Proceedings of the 2025 Winter Simulation Conference
E. Azar, A. Djanatliev, A. Harper, C. Kogler, V. Ramamohan, A. Anagnostou, and S. J. E. Taylor, eds.

REVOLUTIONIZING ORDER FULFILLMENT: A NEURAL NETWORK APPROACH TO OPTIMAL WAREHOUSE SELECTION IN SUPPLY CHAIN SIMULATION

Weilong Wang[1], Michael Bloem[2], Jinxiang Gu[2], and Yunan Liu[1]

[1]Amazon.com, New York, NY, USA
[2]Amazon.com, Bellevue, WA, USA

ABSTRACT

Simulation plays a central role in the strategic planning and operational evaluation of supply chain networks. Within these networks, order fulfillment traditionally requires solving computationally expensive optimization problems in real-time across multiple constraints. For forward-looking simulations evaluating millions of orders, such optimization becomes prohibitively expensive. We develop a neural network-based emulator that approximates optimal fulfillment decisions while maintaining millisecond-level inference speed. Operating at ZIP-code level resolution and incorporating shipping speed constraints, our model handles exponential decision spaces and non-stationary patterns. Empirical results demonstrate 56.75% order-level accuracy, a 20 percentage point improvement over benchmarks. Through novel regularization balancing order-level and network-level efficiency, we achieve 47.13% node-level accuracy while maintaining 50.31% order-level accuracy. Our model captures intricate patterns in historical fulfillment data, enabling efficient forward-looking simulation for strategic planning.

1 INTRODUCTION

The largest modern supply chains span hundreds of fulfillment centers (FCs), millions of unique products, and thousands of zip codes, requiring rapid and accurate decisions about where to fulfill each customer order. These decisions involve balancing factors such as customer delivery expectations, regional inventory positions, node-level capacity, transportation costs, and supplier constraints (Velásquez-Bermúdez, Khakifirooz, and Fathi 2019). For instance, fulfilling a fast-shipping electronics order from a nearby vendor warehouse may reduce transportation time, but may come with higher costs or lower availability compared to a centralized fulfillment center.

In real-world systems such as those operated by Amazon, Alibaba, and Walmart, fulfillment path selection is often powered by large-scale combinatorial optimization engines. These engines evaluate fulfillment paths based on inventory placement, shipping speed options, geographic routing, real-time availability, and capacity constraints. While these optimization engines work well for real-time decisions, incorporating them into simulation environments presents major challenges. Simulation is increasingly relied upon to evaluate strategic changes, such as new shipping programs, regional inventory policies, or node expansion plans. These simulations need to evaluate millions of potential orders across multiple months of future scenarios. Running combinatorial solvers or mixed-integer linear program (MILP) based optimizers for each simulated order across millions of orders is cost-prohibitive (Dolara, Grimaccia, Magistrati, and Marchegiani 2017), with processing times potentially extending to weeks or months for comprehensive scenario analysis.

As a result, emulators such as statistical approximations to replicate the full behavior of a complex system are often used. These models can be trained on historical data to emulate production decisions. Yet, the challenge lies in achieving sufficient fidelity: failing to capture the subtleties of production logic (e.g., regional routing rules, handling preferences, or vendor-specific constraints) can result in simulations that lead to inaccurate downstream conclusions. Prior literature highlights these limitations. (Acimovic and Graves 2015) explore fulfillment decision modeling through MILPs but note the scale challenges for simulation. (Farias, Gijsbrechts, Khojandi, Peng, and Zheng 2024) demonstrate the high computational cost of policy simulation in supply chain reinforcement learning and propose methods like Picard iteration to accelerate inference.

This paper builds on these insights and presents a high-fidelity neural network emulator that approximates production fulfillment behavior at fine geographic granularity. By using structured data and real-world fulfillment logs, we construct a scalable and accurate emulator for simulation-driven experimentation.

1.1 Challenges of Simulating Fulfillment

Simulating fulfillment path selection with high fidelity presents several unique challenges that stem from the complex, dynamic nature of modern e-commerce operations. These challenges are particularly acute in large-scale fulfillment networks where millions of real-time decisions must be made daily.

First, the decision space in fulfillment path selection grows exponentially with order complexity. In a network with hundreds of fulfillment centers, even a single-item order might have dozens of viable fulfillment options. This complexity multiplies dramatically for multi-item orders when considering cross-warehouse sourcing and split shipments. For instance, a five-item order might have thousands of possible fulfillment combinations, making exhaustive evaluation computationally infeasible for real-time decisions.

Second, production fulfillment systems rely heavily on real-time operational data that is difficult to replicate in simulation environments and fulfillment decisions exhibit significant non-stationarity. These include instantaneous inventory positions, processing backlogs, labor availability, and temporary facility constraints. When a fulfillment center experiences unexpected issues like weather disruptions or mechanical failures, production systems dynamically re-route orders - a level of adaptiveness that's challenging to capture in static simulations.

Last but not least, simulation of fulfillment needs to be computationally efficient since our simulation is forward-looking for the next several months. This requirement for computational speed presents a fundamental trade-off between simulation accuracy and processing time. While production fulfillment services provide highly accurate, real-time decisions, they cannot be practically used for forward-looking simulation for several critical reasons. Together, these challenges underscore the difficulty in designing surrogate models that can generalize production behavior in a way that is both scalable and realistic.

1.2 Review of Literature

Emulation of complex systems has received significant attention across simulation, operations research, and machine learning communities. Gaussian process emulators have been explored in industrial applications due to their uncertainty quantification properties (Houlsby et al. 2012), though scalability remains a bottleneck. Matrix factorization and latent class models (Blei et al. 2003) have also been applied to preference learning in logistics and recommendation tasks. FC selection is inherently a multi-class classification problem, where each FC represents a distinct class. While GPs can be extended to multi-class settings, they typically require multiple binary classifiers or complex covariance functions. This becomes unwieldy when dealing with hundreds of potential FCs and millions of orders, leading to both computational and modeling challenges.

Recent reinforcement learning methods for supply chain optimization highlight the potential of deep learning for decision-making but often rely on low-fidelity simulations during training. Our research extends and complements previous works by introducing a more sophisticated approach to outbound FC selection. Previous research has explored various approaches to order management and fulfillment optimization. A Deep Q-Network (DQN) approach was developed for single-vehicle pickup and delivery order acceptance decisions (Kang 2018). The order dispatch problem was also studied by focusing exclusively on acceptance/rejection decisions while implementing a greedy assignment rule (Kavuk et al. 2022). In the multi-agent domain, Kim et al. (2010) proposed a Multi-Agent Reinforcement Learning (MARL) model for multi-stage supply chain optimization. Their model featured retailer agents utilizing linear regression for downstream demand prediction, incorporating these predictions to establish safety lead times. Orders were triggered when predicted inventory levels dropped below threshold within these safety periods. Unlike previous methods that often simplified the FC selection process, our approach maintains high fidelity to real-world operational patterns while remaining computationally efficient. This enables the creation of more realistic simulation environments that better reflect actual supply chain dynamics.

1.3 Summary of Our Contributions

Our work situates itself within the surrogate modeling literature, but with a focus on increasing the fidelity of the emulator to a level that captures production-relevant behavior while maintaining simulation performance.

- **High-fidelity surrogate modeling:** We developed a feed-forward neural network model as a surrogate emulator that significantly improves fidelity to production-level fulfillment behavior while preserving the efficiency necessary for large-scale simulations. The model estimates a probability distribution over fulfillment nodes for each order by learning from historical decisions, effectively approximating production logic. By maximizing the likelihood of past source node selections, the model offers a statistically grounded emulation of operational behavior, enabling more accurate evaluation of network design, capacity planning, and policy interventions.
- **Rich feature integration:** Our model leverages a comprehensive set of inputs, including item-level attributes, candidate fulfillment node features, and contextual metadata such as location and shipping speed. Operating at a fine-grained geographic resolution—specifically at the zip code level—enables the model to capture subtle spatial patterns in fulfillment decisions, such as last-mile delivery costs and regional preferences. We explicitly incorporate shipping speed categories to address delivery promise constraints and include regional alignment indicators to promote in-region fulfillment. These features jointly ensure that the model aligns with both customer expectations and operational realities.
- **Robust generalization and practical relevance:** The architecture demonstrates strong generalization capabilities across a wide array of fulfillment scenarios, including different item types ranging from standard packages to those requiring special handling, and across diverse geographic contexts. This robustness allows the model to be effectively deployed in dynamic and heterogeneous real-world networks. By closely mirroring actual decision processes, the model bridges the gap between abstract theoretical frameworks and the complex realities of operational fulfillment, thereby supporting more informed strategic and tactical decisions.
- **Hybrid feature engineering:** We employ a hybrid approach to feature engineering that combines traditional distance-based metrics with learned representations of fulfillment center capabilities and historical performance. This integration enables the model to capture both explicit business rules—such as proximity-based routing—and implicit operational patterns that emerge over time from production data. The result is a model that achieves a balance between computational efficiency and predictive accuracy, making it well-suited for high-throughput simulation environments. This hybrid design enhances the model's capacity to support scalable, data-driven decision-making in large, complex fulfillment networks.

2 MODEL AND METHODOLOGY

2.1 Supply Chain Network

Supply chain is a complex network of inbound facilities, FCs, sortation centers, and delivery stations designed to efficiently move products from vendors to customers. Simulating complex supply chain networks presents significant challenges due to the high-dimensional nature of the problem and the intricate interdependencies between variables. A typical large-scale supply chain can be represented as a directed graph $G = (V, E)$, where V represents nodes (e.g., fulfillment centers, vendors) and E represents edges (transportation routes). The state of this system at any time t can be described by a high-dimensional vector $s_t \in \mathbb{R}^n$, where n can easily reach millions of dimensions when considering individual SKU-level inventory across thousands of nodes. The evolution of this system over time can be modeled as a function f:

$$s_{t+1} = f(s_t, a_t, \varepsilon_t) \tag{1}$$

where a_t represents actions (e.g., ordering decisions, fulfillment assignments) and ε_t represents stochastic elements (e.g., demand fluctuations, transportation delays). The inbound process is critical, involving the reception and integration of inventory into the fulfillment network. This process begins with vendors shipping products to FCs. Upon arrival, items undergo rigorous quality checks and are logged into the inventory management system. The inbound team then determines optimal storage locations within the facility, considering factors such as item characteristics, expected demand, and current inventory levels.

When a customer places an order, it triggers a sophisticated decision-making process to determine the optimal fulfillment path, considering factors such as inventory location, shipping promises, and network capacity constraints. Efficient outbound simulations are crucial for maintaining stock availability and supporting outbound fulfillment. End-to-end simulation of this network is crucial for several reasons. First, it enables proactive capacity management across different node types. By simulating expected order volumes and their distribution across the network, planners can identify potential bottlenecks, determine optimal inventory placement, and make informed decisions about network expansion or modification. This is particularly critical during peak seasons like Prime Day or holiday periods when order volumes can surge dramatically. From a labor planning perspective, accurate simulation helps optimize workforce management at each facility. This includes determining appropriate staffing levels for different shifts, planning seasonal hiring needs, and optimizing training programs. The simulation must account for various processing activities within FCs, from receiving and stowing to picking, packing, and shipping, each requiring different skill sets and staffing levels. Furthermore, end-to-end simulation supports strategic decision-making about network design and operational policies. For example, when evaluating the impact of new shipping speed offerings or considering the placement of new fulfillment centers, simulation can provide insights into the operational feasibility and resource requirements of different scenarios.

The fulfillment side is particularly complex due to its direct impact on customer experience. Accurate simulation must account for various constraints including physical capacity limitations of each facility, labor productivity rates for different processes, inventory placement and availability, transportation network constraints, delivery promise times, and regional volume variations. By simulating these elements comprehensively, organizations can better anticipate and prepare for operational challenges while maintaining service levels and controlling costs. However, the computational complexity of simulating this system grows exponentially with the number of nodes and SKUs, making exact solutions intractable for large-scale networks. Traditional simulation methods often struggle with the curse of dimensionality and the need for real-time decision-making. Figure 1 illustrates how the fulfillment flow works in simulations.

2.2 Multilayer Perceptron

Our model is a feed-forward neural network, a *multilayer perceptron* (MLP). MLPs can approximate complex, non-linear functions with high accuracy, making them suitable for capturing the intricate relationships in supply chain dynamics. The feedforward structure of MLPs allows for fast inference times, crucial for real-time decision-making in large-scale operations. It aims to optimize source FC selection by evaluating and scoring potential FC candidates for item fulfillment. The first layer is the input layer, and its units take the values of the input features. The last layer is the output layer, and it has the same number of units as the number of FCs we need to select from.

Multilayer Perceptron presents several distinct advantages for FC selection compared to alternative machine learning approaches like XGBoost or other traditional models. First, MLPs excel at learning complex non-linear relationships between input features and target outputs, which is crucial for FC selection where

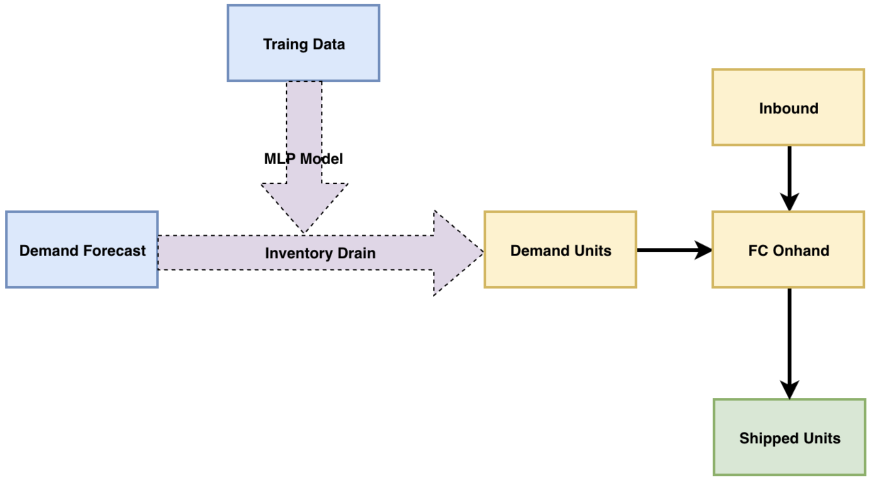

Figure 1: Fulfillment Flow.

multiple factors (geographic location, shipping speed, item characteristics) interact in subtle and complex ways to influence optimal fulfillment decisions.

The architecture of MLPs is particularly well-suited for handling the heterogeneous feature space in FC selection. The model can effectively process both continuous features (distances, coordinates, package dimensions) and categorical features (shipping speeds, FC types) through its hidden layers, learning hierarchical representations that capture both local and global patterns in the data. This is especially important when considering the spatial aspects of fulfillment decisions, where geographic relationships must be learned alongside operational constraints. Furthermore, MLPs offer superior scalability for real-time inference compared to ensemble methods like XGBoost. In production environments where millions of orders require immediate FC assignment, the fixed-size neural network architecture allows for efficient batch processing and parallelization. While XGBoost might achieve comparable accuracy, its tree-based structure can become computationally expensive for real-time predictions at scale. The probabilistic output of MLPs, achieved through the softmax activation in the final layer, naturally aligns with the FC selection task's requirements. This provides not just the optimal FC selection but also a probability distribution over all candidate FCs, which is valuable for simulation and analysis purposes. Traditional models might require additional post-processing to generate such probability distributions. Additionally, MLPs demonstrate better generalization to new scenarios and edge cases, which is crucial in dynamic fulfillment networks where conditions constantly evolve. The continuous nature of neural network representations, unlike the discrete splits in tree-based models, allows for smoother interpolation between seen examples and better handling of novel situations.

Our MLP architecture is designed specifically for the FC selection problem, with a fully connected structure where each unit receives connections from all units in the previous layer (See Figure 2). This dense connectivity ensures comprehensive feature interaction learning across all dimensions of the input space. The network consists of multiple layers, each playing a distinct role in the transformation of input features to FC selection probabilities. Mathematically, we can express the computations as follows: For an input vector $\mathbf{x} \in \mathbb{R}^d$, where d is the dimension of our feature space, the forward pass through the network is computed as:

$$h_i^1 = \phi^1(\sum_j \omega_{ij}^1 x_j + \varepsilon_i^1) \qquad \text{First hidden layer}$$
$$h_i^2 = \phi^2(\sum_j \omega_{ij}^2 h_j^1 + \varepsilon_i^2) \qquad \text{Second hidden layer}$$
$$y_i = \phi^3(\sum_j \omega_{ij}^3 h_j^2 + \varepsilon_i^3) \qquad \text{Output layer}$$

where:

- h_i^k represents the activation of the i-th neuron in the k-th hidden layer
- ω_{ij}^k denotes the weight connecting the j-th neuron in layer $k-1$ to the i-th neuron in layer k
- ε_i^k represents the bias term for the i-th neuron in layer k
- ϕ^k is the activation function for layer k

The activation functions are chosen specifically for each layer:

- ϕ^1: ReLU activation for the first hidden layer, enabling non-linear feature transformations while avoiding vanishing gradient problems
- ϕ^2: ReLU activation for the second hidden layer, maintaining non-linearity in deeper representations
- ϕ^3: Softmax activation for the output layer, producing a valid probability distribution over FC candidates

This architecture allows the model to learn hierarchical representations of the input features. The first hidden layer captures basic feature interactions. The second hidden layer learns higher-order patterns and relationships. The output layer transforms these representations into selection probabilities.

The model is trained end-to-end using backpropagation with the negative log-likelihood loss function, modified with our node-level regularization term. Our neural network emulator directly addresses challenges in Section 1.1 through its ability to learn complex non-linear patterns from historical data while maintaining millisecond-level inference speed. The model's structure enables it to capture the intricate relationships between order characteristics, network state, and fulfillment decisions without requiring explicit modeling of all operational constraints. By learning from actual production decisions, the MLP inherently incorporates the various heterogeneous factors that influence fulfillment choices, while its feedforward architecture ensures the computational efficiency needed for large-scale, forward-looking simulations.

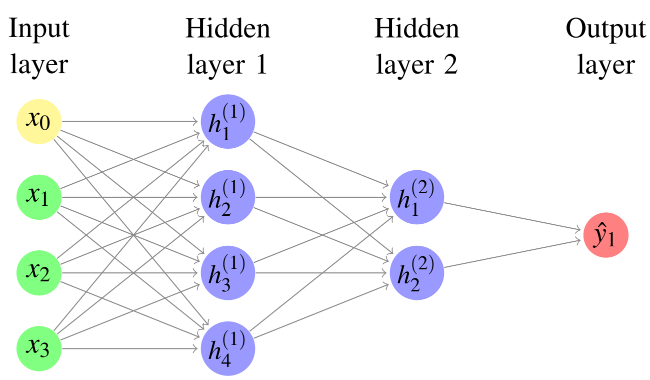

Figure 2: Multi-layer perceptron.

2.3 FC Selection Problem

FC selection problem represents a critical operational challenge in our supply chain simulation. While the mathematical formulation may appear straightforward, the underlying complexity stems from multiple factors including network dynamics, capacity constraints, and the need for real-time decision-making at scale. We focus on single-item order currently. The task is for an item at time t, selecting the right FC to fulfill this item. We use x_t to represent the feature of the item, which includes crucial characteristics such as geographic location, shipping speed requirements, physical dimensions, and special handling needs. w_t^i denotes FC i's feature, where $i \in \{1,\ldots,N_t\}$, encompassing attributes such as current capacity, labor availability, and inventory positions. The number of candidate FCs (N_t) can vary over time due to inventory availability and operational status changes.

The complexity of this selection process is magnified by several factors. First, the feature space is heterogeneous, combining continuous variables (distances, dimensions) with categorical data (shipping speeds, special handling requirements). Second, the decision must be made quickly in simulations since we are simulating millions of products that could be fulfilled from hundreds of FCs. Third, the optimal choice depends on both current network state and anticipated future conditions, which is really hard to be captured by forward-looking simulations. Last but not least, the selection impacts both immediate order fulfillment and longer-term network balance, which affects the equilibrium of the future weeks we are simulating.

Assume $g_t \in \{1,\ldots,N_t\}$ is a FC that have inventory in-stock at time t, the probability that FC is chosen is $p(g_t|x_t,w_t)$. The score output from the model for each FC that have inventory in-stock is denoted as:

$$s_t^i = f(x_t, w_t|\theta),$$

where θ are trainable parameters. This scoring function must capture complex interactions between item features and FC characteristics while remaining computationally efficient. The fulfillment probability is generated after applying softmax on the scores:

$$\mathbb{P}(i \text{ is chosen}|x_t,w_t) = \frac{e^{s_t^i}}{\sum_j e^{s_t^j}}$$

This probabilistic approach allows for uncertainty in the decision-making process and provides a natural way to balance multiple objectives. However, training such a model presents its own challenges, including the need for large-scale historical data, handling of temporal dependencies, and adaptation to changing network conditions.

3 FEATURE SELECTION

The training data comprises two essential feature sets that capture both item-specific characteristics and FC attributes, carefully selected to optimize the source FC selection process. They correspond to the x_i's in Figure 2.

The first set focuses on characteristics that directly impact shipping decisions and costs. Geographic information (demand latitude and longitude) is crucial for determining optimal shipping routes and delivery times. Physical attributes (weight, dimensions) are vital for capacity planning, shipping cost calculations, and determining appropriate handling methods. The hazmat indicator is included because hazardous materials require special handling procedures and may only be processable by certain FCs with appropriate certifications. The ship option (speed) requirement directly influences which FCs can fulfill the order within the promised delivery window. All relevant item features are summarized in Table 1.

The second set captures FC-specific attributes that determine their suitability for fulfilling particular orders. Geographic coordinates of FCs are essential for calculating shipping distances and times. The pre-computed distance between demand location and FC location (in kilometers) provides a direct measure of proximity, which strongly correlates with both shipping costs and delivery times. The FC name category (one-hot encoded) allows the model to learn FC-specific characteristics, such as capacity constraints, specialization in certain item types, or historical performance metrics. All relevant FC features are summarized in Table 1.

These features were chosen because they directly influence key operational decisions and constraints. From a cost optimization perspective, distance and physical item attributes directly impact shipping costs. Considering delivery time compliance for customer experience, geographic features help ensure promised delivery times can be met. We also consider the operational feasibility such that hazmat status and FC capabilities must be matched. Package dimensions and weights affect FC storage and handling capabilities, which are essential for capacity Management. FC name encoding allows learning from past fulfillment patterns. The combination of these features enables the model to make informed decisions that balance multiple objectives: minimizing costs, meeting delivery promises, ensuring operational feasibility, and maintaining high customer satisfaction levels.

Table 1: Shipped Item Features and Candidate Fulfillment Center Features.

Feature	Description
Demand location	Latitude and longitude of the center of the 5-digit zip code the demand was predicted for
Ship Option	Ship speed for the item, e.g., same day, second day, etc.
Ship Weight	Weight of item for shipping purposes, in pounds
Package dimension	Package length, height and width in inches (length > width > height)
$\mathbb{1}$(Hazmat)	Whether item is hazardous and needs special shipping
FC location	Latitude and latitude of the centre of the FC's 5 digit zip code
Distance	Distance on globe computed between latitude and longitude for the demand zip code and FC zip code in kilometers
FC Name Category	Name of FC in one-hot encoding

4 SIMULATION RESULTS

4.1 MLP Training Setup

To verify the performance of our model, we conducted comprehensive offline simulations focusing on single-item orders. The evaluation process utilized historical order data from March 2, 2025 to March 8, 2025, sampling 1% of orders to ensure computational efficiency while maintaining statistical significance. This sample size represented approximately N million orders across our fulfillment network.

The dataset was systematically divided into three equal portions:

- Training set (1/3): Used for model parameter optimization
- Testing set (1/3): Used for model selection and hyperparameter tuning
- Validation set (1/3): Used for final performance evaluation

Given the multi-class classification nature of the FC selection problem, we employed negative log-likelihood (NLL) as our primary training metric:

$$\mathrm{NLL} = -\frac{1}{N}\sum_{i=1}^{N}\log(p(y_i|x_i)) \tag{2}$$

where y_i represents the true FC selection and $p(y_i|x_i)$ is the model's predicted probability for the correct FC given input features x_i.

Our MLP model achieved an NLL of 1.43 on the validation set, compared to our benchmark zip2-level model's NLL of 2.01. This represents a significant 28.9% improvement in predictive accuracy. The zip2-level benchmark model, which operates at a coarser geographic granularity (using only the first two numbers from zip codes), serves as a relevant comparison point as it reflects the current industry standard approach. Our offline simulation methodology evaluates the MLP model using historical out-of-sample orders. For each historical order, we reconstruct the feature set available at decision time and use our trained model to generate fulfillment probabilities. We then simulate fulfillment decisions and compare them against actual historical choices, enabling comprehensive performance assessment across different dimensions of our fulfillment network. We focused on order-level metrics, using the MLP model to generate fulfillment probabilities for each candidate FC. The model demonstrates robust performance across different geographic regions and order types, with consistent improvements over the benchmark model across various operational scenarios.

4.2 Order-Level Accuracy

Order-level accuracy in fulfillment prediction is crucial for modern supply chain operations due to its direct impact on both operational efficiency and customer satisfaction. Unlike aggregate-level predictions, which may mask individual errors while maintaining good average performance, order-level accuracy ensures that each fulfillment decision optimally balances multiple competing objectives and constraints. The importance of order-level accuracy becomes even more pronounced in modern e-commerce environments where same-day and next-day delivery options are increasingly common. These tight delivery windows leave little room for error in fulfillment center selection. Accurate predictions at the order level enable precise inventory placement, optimize transportation routes, and minimize split shipments, all of which contribute to cost reduction and improved customer experience.

To evaluate order-level accuracy, we employ a rigorous methodology. For each iteration, we select a FC based on the model's predicted probability distribution. This predicted FC is then compared against the ground truth FC. The accuracy is calculated as the percentage of correct selections over multiple iterations. Specifically, we run 100 iterations to obtain a robust average performance metric. Our model achieves an impressive order-level accuracy of 56.75%. This represents a substantial improvement over our benchmark zip2 level model, which only attains an accuracy of 36.71%. The 2,000 basis point improvement demonstrates the significant predictive power of our approach, enabling more precise and efficient fulfillment decisions at scale.

4.3 Node-Level Accuracy

Node-level accuracy is critical for effective capacity and labor planning in fulfillment networks. Each fulfillment center represents a complex operational unit with specific constraints around processing capacity, labor availability, and storage capabilities. Accurate prediction of order volume at the node level directly impacts several key operational decisions. From a labor planning perspective, accurate node-level predictions enable precise workforce scheduling and management. This includes determining appropriate staffing levels for different shifts, planning seasonal hiring needs, and optimizing training programs. Even small deviations in volume predictions can lead to either costly overstaffing or operational bottlenecks due to understaffing. For capacity planning, node-level accuracy affects both short-term operational decisions and long-term strategic planning. Daily and weekly volume predictions influence decisions about equipment utilization, dock door scheduling, and inventory placement. Over longer horizons, these predictions inform capital investment decisions, such as facility expansions or automation initiatives. Moreover, node-level accuracy is essential for network balancing. Accurate predictions help prevent individual nodes from becoming overtaxed while others remain underutilized. This balance is crucial for maintaining consistent service levels across the network while optimizing resource utilization and operational costs.

To evaluate the node-level accuracy, for each iteration, we calculate the expected drain units for each FC based on the fulfillment probability distribution. We run 100 iterations to calculate the average expected drain for each order. We then aggregate all orders together to get the node-level total drain. We compare with the actual observations and calculate the weight mean absolute percentage error (wMAPE). Our node-level accuracy is 69.51%. The inbound node level accuracy in our simulations is generally close to 30%, which is only half of the fulfillment accuracy at the node level.

4.4 Loss Function Regularization

The inherent tension between order-level and node-level accuracy presents a significant challenge in fulfillment center selection modeling. While order-level accuracy focuses on making optimal individual decisions, node-level accuracy ensures that the aggregate volume predictions for each fulfillment center align with operational capacity and planning requirements. This dual objective creates a complex optimization problem where improving one metric might come at the expense of the other.

Order-level accuracy prioritizes selecting the optimal FC for each individual order, considering factors such as shipping speed, distance, and item characteristics. However, optimizing solely for order-level accuracy can lead to imbalanced node-level predictions, potentially suggesting volume distributions that exceed FC capacity constraints or create inefficient network utilization patterns. Conversely, focusing exclusively on node-level accuracy might sacrifice optimal individual order routing to maintain aggregate volume targets.

To address this trade-off, we introduce a regularization term in the negative log-likelihood (NLL) loss function:

$$\mathscr{L} = \mathscr{L}_{NLL} + \lambda \mathscr{L}_{node}, \tag{3}$$

where $\mathscr{L}_{NLL}$ represents the standard negative log-likelihood loss for order-level predictions, $\mathscr{L}_{node}$ is the node-level regularization term, and λ is a hyperparameter controlling the trade-off between the two objectives. The node-level regularization term is designed to penalize predictions that deviate significantly from historical node-level volume patterns:

$$\mathscr{L}_{node} = \sum_{i=1}^{N} (p_i - t_i)^2, \tag{4}$$

where p_i represents the predicted proportion of orders assigned to node i, t_i is the target proportion based on historical data, and N is the number of nodes in the network.

This regularization approach helps maintain node-level accuracy by encouraging the model to respect historical volume distributions while still allowing for flexibility in individual order assignments. The hyperparameter λ can be tuned to balance the potentially competing objectives based on operational priorities. A larger λ value places more emphasis on node-level accuracy, while a smaller value prioritizes order-level optimization.

Empirical results demonstrate that this regularized approach achieves a better balance between order and node-level accuracy compared to single-objective optimization. For example, with appropriate tuning of λ, we observe that node-level volume predictions improve to 47.13% while maintaining order-level accuracy at 50.31%. This represents a significant improvement in node-level accuracy over non-regularized approaches, compared with optimizing solely for order-level accuracy.

These results, while seeming modest at first glance, represent a significant advancement in addressing the key challenges outlined in Section 1.1. The order-level accuracy of 50.31% and node-level accuracy of 47.13% should be interpreted in the context of the problem's extreme complexity. Our model handles millions of daily orders across hundreds of fulfillment centers, a decision space with billions of possible combinations. This performance represents a 20 percentage point increase over previous benchmarks, a substantial leap in this domain. Simultaneously improving both order-level and node-level accuracy is a challenging feat, balancing conflicting objectives. In practical terms, this improvement means correctly predicting the optimal fulfillment center for 20 additional orders out of every 100, leading to significant cost savings at scale. Notably, these accuracy levels are achieved while maintaining millisecond-level inference times, crucial for real-time decision-making in large-scale operations. Our model effectively captures non-stationary patterns and real-time operational dynamics without explicit modeling, tackling the complexity of the decision space. By incorporating both order-level and node-level considerations through our novel regularization approach, we've created a versatile tool that balances individual order optimization with network-level efficiency, a previously elusive goal in fulfillment simulation. This balanced approach, combined with the model's computational efficiency, represents a significant step forward in solving one of the most complex challenges in supply chain optimization.

Table 2: Model Performance Comparison.

Metric	**Model Version**	**Performance**	**Benchmark**
Negative Log-likelihood	Base MLP	1.43	2.01
	(Improvement)	>2000 bps	
Order-level accuracy	Base MLP	56.75%	36.71%
	(Improvement)	>2000 bps	
Node-level accuracy (Fulfillment)	Base MLP	69.51%	–
Node-level accuracy (Inbound)	Base MLP	30%	–
Regularized model	Order-level	50.31%	56.75%
	Node-level	47.13%	69.51%

Note: Base MLP represents our primary model without regularization. Benchmark refers to the zip2-level model. Node-level accuracy for fulfillment significantly outperforms inbound accuracy. The regularized version shows a trade-off between order-level and node-level accuracy, achieving better balance.

5 CONCLUSION

This research introduces a novel approach to supply chain network simulation, addressing critical challenges in fulfillment path selection through the development of a high-fidelity neural network-based emulator. The significance of this work lies in its ability to bridge the gap between computational efficiency and predictive accuracy in complex fulfillment systems, offering valuable insights for strategic planning and operational evaluation. The core innovation of our approach is the design of a feed-forward neural network model that estimates probability distributions over fulfillment nodes for each order. By incorporating fine-grained spatial resolution at the ZIP5 level, integrating shipping speed categories, and considering regional alignment indicators, our model captures nuanced geographic behaviors and operational constraints that are crucial for accurate simulation of real-world supply chain dynamics.

The model demonstrates significant performance improvements, achieving an NLL of 1.43 compared to the benchmark's 2.01, representing a 28.9% improvement. Order-level accuracy reaches 56.75%, substantially outperforming the benchmark's 36.71%. For node-level predictions, our model achieves 69.51% accuracy in fulfillment volumes, though inbound accuracy remains at 30%. Through our novel regularization approach, we achieve a balanced trade-off, maintaining 50.31% order-level accuracy while improving node-level performance to 47.13%.

The implementation of this high-fidelity emulator has far-reaching implications for the field of supply chain simulation. By operating at simulation speeds while maintaining high fidelity to production behavior, our model enables more comprehensive and accurate analyses of network performance. This balance between speed and accuracy opens up new possibilities for scenario planning, risk assessment, and strategic decision-making in complex fulfillment networks. Furthermore, the granularity of our approach, particularly in terms of spatial resolution and the inclusion of delivery speed proxies, provides deeper insights into the intricate dynamics of fulfillment operations. This level of detail allows for more precise identification of potential bottlenecks, optimization opportunities, and the impact of policy changes on network performance.

However, it is important to acknowledge that the increased fidelity and complexity of our model may present challenges in terms of data requirements and computational resources. Future research should focus on optimizing the balance between model complexity and practical implementation, possibly exploring techniques such as model compression or distributed computing to further enhance scalability.

In conclusion, our research represents a significant step forward in the field of supply chain simulation, offering a powerful tool for decision-makers to navigate the complexities of modern fulfillment networks. By combining advanced machine learning techniques with domain-specific knowledge, we have created an emulator that not only improves upon existing models in terms of accuracy but also provides a flexible framework for future advancements in supply chain optimization. As supply chains continue to grow in complexity and global reach, the need for sophisticated simulation tools will only increase.

REFERENCES

Acimovic, J., and S. C. Graves. 2015. "Making Better Fulfillment Decisions on The Fly in An Online Retail Environment". *Manufacturing & Service Operations Management* 17(1):34–51.

Blei, D. M., A. Y. Ng, and M. I. Jordan. 2003. "Latent Dirichlet Allocation". *Journal of Machine Learning Research* 3:993–1022.

Dolara, A., F. Grimaccia, G. Magistrati, and G. Marchegiani. 2017. "Optimization Models for Islanded Micro-grids: A Comparative Analysis between Linear Programming and Mixed Integer Programming". *Energies* 10(2):241.

Farias, V., J. Gijsbrechts, A. Khojandi, T. Peng, and A. Zheng. 2024. "Speeding up Policy Simulation in Supply Chain RL". *arXiv* https://doi.org/10.48550/arXiv.2406.01939.

Houlsby, N., F. Huszar, Z. Ghahramani, and J. Hernández-lobato. 2012. "Collaborative Gaussian Processes for Preference Learning". *Advances in neural information processing systems* 25:2096–2104.

Kang, Y. 2018. "An Order Control Policy in Crowdsourced Parcel Pickup and Delivery Service". *Advances in Production Management Systems. Smart Manufacturing for Industry 4.0* AICT-536:164–171.

Kavuk, E. M., A. Tosun, M. Cevik, A. Bozanta, S. B. Sonuç, M. Tutuncu, *et al*. 2022. "Order Dispatching for An Ultra-fast Delivery Service via Deep Reinforcement Learning". *Applied Intelligence*:1–26.

Kim, C. O., I.-H. Kwon, and C. Kwak. 2010. "Multi-agent Based Distributed Inventory Control Model". *Expert Systems with Applications* 37(7):5186–5191.

Velásquez-Bermúdez, J. M., M. Khakifirooz, and M. Fathi. 2019. *Large Scale Optimization in Supply Chains and Smart Manufacturing: Theory and Applications*, Volume 149. Springer Nature.

AUTHOR BIOGRAPHIES

WEILONG WANG is an Applied Scientist II in Supply Chain Optimization Technologies at Amazon. His research interests include causal inference, experiment design, machine learning, optimization, and simulation, with applications to capacity management, unbiased treatment effect estimation, and general supply chain management. His email address is wweilong@amazon.com.

MICHAEL BLOEM a Senior Applied Scientist in Supply Chain Optimization Technologies at Amazon. He deploys mathematical optimization, simulation, and machine learning to solve cross-functional problems at the intersections of buying, placement, and capacity. He received his B.S.E. in electrical and computer engineering and economics from Calvin College in 2004, his M.S. in electrical and computer engineering from the University of Illinois at Urbana-Champaign in 2007, and his PhD in operations research from the Department of Management Science and Engineering at Stanford University in 2015. His email address is bloemm@amazon.com.

JINXIANG GU is a Senior Manager, Research Science in Supply Chain Optimization Technologies at Amazon. He has experience working at Amazon and holds an educational background from the Georgia Institute of Technology. Gu is actively involved in research and has published several papers in the field of Operations Research. His email address is gjinxian@amazon.com.

YUNAN LIU is a Principal Research Scientist in the Supply Chain Optimization Technology team at Amazon. He is also an Adjunct Professor in the ISE Department of NC State University. He earned his Ph.D. in Operations Research from Columbia University. His research interests include stochastic modeling, simulation, optimal control and online learning, with applications to supply chain and call centers. His work was awarded first place in the INFORMS Junior Faculty Interest Group Paper Competition in 2016. His email address is yunanliu@amazon.com. His website is https://yliu48.github.io/.