated in Equation 5.1 of the thesis for all test cases are detailed in this section, where the "Training City" refers to the city whose data was utilized during the *Model Training* phase, and the "Test City" indicates the city whose data was used for the *Recommendation Generation* phase of the framework. In Tables A.1, A.3, and A.5, the lack of data availability has limited the experiments to only the restaurants and other eating places category. The recommendation performances by training cities are given in Tables A.1, A.3, A.4, A.5, A.6, A.7, A.8, A.9, A.10, A.11, A.12, and A.13 below.

Table A.1 PRS@3 for training city Atlanta and business category restaurants and other eating places

Test City	**Restaurants**		**Grocery Stores**	
	DQL Framework	Huff Gravity Model	DQL Framework	Huff Gravity Model
Austin	0.6723	0.6684	-	-
Boston	0.6149	0.4391	-	-
Chicago	0.9064	0.7269	-	-
Cleveland	0.7717	0.3669	-	-
Dallas	0.8395	0.8462	-	-
Houston	0.9301	0.5613	-	-
Los Angeles	0.9069	0.6131	-	-
Manhattan	0.9076	0.2603	-	-
Philadelphia	0.8517	0.7080	-	-
Phoenix	0.8698	0.2970	-	-
Sacramento	0.7891	0.7886	-	-
Tampa	0.7304	0.8188	-	-

Table A.2 PRS@3 for training city Austin

Test City	**Restaurants**		**Grocery Stores**	
	DQL Framework	Huff Gravity Model	DQL Framework	Huff Gravity Model
Atlanta	0.7222	0.5575	-	-
Boston	0.6788	0.6802	-	-
Chicago	0.8816	0.7312	0.7967	0.5414
Cleveland	0.6691	0.6017	-	-
Dallas	0.8064	0.8077	0.7551	0.5886
Houston	0.8684	0.5256	0.7843	0.6011
Los Angeles	0.8346	0.9097	0.8168	0.7071
Manhattan	0.8591	0.6328	0.8975	0.6793
Philadelphia	0.7657	0.8833	0.6213	0.7566
Phoenix	0.9252	0.2878	0.6753	0.4881
Sacramento	0.7377	0.7886	0.7531	0.4381
Tampa	0.6951	0.8188	0.6622	0.8557

Table A.3 PRS@3 for training city Boston and business category restaurants and other eating places

Test City	**Restaurants**		**Grocery Stores**	
	DQL Framework	Huff Gravity Model	DQL Framework	Huff Gravity Model
Atlanta	0.8441	0.7758	-	-
Austin	0.7873	0.7675	-	-
Chicago	0.8911	0.8502	-	-
Cleveland	0.7207	0.3375	-	-
Dallas	0.7784	0.8462	-	-
Houston	0.9150	0.6157	-	-
Los Angeles	0.8376	0.8961	-	-
Manhattan	0.8946	0.2603	-	-
Philadelphia	0.9006	0.9632	-	-
Phoenix	0.8893	0.3596	-	-
Sacramento	0.7245	0.7581	-	-
Tampa	0.8345	0.8188	-	-

Table A.4 PRS@3 for training city Chicago

Test City	**Restaurants**		**Grocery Stores**	
	DQL Framework	Huff Gravity Model	DQL Framework	Huff Gravity Model
Atlanta	0.8757	0.7168	-	-
Austin	0.7150	0.6615	0.7002	0.6761
Boston	0.7399	0.6168	-	-
Cleveland	0.7532	0.5241	-	-
Dallas	0.8867	0.8415	0.7589	0.5676
Houston	0.8858	0.5182	0.8096	0.5228
Los Angeles	0.9069	0.8961	0.7038	0.6147
Manhattan	0.9322	0.5655	0.7628	0.5778
Philadelphia	0.8796	0.8833	0.7915	0.7252
Phoenix	0.9057	0.2878	0.7364	0.5886
Sacramento	0.9266	0.7581	0.6660	0.5581
Tampa	0.8778	0.7773	0.6239	0.6682

Table A.5 PRS@3 for training city Cleveland and business category restaurants and Other eating places

Test City	**Restaurants**		**Grocery Stores**	
	DQL Framework	Huff Gravity Model	DQL Framework	Huff Gravity Model
Atlanta	0.7056	0.6519	-	-
Austin	0.7553	0.7675	-	-
Boston	0.5344	0.5431	-	-
Chicago	0.8481	0.8257	-	-
Dallas	0.7874	0.8462	-	-
Houston	0.9295	0.5208	-	-
Los Angeles	0.8914	0.5102	-	-
Manhattan	0.8755	0.2603	-	-
Philadelphia	0.7631	0.7080	-	-
Phoenix	0.8466	0.3170	-	-
Sacramento	0.8299	0.7505	-	-
Tampa	0.7602	0.8557	-	-

Table A.6 PRS@3 for training city Dallas

Test City	**Restaurants**		**Grocery Stores**	
	DQL Framework	Huff Gravity Model	DQL Framework	Huff Gravity Model
Atlanta	0.7115	0.5428	-	-
Austin	0.7934	0.7983	0.5414	0.6761
Boston	0.6553	0.7234	-	-
Chicago	0.9265	0.5217	0.7119	0.5378
Cleveland	0.7838	0.3669	-	-
Houston	0.9073	0.5427	0.6451	0.6011
Los Angeles	0.8351	0.4570	0.7607	0.7071
Manhattan	0.8921	0.2603	0.5973	0.5391
Philadelphia	0.8149	0.7080	0.8422	0.7566
Phoenix	0.9042	0.5032	0.8101	0.3898
Sacramento	0.9056	0.3638	0.6666	0.5229
Tampa	0.8492	0.6402	0.5848	0.8557

Table A.7 PRS@3 for training city Houston

Test City	**Restaurants**		**Grocery Stores**	
	DQL Framework	Huff Gravity Model	DQL Framework	Huff Gravity Model
Atlanta	0.6892	0.8348	-	-
Austin	0.6746	0.7983	0.7843	0.6011
Boston	0.7957	0.4391	-	-
Chicago	0.8939	0.8257	0.8096	0.5228
Cleveland	0.6070	0.3669	-	-
Dallas	0.7053	0.8462	0.6451	0.6011
Los Angeles	0.8883	0.6346	0.6758	0.6011
Manhattan	0.9059	0.2603	0.8327	0.3792
Philadelphia	0.9295	0.9114	0.7075	0.6011
Phoenix	0.8274	0.5032	0.7968	0.6011
Sacramento	0.7413	0.7505	0.8298	0.6011
Tampa	0.8039	0.8188	0.8309	0.5359

Table A.8 PRS@3 for training city Los Angeles

Test City	**Restaurants**		**Grocery Stores**	
	DQL Framework	Huff Gravity Model	DQL Framework	Huff Gravity Model
Atlanta	0.7169	0.4926	-	-
Austin	0.7964	0.7282	0.8168	0.7071
Boston	0.6563	0.7335	-	-
Chicago	0.8445	0.6604	0.7038	0.6147
Cleveland	0.7452	0.5157	-	-
Dallas	0.7843	0.8019	0.7607	0.7071
Houston	0.8808	0.5772	0.6758	0.7207
Manhattan	0.8989	0.2603	0.7437	0.5339
Philadelphia	0.8481	0.7080	0.6738	0.6262
Phoenix	0.8269	0.3596	0.8091	0.5596
Sacramento	0.8137	0.6743	0.6916	0.4945
Tampa	0.7797	0.5735	0.6552	0.6404

Table A.9 PRS@3 for training city Manhattan

Test City	**Restaurants**		**Grocery Stores**	
	DQL Framework	Huff Gravity Model	DQL Framework	Huff Gravity Model
Atlanta	0.6530	0.6903	-	-
Austin	0.7200	0.5231	0.7693	0.6761
Boston	0.7374	0.7995	-	-
Chicago	0.8650	0.6359	0.7686	0.7115
Cleveland	0.6778	0.5367	-	-
Dallas	0.7384	0.8415	0.6983	0.7319
Houston	0.9369	0.7308	0.8327	0.3792
Los Angeles	0.9259	0.8961	0.7437	0.5339
Philadelphia	0.8788	0.8833	0.9059	0.5864
Phoenix	0.8112	0.3132	0.7837	0.6212
Sacramento	0.7979	0.6343	0.7495	0.5778
Tampa	0.8014	0.7196	0.6730	0.3153

Table A.10 PRS@3 for training city Philadelphia

Test City	**Restaurants**		**Grocery Stores**	
	DQL Framework	Huff Gravity Model	DQL Framework	Huff Gravity Model
Atlanta	0.7506	0.6431	-	-
Austin	0.7230	0.8103	0.6213	0.7566
Boston	0.7163	0.5203	-	-
Chicago	0.9222	0.8257	0.7915	0.7252
Cleveland	0.7677	0.3669	-	-
Dallas	0.7293	0.9138	0.8422	0.7566
Houston	0.9551	0.5208	0.8171	0.7566
Los Angeles	0.8575	0.6362	0.6115	0.7566
Manhattan	0.9100	0.2603	0.7335	0.7566
Phoenix	0.8916	0.6091	0.7081	0.7566
Sacramento	0.7324	0.6743	0.6220	0.7566
Tampa	0.7087	0.6474	0.7179	0.7566

Table A.11 PRS@3 for training city Phoenix

Test City	**Restaurants**		**Grocery Stores**	
	DQL Framework	Huff Gravity Model	DQL Framework	Huff Gravity Model
Atlanta	0.6386	0.7758	-	-
Austin	0.8050	0.7675	0.6753	0.4881
Boston	0.7022	0.6168	-	-
Chicago	0.9240	0.7926	0.7364	0.5886
Cleveland	0.7282	0.3375	-	-
Dallas	0.7761	0.8462	0.8101	0.3898
Houston	0.8739	0.5812	0.6279	0.4266
Los Angeles	0.8393	0.6346	0.8756	0.4266
Manhattan	0.9002	0.2603	0.7110	0.6793
Philadelphia	0.7945	0.7080	0.6942	0.3121
Sacramento	0.7988	0.7505	0.6590	0.5886
Tampa	0.8742	0.7773	0.7076	0.6425

Table A.12 PRS@3 for training city Sacramento

Test City	**Restaurants**		**Grocery Stores**	
	DQL Framework	Huff Gravity Model	DQL Framework	Huff Gravity Model
Atlanta	0.5677	0.6903	-	-
Austin	0.7548	0.6684	0.7007	0.6761
Boston	0.6916	0.6168	-	-
Chicago	0.8726	0.7398	0.8954	0.5378
Cleveland	0.7622	0.5367	-	-
Dallas	0.8570	0.8462	0.8558	0.5886
Houston	0.8786	0.6142	0.8298	0.6011
Los Angeles	0.9438	0.8961	0.6916	0.4945
Manhattan	0.9138	0.2618	0.7495	0.5778
Philadelphia	0.8155	0.9632	0.6220	0.7566
Phoenix	0.9055	0.3132	0.6590	0.3610
Tampa	0.8820	0.7773	0.4999	0.3962

Table A.13 PRS@3 for training city Tampa

Test City	**Restaurants**		**Grocery Stores**	
	DQL Framework	Huff Gravity Model	DQL Framework	Huff Gravity Model
Atlanta	0.7607	0.5605	-	-
Austin	0.6959	0.6684	0.6622	0.8557
Boston	0.6726	0.6168	-	-
Chicago	0.9220	0.7269	0.6239	0.6682
Cleveland	0.7248	0.5241	-	-
Dallas	0.8072	0.8462	0.5848	0.8557
Houston	0.8468	0.5256	0.6418	0.7295
Los Angeles	0.9453	0.9097	0.6340	0.8557
Manhattan	0.9694	0.2610	0.6854	0.5933
Philadelphia	0.9209	0.9632	0.8026	0.8557
Phoenix	0.8476	0.3548	0.8079	0.7295
Sacramento	0.8720	0.7886	0.8736	0.8142

Alternative Performance Metrics

Experimental results with respective to alternative performance metrics $PRS@1$, $PRS@2$, $PRS@4$, and $PRS@5$, as calculated in Equation 5.1 of the thesis, are detailed in in this section. The metrics are consistent, as increasing k increases the PRS by about 1.5%. The alternative metrics do not highlight any pattern not observable by the chosen performance metric, $PRS@3$, and are therefore excluded from the main chapters of this thesis. For training cities of Atlanta, Boston and Cleveland; the lack of available data has limited the experiments to only the restaurants and other eating places category. The alternative performance metrics by training cities are given in Tables A.14, A.16, A.17, A.18, A.19, A.20, A.21, A.22, A.23, A.24, A.25, and A.26 below.

Table A.14 Alternative performance metrics for training city Atlanta

Test City	**Restaurants**				**Grocery Stores**			
	PRS@1	PRS@2	PRS@4	PRS@5	PRS@1	PRS@2	PRS@4	PRS@5
Austin	0.7420	0.7013	0.6853	0.7183	-	-	-	-
Boston	0.3742	0.5730	0.6909	0.7414	-	-	-	-
Chicago	0.8587	0.9161	0.9278	0.9333	-	-	-	-
Cleveland	0.7907	0.7277	0.7498	0.7828	-	-	-	-
Dallas	0.9313	0.8175	0.8015	0.8464	-	-	-	-
Houston	0.8696	0.9230	0.9256	0.9480	-	-	-	-
Los Angeles	0.8112	0.8912	0.8872	0.9052	-	-	-	-
Manhattan	0.8538	0.8746	0.8962	0.8912	-	-	-	-
Philadelphia	0.7379	0.8159	0.7986	0.8125	-	-	-	-
Phoenix	0.7951	0.8506	0.8655	0.8650	-	-	-	-
Sacramento	0.5945	0.6889	0.7954	0.8141	-	-	-	-
Tampa	0.5945	0.6761	0.8028	0.7667	-	-	-	-

Table A.15 Alternative performance metrics for training city Austin

Test City	Restaurants				Grocery Stores			
	PRS@1	PRS@2	PRS@4	PRS@5	PRS@1	PRS@2	PRS@4	PRS@5
Atlanta	0.4921	0.6766	0.7068	0.6642	-	-	-	-
Boston	0.7730	0.7677	0.7409	0.6965	-	-	-	-
Chicago	0.8129	0.8495	0.8964	0.8963	0.7699	0.7761	0.7682	0.8105
Cleveland	0.4394	0.5484	0.6471	0.7005	-	-	-	-
Dallas	0.7267	0.8471	0.8233	0.8579	0.5808	0.6684	0.7638	0.8268
Houston	0.7798	0.8118	0.8868	0.9031	0.7902	0.8119	0.7806	0.7751
Los Angeles	0.7938	0.8382	0.8529	0.8499	0.8456	0.8265	0.7624	0.7391
Manhattan	0.8444	0.8543	0.8627	0.8668	0.7714	0.8316	0.8699	0.8256
Philadelphia	0.7086	0.7410	0.8124	0.8203	0.5919	0.6247	0.6527	0.6325
Phoenix	0.8840	0.9479	0.9067	0.9178	0.6160	0.5561	0.6229	0.6568
Sacramento	0.8114	0.8134	0.7221	0.7431	0.6272	0.7031	0.7122	0.7529
Tampa	0.7135	0.7594	0.7060	0.7133	0.8510	0.7712	0.7400	0.7779

Table A.16 Alternative performance metrics for training city Boston

Test City	Restaurants				Grocery Stores			
	PRS@1	PRS@2	PRS@4	PRS@5	PRS@1	PRS@2	PRS@4	PRS@5
Atlanta	0.7838	0.8811	0.8478	0.8705	-	-	-	-
Austin	0.8527	0.8273	0.7863	0.7623	-	-	-	-
Chicago	0.9508	0.9338	0.9135	0.8992	-	-	-	-
Cleveland	0.5960	0.7143	0.7003	0.7586	-	-	-	-
Dallas	0.6813	0.8456	0.8141	0.8206	-	-	-	-
Houston	0.8157	0.8965	0.9012	0.9125	-	-	-	-
Los Angeles	0.8310	0.8302	0.8660	0.8763	-	-	-	-
Manhattan	0.8460	0.8451	0.9020	0.9236	-	-	-	-
Philadelphia	0.9047	0.8728	0.9265	0.9362	-	-	-	-
Phoenix	0.8961	0.9080	0.8520	0.8697	-	-	-	-
Sacramento	0.6130	0.7205	0.7698	0.8052	-	-	-	-
Tampa	0.9485	0.8649	0.8230	0.8208	-	-	-	-

Table A.17 Alternative performance metrics for training city Chicago

Test City	Restaurants				Grocery Stores			
	PRS@1	PRS@2	PRS@4	PRS@5	PRS@1	PRS@2	PRS@4	PRS@5
Atlanta	0.7478	0.8060	0.8433	0.8342	-	-	-	-
Austin	0.7127	0.7643	0.7377	0.7619	0.6316	0.7843	0.7474	0.6951
Boston	0.5692	0.6643	0.7693	0.7905	-	-	-	-
Cleveland	0.7293	0.7008	0.7580	0.7626	-	-	-	-
Dallas	0.8074	0.8877	0.8655	0.7934	0.7924	0.7775	0.8030	0.8311
Houston	0.8687	0.8569	0.8833	0.8709	0.7028	0.7682	0.7908	0.7448
Los Angeles	0.8794	0.8813	0.9262	0.9040	0.7541	0.7806	0.7278	0.7757
Manhattan	0.9584	0.9114	0.9453	0.9432	0.8054	0.8487	0.8312	0.8443
Philadelphia	0.9394	0.8575	0.9064	0.8968	0.8260	0.7429	0.7506	0.7813
Phoenix	0.7970	0.9041	0.8554	0.8759	0.8619	0.6791	0.7838	0.7496
Sacramento	0.9654	0.8918	0.9156	0.8643	0.5067	0.6419	0.7155	0.7547
Tampa	0.9872	0.8964	0.8447	0.8345	0.7036	0.6963	0.5802	0.6346

Table A.18 Alternative performance metrics for training city Cleveland

Test City	Restaurants				Grocery Stores			
	PRS@1	PRS@2	PRS@4	PRS@5	PRS@1	PRS@2	PRS@4	PRS@5
Atlanta	0.6480	0.6624	0.7272	0.7799	-	-	-	-
Austin	0.6824	0.7484	0.7807	0.7905	-	-	-	-
Boston	0.3838	0.4367	0.6350	0.6915	-	-	-	-
Chicago	0.9061	0.8514	0.8517	0.8819	-	-	-	-
Dallas	0.7121	0.7503	0.7801	0.7795	-	-	-	-
Houston	0.9173	0.9201	0.9307	0.9314	-	-	-	-
Los Angeles	0.9343	0.9203	0.8926	0.9013	-	-	-	-
Manhattan	0.8877	0.8796	0.8818	0.8959	-	-	-	-
Philadelphia	0.6378	0.7318	0.7909	0.8245	-	-	-	-
Phoenix	0.7665	0.8239	0.8621	0.8738	-	-	-	-
Sacramento	0.6382	0.7840	0.7754	0.7755	-	-	-	-
Tampa	0.7608	0.8076	0.7524	0.7684	-	-	-	-

Table A.19 Alternative performance metrics for training city Dallas

Test City	**Restaurants**				**Grocery Stores**			
	PRS@1	PRS@2	PRS@4	PRS@5	PRS@1	PRS@2	PRS@4	PRS@5
Atlanta	0.6139	0.5621	0.7186	0.7430	-	-	-	-
Austin	0.9900	0.8802	0.7970	0.8169	0.5445	0.5779	0.5362	0.5965
Boston	0.4454	0.5396	0.7048	0.6494	-	-	-	-
Chicago	0.8931	0.9463	0.9310	0.9083	0.6419	0.6855	0.7586	0.7261
Cleveland	0.7667	0.7453	0.7815	0.8220	-	-	-	-
Houston	0.9331	0.8889	0.8995	0.9212	0.5643	0.6271	0.6903	0.7039
Los Angeles	0.8083	0.7739	0.8721	0.8934	0.3693	0.6871	0.6861	0.7209
Manhattan	0.8704	0.8515	0.9152	0.8991	0.8025	0.6384	0.6700	0.6778
Philadelphia	0.8213	0.7610	0.8578	0.8530	0.6785	0.7932	0.8685	0.7854
Phoenix	0.8997	0.9018	0.8542	0.8526	0.9101	0.8857	0.7133	0.7529
Sacramento	0.7673	0.8608	0.8997	0.8898	0.5105	0.5246	0.6563	0.6659
Tampa	0.7055	0.8537	0.8231	0.8425	0.7143	0.5959	0.6675	0.7026

Table A.20 Alternative performance metrics for training city Houston

Test City	**Restaurants**				**Grocery Stores**			
	PRS@1	PRS@2	PRS@4	PRS@5	PRS@1	PRS@2	PRS@4	PRS@5
Atlanta	0.4727	0.6510	0.7544	0.7842	-	-	-	-
Austin	0.5994	0.6609	0.6894	0.7216	0.9472	0.7225	0.8262	0.7925
Boston	0.8440	0.7888	0.8061	0.8350	-	-	-	-
Chicago	0.8817	0.9380	0.8752	0.8912	0.5942	0.7424	0.7666	0.7637
Cleveland	0.4996	0.5487	0.6764	0.7230	-	-	-	-
Dallas	0.5806	0.7366	0.6843	0.7522	0.5518	0.6478	0.7357	0.7372
Los Angeles	0.9178	0.9060	0.8773	0.8697	0.8597	0.6573	0.7124	0.7783
Manhattan	0.8342	0.8621	0.9248	0.9140	0.7800	0.8189	0.8847	0.8107
Philadelphia	0.8587	0.9160	0.9224	0.9119	0.6461	0.7823	0.8447	0.8636
Phoenix	0.8496	0.8158	0.8466	0.8497	0.6779	0.5378	0.6035	0.6645
Sacramento	0.6165	0.7453	0.7158	0.7499	0.5125	0.5209	0.6022	0.6080
Tampa	0.7906	0.8191	0.7912	0.7573	0.6344	0.7683	0.7338	0.6860

Table A.21 Alternative performance metrics for training city Los Angeles

Test City	Restaurants				Grocery Stores			
	PRS@1	PRS@2	PRS@4	PRS@5	PRS@1	PRS@2	PRS@4	PRS@5
Atlanta	0.6170	0.6972	0.6802	0.7120	-	-	-	-
Austin	0.8929	0.8158	0.7449	0.7747	0.5717	0.6403	0.7282	0.6989
Boston	0.3896	0.5496	0.7226	0.6639	-	-	-	-
Chicago	0.8232	0.8323	0.8480	0.8418	0.9278	0.8544	0.8028	0.8306
Cleveland	0.7372	0.7426	0.6910	0.7483	-	-	-	-
Dallas	0.7516	0.8005	0.8030	0.8001	0.8989	0.8557	0.8501	0.8729
Houston	0.8768	0.9113	0.8813	0.9066	0.5854	0.6237	0.6803	0.7007
Manhattan	0.8996	0.8828	0.8963	0.8840	0.8239	0.7732	0.6979	0.7380
Philadelphia	0.8673	0.8452	0.8581	0.8532	0.5287	0.4590	0.6272	0.6058
Phoenix	0.8050	0.8412	0.8393	0.8406	0.8879	0.8696	0.8688	0.8845
Sacramento	0.8385	0.8667	0.7645	0.7808	0.4277	0.4576	0.5341	0.5977
Tampa	0.8375	0.7476	0.7653	0.7959	0.6973	0.7116	0.6198	0.6274

Table A.22 Alternative performance metrics for training city Manhattan

Test City	Restaurants				Grocery Stores			
	PRS@1	PRS@2	PRS@4	PRS@5	PRS@1	PRS@2	PRS@4	PRS@5
Atlanta	0.6611	0.7301	0.7142	0.6994	-	-	-	-
Austin	0.8070	0.7621	0.7525	0.7635	0.4694	0.6603	0.7372	0.7105
Boston	0.4773	0.6601	0.7575	0.7851	-	-	-	-
Chicago	0.7964	0.8565	0.8704	0.8871	0.8723	0.8546	0.8286	0.8531
Cleveland	0.6678	0.7243	0.6890	0.7085	-	-	-	-
Dallas	0.7797	0.8078	0.7995	0.8074	0.5805	0.7477	0.6903	0.7153
Houston	0.8981	0.9102	0.9164	0.9273	0.7685	0.7425	0.7959	0.7613
Los Angeles	0.8955	0.9233	0.9176	0.9266	0.5698	0.6795	0.7450	0.7685
Philadelphia	0.8113	0.8782	0.8538	0.8687	0.9218	0.7649	0.6776	0.7310
Phoenix	0.8711	0.7801	0.8689	0.8911	0.5397	0.7166	0.6454	0.7126
Sacramento	0.9026	0.8631	0.8094	0.8213	0.5592	0.6546	0.6527	0.6370
Tampa	0.8526	0.8017	0.8073	0.8152	0.9511	0.6987	0.7212	0.6762

Table A.23 Alternative performance metrics for training city Philadelphia

Test City	**Restaurants**				**Grocery Stores**			
	PRS@1	PRS@2	PRS@4	PRS@5	PRS@1	PRS@2	PRS@4	PRS@5
Atlanta	0.7066	0.6616	0.6833	0.7511	-	-	-	-
Austin	0.7747	0.7178	0.7510	0.8078	0.3903	0.6721	0.6789	0.7183
Boston	0.5727	0.6737	0.8021	0.7867	-	-	-	-
Chicago	0.9743	0.9739	0.9209	0.9055	0.8169	0.7900	0.7516	0.7480
Cleveland	0.7775	0.7207	0.7080	0.7599	-	-	-	-
Dallas	0.8628	0.7830	0.7354	0.7957	0.6527	0.7902	0.7277	0.7327
Houston	0.9730	0.9754	0.9637	0.9598	0.5850	0.6709	0.7550	0.7489
Los Angeles	0.7644	0.8677	0.8527	0.8794	0.7079	0.7206	0.7468	0.7721
Manhattan	0.8592	0.8773	0.8953	0.8835	0.9494	0.8353	0.8995	0.8566
Phoenix	0.8777	0.8922	0.8898	0.9059	0.7073	0.7608	0.7277	0.7774
Sacramento	0.7710	0.7787	0.7769	0.7947	0.5905	0.5623	0.6055	0.6011
Tampa	0.5925	0.6923	0.7313	0.7433	0.8538	0.7124	0.8394	0.8420

Table A.24 Alternative performance metrics for training city Phoenix

Test City	**Restaurants**				**Grocery Stores**			
	PRS@1	PRS@2	PRS@4	PRS@5	PRS@1	PRS@2	PRS@4	PRS@5
Atlanta	0.7035	0.6965	0.7257	0.7110	-	-	-	-
Austin	0.8236	0.8061	0.8184	0.8011	0.5717	0.7851	0.7301	0.7044
Boston	0.5520	0.7225	0.7678	0.7845	-	-	-	-
Chicago	0.8858	0.8863	0.9393	0.9467	0.9278	0.8496	0.7493	0.7596
Cleveland	0.6843	0.7178	0.7017	0.7059	-	-	-	-
Dallas	0.7733	0.8049	0.7938	0.7801	0.8989	0.7636	0.7470	0.7332
Houston	0.8203	0.8881	0.8855	0.9058	0.5854	0.6882	0.7217	0.6610
Los Angeles	0.8115	0.8882	0.8340	0.8657	0.6731	0.7621	0.8258	0.8189
Manhattan	0.8941	0.9334	0.9118	0.9105	0.8239	0.8046	0.7709	0.7603
Philadelphia	0.7940	0.8396	0.8040	0.8251	0.5287	0.5659	0.7021	0.7711
Sacramento	0.6833	0.7078	0.7775	0.7861	0.4277	0.5358	0.5580	0.5725
Tampa	0.6973	0.8057	0.8550	0.8508	0.6973	0.7869	0.7610	0.7266

Table A.25 Alternative performance metrics for training city Sacramento

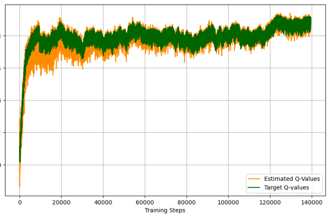
Test City	**Restaurants**				**Grocery Stores**			
	PRS@1	PRS@2	PRS@4	PRS@5	PRS@1	PRS@2	PRS@4	PRS@5
Atlanta	0.4639	0.5593	0.5984	0.6664	-	-	-	-
Austin	0.8218	0.7641	0.7464	0.7412	0.5144	0.5721	0.6925	0.7003
Boston	0.7532	0.7860	0.6289	0.6146	-	-	-	-
Chicago	1.0000	0.8870	0.8812	0.9049	0.9643	0.9595	0.9244	0.9371
Cleveland	0.6869	0.6539	0.8637	0.8683	-	-	-	-
Dallas	0.7458	0.8463	0.8264	0.8157	0.7911	0.9243	0.8562	0.8430
Houston	0.8103	0.8611	0.8969	0.8856	0.9242	0.8258	0.8028	0.7789
Los Angeles	0.9125	0.9509	0.9467	0.9504	0.6423	0.6281	0.7109	0.7199
Manhattan	0.9435	0.9448	0.9099	0.9308	0.7442	0.6738	0.7754	0.7749
Philadelphia	0.7316	0.8197	0.7754	0.8181	0.4265	0.4785	0.6822	0.6547
Phoenix	0.9710	0.9510	0.8806	0.8817	0.7024	0.7251	0.7174	0.6934
Tampa	0.8500	0.8887	0.8573	0.8847	0.7160	0.8097	0.8408	0.8084

Table A.26 Alternative performance metrics for training city Tampa

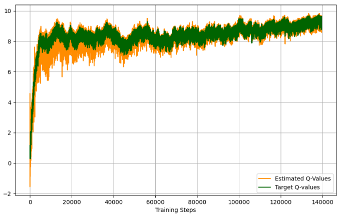
Test City	**Restaurants**				**Grocery Stores**			
	PRS@1	PRS@2	PRS@4	PRS@5	PRS@1	PRS@2	PRS@4	PRS@5
Atlanta	0.6046	0.7509	0.7675	0.7230	-	-	-	-
Austin	0.6859	0.6696	0.6980	0.7259	0.6813	0.6058	0.6678	0.7061
Boston	0.9002	0.7038	0.7131	0.7109	-	-	-	-
Chicago	0.7731	0.8850	0.8930	0.9149	0.6725	0.7334	0.6823	0.6849
Cleveland	0.5498	0.6719	0.7342	0.7454	-	-	-	-
Dallas	0.8432	0.7962	0.7665	0.7589	0.7125	0.7367	0.7429	0.7528
Houston	0.8165	0.8234	0.8749	0.8705	0.9754	0.9586	0.8479	0.8341
Los Angeles	1.0000	0.9716	0.9240	0.8912	0.5089	0.4850	0.7328	0.7102
Manhattan	0.9403	0.9722	0.9770	0.9501	0.6983	0.7290	0.7232	0.7376
Philadelphia	0.8265	0.9002	0.9299	0.8869	0.6713	0.5940	0.6988	0.7092
Phoenix	0.8833	0.8606	0.8521	0.8722	0.6370	0.5659	0.7379	0.6870
Sacramento	0.6309	0.7949	0.8385	0.8699	0.4255	0.4358	0.5331	0.5406

APPENDIX B TRAINING STABILITY TRAINING STABILITY

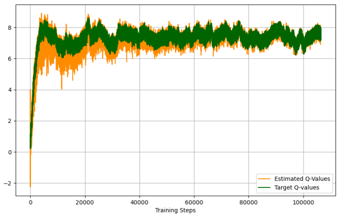
To ensure the stability of training, particularly given the novelty of the work and the vulnerability of RL algorithms to divergence, the Q-value updates are analyzed and visualized in Figures B.1 and B.2. It is observed that the proposed DQL framework does not suffer from divergence or overestimation of Q-values.



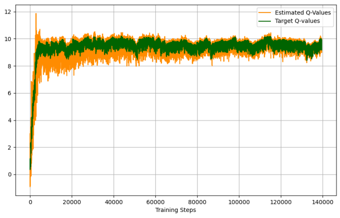
(a) Atlanta



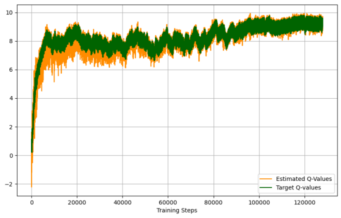
(b) Austin



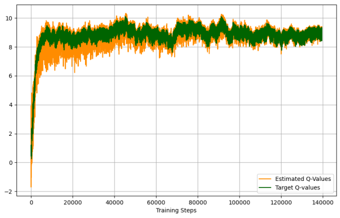
(c) Boston



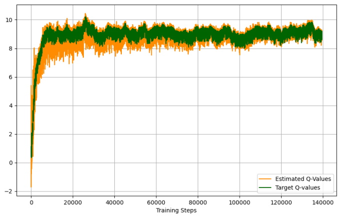
(d) Chicago



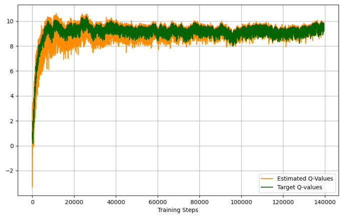
(e) Cleveland



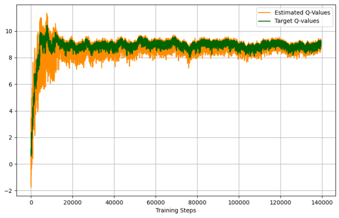
(f) Dallas



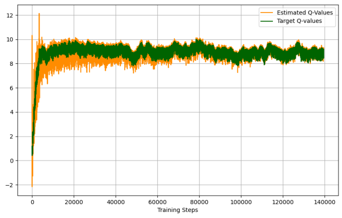
(g) Houston



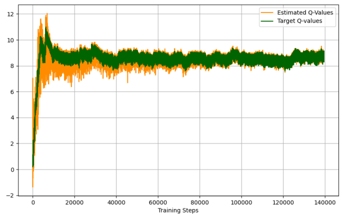
(h) Los Angeles



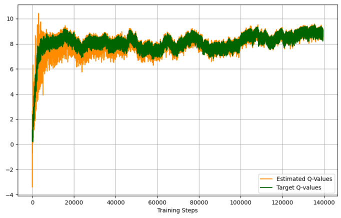
(i) Manhattan



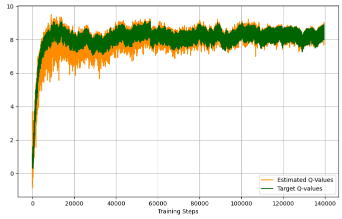
(j) Philadelphia



(k) Phoenix

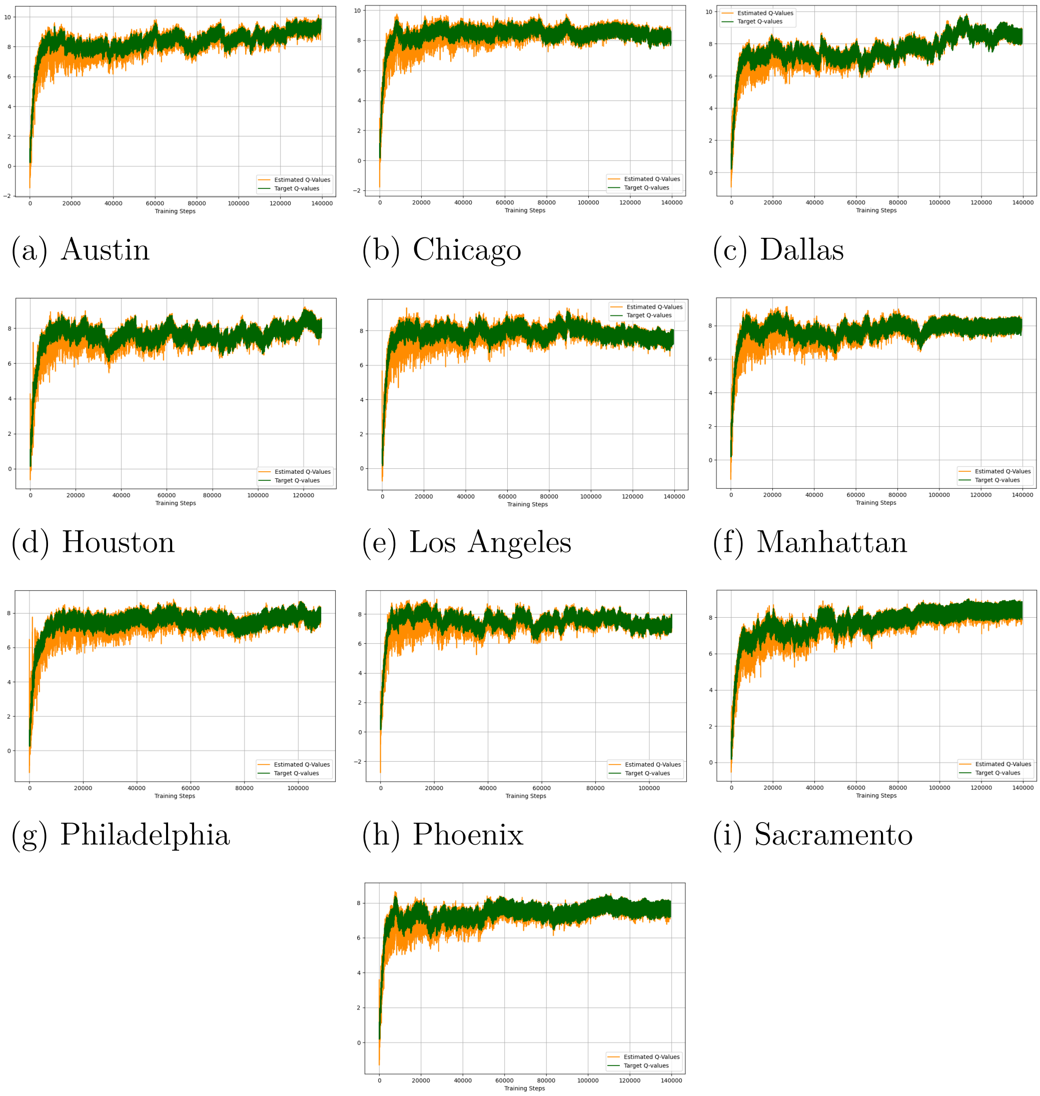


(l) Sacramento



(m) Tampa

Figure B.1 Estimated and target Q-values throughout model training for business category restaurants and other eating places



(a) Austin (b) Chicago (c) Dallas

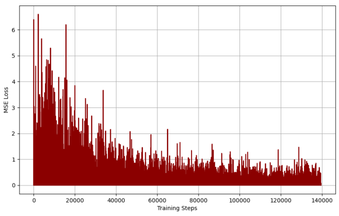
(d) Houston (e) Los Angeles (f) Manhattan

(g) Philadelphia (h) Phoenix (i) Sacramento

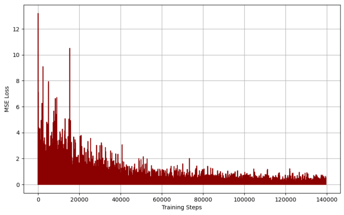
(j) Tampa

Figure B.2 Estimated and target Q-values throughout model training for business category grocery stores

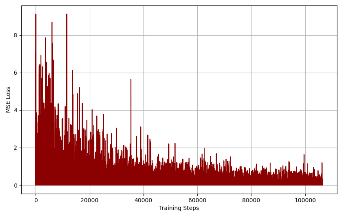
In addition to monitoring the Q-values, the training loss is also observed to ensure stability. As illustrated in Figures B.3 and B.4, no divergence was detected during the training in any of the cities or business categories.



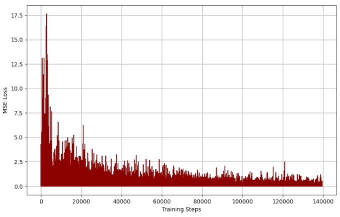
(a) Atlanta



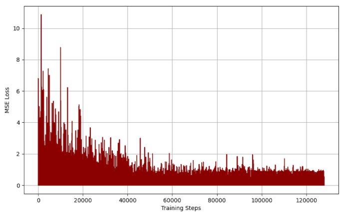
(b) Austin



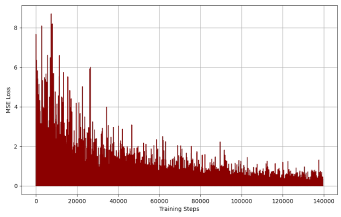
(c) Boston



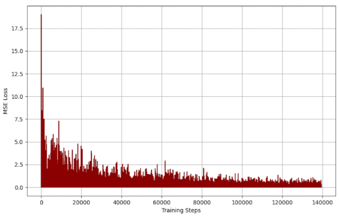
(d) Chicago



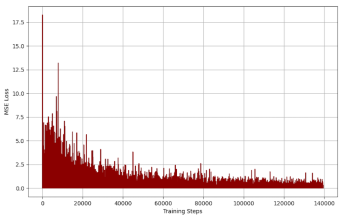
(e) Cleveland



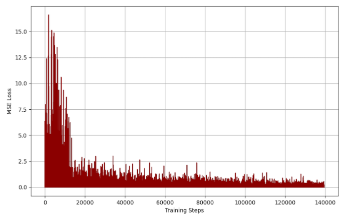
(f) Dallas



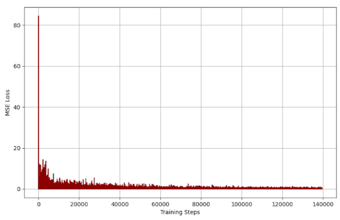
(g) Houston



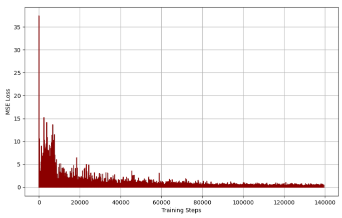
(h) Los Angeles



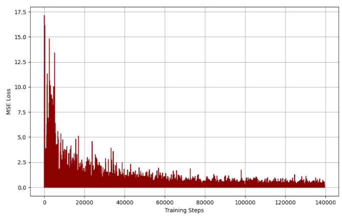
(i) Manhattan



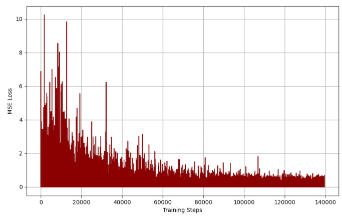
(j) Philadelphia



(k) Phoenix

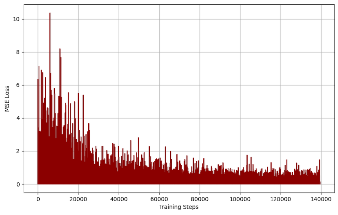


(l) Sacramento

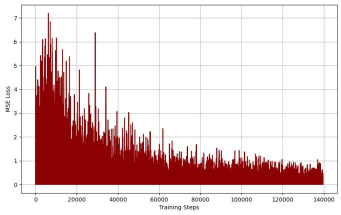


(m) Tampa

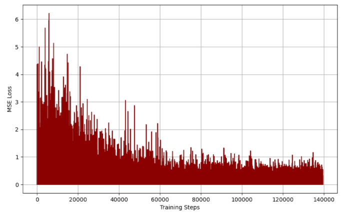
Figure B.3 Training loss for business category restaurants and other eating places



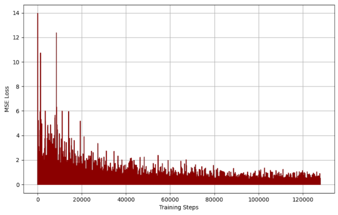
(a) Austin



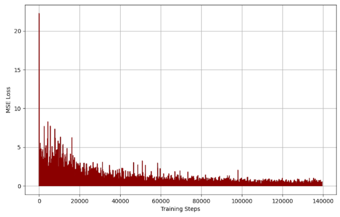
(b) Chicago



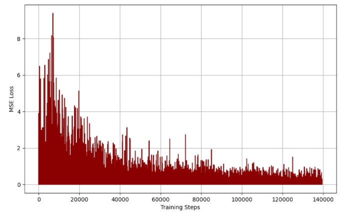
(c) Dallas



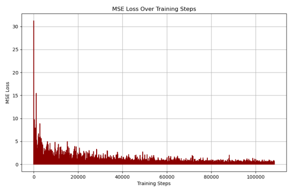
(d) Houston



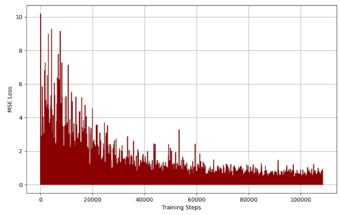
(e) Los Angeles



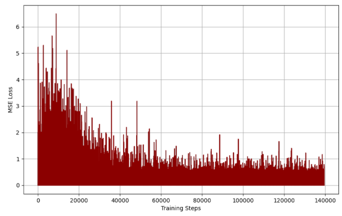
(f) Manhattan



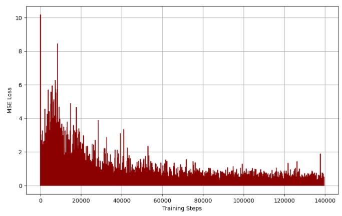
(g) Philadelphia



(h) Phoenix



(i) Sacramento



(j) Tampa

Figure B.4 Training loss for business category grocery stores