INSPIRE-GNN: Intelligent Sensor Placement to Improve Sparse Bicycling Network Prediction via Reinforcement Learning Boosted Graph Neural Networks

Anonymous Author
Anonymous Institution

Abstract

Accurately estimating bicycling volumes in urban networks is hindered by sparse sensor data, resulting from limited bike sensor coverage. To address this, we propose INSPIRE-GNN, a reinforcement learning-boosted hybrid Graph Neural Network framework that integrates Graph Convolutional Networks (GCN) and Graph Attention Networks (GAT) for optimal sensor placement. Applied to Melbourne's bikeable road network of 15,933 segments with only 141 existing sensors (99% sparsity), INSPIRE-GNN strategically selects additional sensor locations for deployments of 50, 100, 200, and 500 sensors to enhance link-level volume estimation accuracy. Comparative evaluations against random placement and heuristic-based methods, including betweenness and closeness centrality, demonstrate that INSPIRE-GNN significantly outperforms these approaches. Our data-driven strategy facilitates improved sensor coverage, leading to more accurate bicycling data collection and supporting sustainable urban planning initiatives.

1 INTRODUCTION

Urban bicycling networks have gained substantial attention as cities aim to promote bicycling as a sustainable mode of transportation, contributing to public health, environmental sustainability, and improved urban mobility (Pucher and Buehler, 2012). Accurate estimation of bicycling volumes at the link level — predicting the number of cyclists on specific road/path segments — is crucial for understanding demand, guiding infrastructure investments, improving connectivity, and enhancing safety (Bhowmick et al., 2023; Heinen et al., 2010). However, this task faces significant challenges due to high data sparsity resulting from limited sensor coverage in bicycling networks. Globally, urban areas frequently face challenges related to the limited deployment of sensors for tracking bicycling volumes, leading to significant gaps in comprehensive cycling data (Pucher and Buehler, 2012; Zhao and Li, 2017). Studies have highlighted that many cities worldwide maintain minimal sensor coverage, which impedes the ability to accurately monitor and analyze bicycling patterns. For instance, in Melbourne, Australia, the bicycling road network comprises approximately 15,933 road/ path segments, yet only 141 of these segments (roughly 1%) are equipped with sensors to record bicycling counts. This results in approximately 99% data sparsity, posing substantial challenges for accurate link-level volume estimation (Winters et al., 2016). Unlike motorized transportation networks, which benefit from extensive sensor coverage such as loop detectors, cameras, and automated traffic counters (Gu et al., 2017; Klein et al., 2006), bicycling networks have historically been overlooked, leading to significant gaps in sensor coverage and high data sparsity (Buehler and Dill, 2016). These challenges necessitate innovative approaches to maximize the value of limited count data and propose strategic avenues to expand the network of cycling count sensors.

Traditional statistical models and machine learning approaches have been employed for bicycling volume estimation (Griswold et al., 2011; Lin and Fan, 2020; Livingston et al., 2021). However, these methods often rely on comprehensive datasets with adequate coverage and representation to generalize effectively, which is challenging in sparse data environments (Salon, 2016). Moreover, they frequently fail to account for the complex spatial dependencies within transportation networks, limiting their ability to accurately

Preliminary work. Under review by AISTATS 2025. Do not distribute.

model bicycling volumes across urban road networks.

We propose INSPIRE-GNN (**In**telligent **S**ensor **P**lacement to **I**mprove Sparse bicycling **Ne**twork Prediction via **Re**inforcement Learning-Enhanced Graph Neural Networks), an intelligent sensor placement framework designed to improve bicycling volume estimation in networks with sparse sensor coverage. INSPIRE-GNN integrates a Graph Neural Network (GNN) model with a Deep Q-Network (DQN)-based Reinforcement Learning (RL) agent to adaptively learn optimal sensor placement strategies. By strategically selecting additional sensor locations, the framework aims to maximize the performance of bicycling volume estimation across the entire network.

We apply INSPIRE-GNN to the bicycling network in Melbourne, utilizing crowd-sourced bicycle volume data from Strava (Strava Metro contributors, 2022) and infrastructure data from OpenStreetMap (OSM) (OpenStreetMap contributors, 2022). The RL agent is trained to select nodes that yield the most significant improvement in the performance of the GNN for the task of link-level bicycling volume estimation. We compare INSPIRE-GNN's sensor placement policy against random and heuristic-based policies, such as betweenness centrality, closeness centrality, and observed bicycling activity (Freeman, 1977; Newman, 2005; Paluch et al., 2020; Senturk and Akkaya, 2014).

Our primary contributions are as follows:

- **Framework Development:** We propose INSPIRE-GNN, an RL-enhanced GNN framework that optimizes sensor placement to improve bicycling volume estimation in networks with limited sensor coverage and high data sparsity. We integrate Graph Convolutional Networks (GCN) with Graph Attention Networks (GAT) within a Deep Q-Network (DQN)-based reinforcement learning agent. This combination effectively captures both local and global graph structures, enabling the learning of optimal sensor placement strategies that maximize estimation accuracy across the network.
- **Empirical Validation:** We validate INSPIRE-GNN on Melbourne's bicycling network, demonstrating its effectiveness in sensor placement scenarios and confirming its practical utility in real-world urban environments.
- **Comprehensive Evaluation:** We conduct a thorough comparison against traditional heuristic sensor placement methods, including random selection, betweenness centrality, closeness centrality, and observed bicycling activity. Our results showcase significant improvements in link-level estimation performance, highlighting the advantages of our approach over conventional strategies.
- **Practical Insights:** We provide actionable insights for urban planners on expanding sensor networks in a cost-efficient manner. Our findings emphasize the limitations of heuristic approaches and the benefits of RL-driven optimization, guiding policy decisions in sustainable urban planning.

By developing and evaluating INSPIRE-GNN, this work contributes to improving the accuracy of bicycling volume estimation in urban networks with sparse sensor coverage. The findings have practical implications for cities aiming to expand their sensor networks strategically, ensuring that limited resources are utilized optimally for the greatest impact.

2 RELATED WORK

Optimizing sensor placement in urban transportation networks is a critical challenge that has received considerable attention. Traditional heuristic methods based on network centrality measures, such as betweenness and closeness centrality, have been employed to identify key locations for sensor deployment (Paluch et al., 2020; Senturk and Akkaya, 2014). While these approaches are straightforward, they may not effectively capture complex network interactions, especially in sparse sensor environments. Optimization techniques, including greedy algorithms and integer programming, have been explored to maximize coverage and minimize uncertainty in sensor networks (Fakhouri and Soltani, 2020; Fu et al., 2023). Entropy-based methods have also been utilized to select sensor locations that provide the most information gain (Iqbal et al., 2016). However, these methods often require significant computational resources and may not scale well to large urban networks.

Reinforcement Learning (RL) has shown great potential in addressing optimization problems within transportation systems due to its ability to learn optimal policies through interaction with the environment. Applications of RL include traffic signal control (Gao et al., 2017), dynamic routing (Pan and Liu, 2023), and resource allocation (Haydari and Yılmaz, 2020). More general frameworks for graph-based RL have emerged to handle a wide array of combinatorial optimization tasks—ranging from route planning and network design to partitioning and matching. A recent comprehensive survey (Darvariu et al., 2024) unifies these perspectives, illustrating how RL can leverage Graph Neural Networks (GNNs) to model topological and contextual information and then learn policies to solve complex discrete optimization problems over graphs. However, the use of RL for sensor placement optimization in transportation networks, particularly bicycling networks, remains underexplored.

Recent studies have started to integrate Graph Neural Networks with Reinforcement Learning to tackle complex problems that involve both spatial dependencies and sequential decision-making. For example, Chen et al. (2020) proposed a GNN-RL framework for adaptive traffic signal control, capturing the network topology and learning optimal control policies. Similarly, Wang et al. (2024) employed GNNs within an RL framework for traffic flow optimization, demonstrating improved performance over traditional methods.

However, to the best of our knowledge, no prior work has combined GNNs with RL for the specific problem of optimizing sensor placement in sparse bicycling networks. This integration holds the promise of leveraging the strengths of both approaches: GNNs' ability to model complex spatial relationships and RL's capability to learn optimal policies for sequential decision-making.

Our work addresses this gap by introducing INSPIRE-GNN, an RL-enhanced GNN framework that integrates Graph Convolutional Networks (GCN) and Graph Attention Networks (GAT) within a Deep Q-Network (DQN)-based reinforcement learning agent. This combination leverages GNNs' ability to model complex spatial relationships and RL's capacity for sequential decision-making, enabling the framework to optimize sensor placement and improve link-level bicycling volume estimation in data-sparse urban environments."

3 ACHIEVING INTELLIGENT SENSOR PLACEMENT IN SPARSE NETWORK

3.1 Problem Formulation

Accurately estimating bicycling volumes on each link (road/path segment) within an urban network is essential for informed urban planning and infrastructure development. However, the vast majority of links lack direct bicycling count data, likely due to a combination of factors, including financial constraints and the limited funding for active transport infrastructure, which can restrict the installation of widespread sensor networks and therefore, resulting in high data sparsity. To address this challenge, we aim to strategically place additional sensors to maximize the improvement in link-level bicycling volume estimation across the entire network.

We represent the urban bicycling network as an undirected graph $G = (V, E)$, where $V = \{1, 2, \ldots, N\}$ is the set of nodes, each corresponding to a road/path segment and $E \subseteq V \times V$ is the set of edges, representing the connectivity between segments (i.e., if two segments are directly connected). Each node $i \in V$ is associated with a feature vector $\mathbf{x}_i \in \mathbb{R}^d$, capturing attributes such as road surface type, slope, infrastructure characteristics, and a true bicycling count value $y_i \in \mathbb{R}$, known only for nodes with sensors.

Let $\mathcal{S}_{\text{existing}} \subset V$ denote the set of nodes with existing sensors, and $\mathcal{S}_{\text{unlabeled}} = V \setminus \mathcal{S}_{\text{existing}}$ denote the nodes without sensors. In the City of Melbourne's bicycling network, $|\mathcal{S}_{\text{existing}}| = 141$ and $|V| = 15{,}933$, resulting in approximately 99% data sparsity. Our primary objective is to select a subset $\mathcal{S}_{\text{new}} \subseteq \mathcal{S}_{\text{unlabeled}}$ of K nodes to place additional sensors maximizing the overall accuracy of bicycling volume estimation across all unlabeled nodes. Formally, we aim to solve:

$$\max_{\mathcal{S}_{\text{new}} \subseteq \mathcal{S}_{\text{unlabeled}}, |\mathcal{S}_{\text{new}}| = K} \mathcal{F}\left(\hat{Y} \,\middle|\, S_{\text{train}} = \mathcal{S}_{\text{existing}} \cup \mathcal{S}_{\text{new}}\right) \tag{1}$$

where, $\hat{Y} = \{\hat{y}_i : i \in \mathcal{S}_{\text{unlabeled}}\}$ is the set of estimated bicycling volumes for unlabeled nodes, $\mathcal{F}(\cdot)$ is the evaluation metric (Mean Squared Error) and S_{train} is the set of nodes used for training the estimation model.

3.2 The INSPIRE-GNN Framework

To address the optimization problem defined in Equation 1, we propose the **INSPIRE-GNN** (**In**telligent **S**ensor **P**lacement to **I**mprove Sparse Bicycling Network Prediction via **Re**inforcement Learning Boosted Graph Neural Networks) framework. INSPIRE-GNN integrates two neural networks: a Graph Convolutional Network-Graph Attention Network (GCN-GAT) and a Deep Q-Network (DQN) based Reinforcement Learning (RL) agent. The GCN-GAT is used for bicycling volume estimation, while the DQN adaptively learns optimal sensor placement strategies. The DQN's goal is to maximize the GCN-GAT's performance by strategically selecting sensor locations.

3.2.1 GNN Architecture

The GNN model effectively learns representations from the bicycling network graph $G = (V, E)$ and node features $\mathbf{x}_i$ by combining Graph Convolutional Network (GCN) layers and Graph Attention Network (GAT) layers.

- **GCN Layers with Residual Connections:** Multiple GCN layers with residual connections (He et al., 2016) and layer normalization (Ba,

2016) capture both local and global structural patterns while facilitating deeper network training and stabilizing convergence.

- **Dynamic Edge Feature Encoding:** We compute dynamic edge features through a learnable transformation of existing edge attributes, enhancing the model's capacity to capture nuanced relationships.

- **GAT Layers with Edge Features:** The transformed edge features are integrated into GAT layers (Velickovic et al., 2017), allowing the model to assign importance weights to edges based on their attributes, focusing on the most relevant connections.

- **Global Readout and Prediction:** Node embeddings are aggregated using Global Mean Pooling and Global Max Pooling to form a unified graph-level representation, which is passed through fully connected layers to produce the final bicycling volume estimations $\hat{y}_i$

3.2.2 Reinforcement Learning Agent with Advanced Exploration Strategies

The RL agent selects nodes for sensor placement to significantly improve estimation accuracy. To efficiently navigate the large action space, we enhance a standard DQN agent with advanced exploration strategies.

- **State Representation:** The state s_t at time step t is represented by an aggregated feature vector of the currently selected sensor nodes, specifically the mean features of $S_{\text{train}} = S_{\text{existing}} \cup S_{\text{new}}$.

- **Action Space:** The set of possible actions $\mathcal{A}_t$ consists of indices of candidate nodes in $S_{\text{unlabeled}}$. This is a very large action space, as there are many candidate nodes which sensors can be placed.

- **Advanced Exploration Strategies:** Adaptive Epsilon-Greedy Strategy - An epsilon-greedy policy where the exploration rate ϵ decays over time Tokic (2010), allowing the agent to balance exploration and exploitation effectively. Curiosity-Driven Exploration - Intrinsic rewards based on state novelty encourage the agent to explore less-visited states, with intrinsic reward r_t^{int} inversely proportional to the square root of the state's visit count.

- **Reward Function:** The total reward r_t combines extrinsic and intrinsic components:

$$r_t = \left(L_{\text{val}}^{(t-1)} - L_{\text{val}}^{(t)} \right) + \beta \cdot r_t^{\text{int}} \tag{2}$$

where $L_{\text{val}}^{(t)}$ is the validation loss at time t, and β is a weighting factor for the intrinsic reward.

3.3 Training Procedure

The training of the GNN model and the RL agent is interleaved within the INSPIRE-GNN framework.

3.3.1 GNN Model Training

The GNN model minimizes the Mean Squared Error (MSE) between predicted and true bicycling volumes at sensor-equipped nodes:

$$L_{\text{GNN}} = \frac{1}{|S_{\text{train}}|} \sum_{i \in S_{\text{train}}} (y_i - \hat{y}_i)^2 \tag{3}$$

Initially trained using S_{existing}, the model is retrained after each new sensor placement. The Adam optimizer (Kingma, 2014) with early stopping based on validation loss is used to prevent overfitting.

3.3.2 RL Agent Training

The RL agent learns a policy $\pi(a_t \mid s_t; \theta)$ to maximize the expected cumulative reward:

$$\max_{\theta} \mathbb{E} \left[\sum_{t=1}^{K} r_t \mid \pi(a_t \mid s_t; \theta) \right] \tag{4}$$

where, $\gamma \in [0, 1]$ is the discount factor

The DQN agent minimizes the Temporal Difference (TD) loss:

$$L_{\text{DQN}} = \mathbb{E}_{(s_t, a_t, r_t, s_{t+1})} \left[(y_t - Q(s_t, a_t; \theta))^2 \right] \tag{5}$$

with target $y_t = r_t + \gamma \max_{a'} Q(s_{t+1}, a'; \theta^-)$, where θ^- are the parameters of the target network.

The training process involves:

Experience Replay: Transitions (s_t, a_t, r_t, s_{t+1}) are stored in a replay buffer to decorrelate samples and stabilize training (Mnih et al., 2015). **Mini-Batch Updates:** The agent samples mini-batches from the replay buffer to perform gradient descent updates on θ. **Adaptive Exploration:** The exploration rate ϵ is decayed over time, and intrinsic rewards encourage exploration. **Target Network Updates:** The target network parameters θ^- are periodically updated to match θ.

3.4 Algorithm

The overall INSPIRE-GNN framework operates as outlined in Algorithm 1.

Algorithm 1 INSPIRE-GNN Framework

Input: Graph $G = (V, E)$, initial sensor nodes S_{existing}, budget K
Initialize: GNN model parameters Φ, RL agent parameters θ, replay buffer D
Train GNN model on S_{existing} to obtain initial estimation
for episode = 1 to NumEpisodes **do**
  Reset environment: $s_0 \leftarrow s_0$ (current sensor placements)
  for $t = 1$ to K **do**
    Select action a_t using policy π (e.g., epsilon-greedy with adaptive ϵ)
    Place sensor at node a_t and update $S_{\text{new}} \leftarrow S_{\text{new}} \cup \{a_t\}$
    Retrain GNN model on $S_{\text{train}} = S_{\text{existing}} \cup S_{\text{new}}$
    Compute extrinsic reward $r_t^{\text{ext}} = L_{\text{val}}(t-1) - L_{\text{val}}(t)$
    Compute intrinsic reward r_t^{int} based on state novelty
    Total reward $r_t = r_t^{\text{ext}} + \beta \cdot r_t^{\text{int}}$
    Observe new state s_{t+1}
    Store transition (s_t, a_t, r_t, s_{t+1}) in D
    Sample mini-batch from D and update θ using SGD
    if modulo t == target update frequency **then**
      Update target network parameters $\theta^- \leftarrow \theta$
    end if
    Update state $s_t \leftarrow s_{t+1}$
  end for
  Decay exploration rate ϵ if using adaptive strategy
end for
Output: Optimized sensor placement S_{new}, trained GNN model

3.5 Theoretical Properties of Optimal Sensor Placement for Prediction Loss Minimization

We propose a theorem that formalizes the INSPIRE-GNN framework's ability to minimize prediction loss through optimal sensor placement.

Theorem 1. *Let $G = (V, E)$ represents an undirected graph of the bicycling network, where V is the set of nodes (road/path segments) and E is the set of edges (connections between nodes). The existing sensor placements are denoted by $\mathcal{S}_{existing} \subset V$. The set of unlabeled nodes (without sensors) is $\mathcal{S}_{unlabeled} = V \setminus \mathcal{S}_{existing}$. The prediction model Φ trained on nodes $\mathcal{S}_{train} = \mathcal{S}_{existing} \cup \mathcal{S}_{new}$ minimizes the Mean Squared Error (MSE) loss $L_{MSE}(\Phi, \mathcal{S}_{val})$, where $\mathcal{S}_{val} \subseteq \mathcal{S}_{unlabeled}$.*

If: Φ is a sufficiently expressive model capable of approximating the underlying function $f : V \to \mathbb{R}$, the RL policy π^* ensures sufficient exploration of the action space $S_{unlabeled}$, and the reward function is aligned with $-L_{MSE}$, meaning the agent is rewarded for reducing the MSE loss. Then the RL agent's optimal policy π^* selects the sensor placement $S_{new} \subseteq S_{unlabeled}$ such that:

$$S_{new} = \underset{S \subseteq S_{unlabeled}, |S| = K}{\arg\min} \mathcal{L}_{MSE}(\Phi, S_{val})$$

4 EXPERIMENTS

In this section, we evaluate the performance of the proposed INSPIRE-GNN framework in improving link-level bicycling volume estimation within sparse sensor networks. The experiments are designed to assess the model's capability in selecting optimal sensor placements and enhancing volume prediction accuracy compared to existing heuristic-based selection methods.

4.1 Experimental Setup

We focused our study on the City of Melbourne's bicycling network, starting with the existing 141 sensor locations corresponding to actual bicycle counters. The current sensors are strategically placed in areas with high cycling volumes, such as major roads and cycling paths, to capture critical data for traffic management and urban planning. The RL agent was tasked with selecting additional sensor locations from the remaining 15,792 segments. We evaluated the INSPIRE-GNN framework by adding 50, 100, 200, and 500 additional sensors, simulating real-world conditions to assess the framework's effectiveness in optimizing sensor placements using Strava-derived bicycling volumes.

To conduct the experiments, we used a combination of OpenStreetMap (OSM) data for network topology and Strava Metro data for bicycling volumes.

OpenStreetMap (OSM) Network Data: This dataset includes 15,933 road and path segments in Melbourne, with attributes such as road surface type, slope, speed limits, and bicycle infrastructure (e.g., bike lanes, shared paths). Each segment serves as a node in our graph representation of the bicycling network, with physical connections modeled as edges. This data forms the foundation for capturing the spatial relationships between road segments.

Strava Metro Data: Strava Metro data provides daily bicycling volumes for all 15,933 segments within the bicycle network in Melbourne. We derived the

Annual Average Daily Bicycle (AADB) counts from these daily volumes, representing the average number of cyclists passing through each segment per day over a year. While Strava data captures only a subset of the total cycling population and is subject to sampling biases, its broad geographic coverage provides estimated bicycling volumes (from Strava users) for every segment, in contrast to the city's 141 physical bicycling count sensors. This allows us to simulate real-world conditions and test the effects of different levels of data sparsity in our experiments.

4.1.1 Dataset Splits

To ensure meaningful generalization of our predictive models and prevent overfitting, we meticulously split the dataset into distinct training, validation, and test sets. Training set comprises 70% of the existing sensor placements (S_{existing}) combined with the newly selected sensor placements (S_{new}) determined by the RL agent. This set is used to train the predictive model Φ. Validation set constitutes 15% of the unlabeled nodes ($S_{\text{unlabeled}}$). This set is utilized for hyperparameter tuning and to monitor the model's performance during training, ensuring that the model does not overfit to the training data. Test Set (15%) contains the remaining 15% of the unlabeled nodes ($S_{\text{unlabeled}}$). This set is strictly used for evaluating the final performance of the predictive model after training and hyperparameter tuning are complete. Training, validation, and test sets are mutually exclusive, ensuring that no overlap exists between them. This separation guarantees that the model's performance on the test set is indicative of its generalization capabilities to unseen data. Further, to account for inherent randomness in the training process, each model was trained and evaluated across **5 independent runs**, each initialized with a unique random seed. Performance metrics reported in the results are the **mean** of these runs, providing a more reliable assessment of the model's performance.

4.1.2 Baseline Methods

To benchmark the performance of INSPIRE-GNN, we compared it against the following baseline methods: textbfRandom Selection: Sensors are placed randomly across the network without any specific criteria. **Betweenness Centrality:** Sensors are placed at nodes with the highest betweenness centrality scores, identifying critical connectors in the network. **Closeness Centrality:** Sensors are placed at nodes with the highest closeness centrality, targeting segments that are centrally located within the network **Observed Bicycling Activity:** Sensors are placed at segments with the highest observed bicycling activity based on Strava data, focusing on frequently used paths.

4.1.3 RL Approaches

We evaluated the INSPIRE-GNN framework using three RL approaches to assess the impact of advanced exploration strategies individually - **Standard Reinforcement Learning (RL):** The agent learns to place sensors based on observed rewards without advanced exploration strategies. This serves as the baseline RL method. **RL with Adaptive Epsilon-Greedy Strategy:** Enhances the standard RL by adaptively balancing exploration and exploitation using a decaying epsilon-greedy policy. The exploration rate ϵ decays over time, allowing the agent to focus more on exploitation as learning progresses. **RL with Curiosity-Driven Exploration:** Incorporates intrinsic rewards based on state novelty to encourage the agent to explore less-visited areas of the network. The intrinsic reward r_t^{int} is inversely proportional to the square root of the state's visit count, promoting exploration of new states.

The performance of each model was assessed using Mean Squared Error (MSE) Loss, Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE).

4.2 Results and Discussion

The experimental results are summarized in Table 1.

4.2.1 Impact of Reinforcement Learning Approaches

The results demonstrate the effectiveness of the INSPIRE-GNN framework with different reinforcement learning (RL) strategies applied individually. The standard RL approach provides significant improvements over heuristic methods, indicating the advantage of learning-based approaches even without advanced exploration strategies. Building on the standard RL, the Adaptive Epsilon-Greedy Strategy enhances performance by better balancing exploration and exploitation. By adaptively adjusting the exploration rate, the agent can explore sufficiently during early training and focus more on exploitation as learning progresses. This strategy achieves lower RMSE and MAE values compared to standard RL in most cases. The Curiosity-Driven Exploration strategy further improves performance by encouraging the agent to explore less-visited states, capturing diverse network patterns. It introduces intrinsic rewards based on state novelty, promoting exploration of novel states that may contain critical information. This approach often attains the lowest MSE Loss and RMSE values; with 50 additional sensors. This pattern persists across different numbers of additional sensors, highlighting the incremental benefits of each advanced ex-

Selection Method	Number of Additional Nodes	MSE Loss	RMSE	MAPE	MAE
Original Counters	0	1976.30	44.46	18.49	33.21
Betweenness Centrality	50	1533.25	39.16	11.63	26.22
Closeness Centrality	50	2694.97	51.91	16.90	36.71
Strava Activity	50	5562.82	74.58	20.69	47.79
Random Selection	50	2217.36	47.09	14.58	32.08
Reinforcement Learning	50	1029.04	32.78	4.36	20.98
Adaptive Epsilon-Greedy	50	1045.92	32.34	4.51	21.24
Curiosity-Driven Exploration	50	**1020.48**	**32.00**	**4.31**	**20.94**
Betweenness Centrality	100	1806.87	42.51	15.04	32.52
Closeness Centrality	100	2200.15	46.91	15.02	33.00
Strava Activity	100	8023.02	89.57	26.65	60.77
Random Selection	100	1900.51	43.59	12.23	28.71
Reinforcement Learning	100	991.79	31.93	5.36	20.20
Adaptive Epsilon-Greedy	100	1000.54	**31.62**	5.20	20.12
Curiosity-Driven Exploration	100	**990.26**	31.80	**5.10**	**20.03**
Betweenness Centrality	200	2079.14	45.60	13.64	30.11
Closeness Centrality	200	3000.51	54.78	18.91	38.68
Strava Activity	200	6896.09	83.04	28.82	61.98
Random Selection	200	1028.62	32.07	8.38	20.45
Reinforcement Learning	200	909.48	31.16	4.05	18.86
Adaptive Epsilon-Greedy	200	910.34	30.15	**4.00**	**18.84**
Curiosity-Driven Exploration	200	**905.67**	**30.08**	4.04	18.89
Betweenness Centrality	500	2167.15	46.55	14.53	31.97
Closeness Centrality	500	2368.84	48.67	13.84	31.07
Strava Activity	500	5409.96	73.55	28.66	57.82
Random Selection	500	1061.85	32.59	8.32	19.87
Reinforcement Learning	500	865.90	29.23	3.54	**17.46**
Adaptive Epsilon-Greedy	500	870.92	29.50	3.61	17.53
Curiosity-Driven Exploration	500	**865.32**	**29.22**	**3.52**	17.48

Table 1: Experimental results (all values are in AADB except for MAPE which is in percentages)

ploration strategy when applied individually.

4.2.2 Comparison with Baseline Methods

When comparing the RL approaches to the heuristic baseline methods, the superiority of RL-based sensor placement becomes evident. The heuristic methods, such as betweenness bentrality and closeness centrality, provide modest improvements but are consistently outperformed by RL methods. Interestingly, the observed bicycling activity selection method results in higher RMSE, suggesting that selecting high-activity areas exclusively does not necessarily contribute to overall estimation accuracy. This indicates that high-activity segments may already be well-represented or may not provide additional valuable information when chosen for sensor placement. Random selection performs inconsistently, underscoring the inefficiency of unguided sensor placement strategies. In contrast, all RL methods significantly outperform the heuristics. The incremental improvements from standard RL to the advanced strategies demonstrate the value of sophisticated exploration techniques. For instance, with 200 additional sensors, betweenness centrality achieves an RMSE of 45.60 AADB, while the standard RL reduces the RMSE to 31.16 AADB. The Adaptive Epsilon-Greedy and Curiosity-Driven Exploration strategies further reduce the RMSE to 30.15 AADB and 30.08 AADB, respectively. These results confirm that RL-based sensor placement is more effective than heuristic methods, and that advanced exploration strategies individually enhance the RL agent's performance.

4.2.3 Effect of Individual RL Exploration Strategies

By applying the advanced exploration strategies one by one, we can isolate their specific impacts on the model's performance. The standard RL approach serves as a strong baseline and demonstrates the fundamental capability of RL in optimizing sensor placement, achieving substantial improvements over heuristic methods. The Adaptive Epsilon-Greedy Strategy improves upon the standard RL by adaptively adjusting the exploration rate. This allows the agent to balance exploration and exploitation effectively, exploring the environment sufficiently during early training and focusing more on exploitation as it learns. This strategy often results in lower MAE and MAPE values, indicating more precise estimations. Curiosity-Driven Exploration encourages the agent to explore novel states by introducing intrinsic rewards based on state novelty. This approach often achieves the lowest RMSE and MSE Loss, suggesting that exploring less-visited areas leads to capturing critical network information that enhances prediction accuracy. It is particularly effective when the number of additional sensors is limited, maximizing the impact of each sensor placement. With 500 additional sensors, Curiosity-Driven Exploration attains the lowest RMSE and the lowest MAPE, indicating both high accuracy and reliability in predictions. These observations indicate that while all RL methods enhance performance over heuristic approaches, advanced exploration strategies contribute unique benefits when applied individually.

4.2.4 Analysis of Performance Metrics

The consistent reduction in all performance metrics across the RL methods reinforces their effectiveness in optimizing sensor placement for bicycling volume estimation. The MSE Loss and RMSE metrics reflect overall estimation accuracy improvements, with advanced RL strategies achieving the lowest values. For instance, Curiosity-Driven Exploration consistently attains the lowest MSE Loss and RMSE, indicating superior performance in minimizing estimation errors. The MAE and MAPE metrics provide insights into absolute and relative errors, respectively. Advanced strategies often yield lower MAE and MAPE values, indicating more reliable and precise predictions. The Adaptive Epsilon-Greedy Strategy, in particular, shows improvements in MAE, suggesting that it enhances the average accuracy of the estimations. The differences between the RL methods, though sometimes subtle, highlight the importance of exploration strategies in refining the agent's learning process. Advanced exploration strategies enable the agent to discover valuable information that may not be accessible through standard RL, leading to better optimization of sensor placements and improved estimation accuracy.

4.3 Implications for Sensor Placement Optimization

The results of this study underscore the potential of the INSPIRE-GNN framework in practical applications for urban planning and infrastructure development. The effectiveness of advanced RL strategies, when applied individually, enhances the agent's ability to identify optimal sensor placements, leading to significant accuracy improvements in bicycling volume estimation. This is particularly valuable for urban planners working under budget constraints, as achieving substantial performance gains with a limited number of additional sensors allows for efficient resource utilization. Furthermore, the framework's ability to incorporate different RL strategies makes it adaptable to various urban network configurations and requirements. Its scalability and adaptability demonstrate that it can be customized to suit different cities' needs, network sizes, and complexities. This flexibility enhances its applicability in real-world scenarios where urban environments and transportation networks vary widely.

4.4 Limitations and Future Work

While the INSPIRE-GNN framework shows promising results, future research could further enhance its effectiveness. Our framework leverages standard exploration methods such as decaying epsilon-greedy and count-based exploration, which have proven effective in RL applications. Future enhancements could incorporate more advanced exploration methods to further enhance the model's performance. Combining exploration strategies may lead to better performance; understanding how these strategies interact could optimize the agent's learning process. Furthermore, evaluating INSPIRE-GNN in diverse urban environments with varying network topologies and traffic patterns would further validate its effectiveness and generalizability. Incorporating temporal dynamics is another avenue. While the current framework primarily addresses spatial sensor placement, integrating temporal dynamics—such as seasonal trends, time of day, and peak hours—could significantly enhance predictive capabilities. This would allow for dynamic sensor placement strategies that adjust over time, making the model more robust and adaptable to changes in cyclist behavior. Extending the framework to other domains, like vehicular traffic or public transportation networks, could evaluate its generalizability and robustness. Applying the framework to different types of networks would assess its adaptability and identify any domain-specific challenges. Finally, integrating additional data sources, such as demographic information, environmental factors, or socioeconomic data, could enrich the model's inputs and improve performance. This approach may capture underlying factors influencing bicycling patterns, leading to more accurate volume estimations and more effective sensor placement strategies.

References

Ba, J. L. (2016). Layer normalization. *arXiv preprint arXiv:1607.06450*.

Bhowmick, D., Saberi, M., Stevenson, M., Thompson, J., Winters, M., Nelson, T., Leao, S. Z., Seneviratne, S., Pettit, C., Vu, H. L., et al. (2023). A systematic scoping review of methods for estimating link-level bicycling volumes. *Transport reviews*, 43(4):622–651.

Buehler, R. and Dill, J. (2016). Bikeway networks: A review of effects on cycling. *Transport reviews*, 36(1):9–27.

Chen, C., Wei, H., Xu, N., Zheng, G., Yang, M., Xiong, Y., Xu, K., and Li, Z. (2020). Toward a thousand lights: Decentralized deep reinforcement learning for large-scale traffic signal control. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 3414–3421.

Darvariu, V.-A., Hailes, S., and Musolesi, M. (2024). Graph reinforcement learning for combinatorial optimization: A survey and unifying perspective. *arXiv preprint arXiv:2404.06492*.

Fakhouri, A. A. and Soltani, R. (2020). Multi-objective robust optimization for the traffic sensors location problem. *IEEE Access*, 9:6225–6238.

Freeman, L. (1977). A set of measures of centrality based on betweenness. *Sociometry*.

Fu, H., Lam, W. H., Ho, H., and Ma, W. (2023). Optimization of multi-type traffic sensor locations for network-wide link travel time estimation with consideration of their covariance. *Transportmetrica B: Transport Dynamics*, 11(1):760–782.

Gao, J., Shen, Y., Liu, J., Ito, M., and Shiratori, N. (2017). Adaptive traffic signal control: Deep reinforcement learning algorithm with experience replay and target network. *arXiv preprint arXiv:1705.02755*.

Griswold, J. B., Medury, A., and Schneider, R. J. (2011). Pilot models for estimating bicycle intersection volumes. *Transportation research record*, 2247(1):1–7.

Gu, Y., Qian, Z., and Zhang, G. (2017). Traffic state estimation for urban road networks using a link queue model. *Transportation research record*, 2623(1):29–39.

Haydari, A. and Yılmaz, Y. (2020). Deep reinforcement learning for intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(1):11–32.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Heinen, E., Van Wee, B., and Maat, K. (2010). Commuting by bicycle: an overview of the literature. *Transport reviews*, 30(1):59–96.

Iqbal, M., Naeem, M., Anpalagan, A., Qadri, N. N., and Imran, M. (2016). Multi-objective optimization in sensor networks: Optimization classification, applications and solution approaches. *Computer Networks*, 99:134–161.

Kingma, D. P. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Klein, L. A., Mills, M. K., Gibson, D. R., et al. (2006). Traffic detector handbook: Volume i. Technical report, Turner-Fairbank Highway Research Center.

Lin, Z. and Fan, W. D. (2020). Modeling bicycle volume using crowdsourced data from strava smartphone application. *International journal of transportation science and technology*, 9(4):334–343.

Livingston, M., McArthur, D., Hong, J., and English, K. (2021). Predicting cycling volumes using crowdsourced activity data. *Environment and Planning B: Urban Analytics and City Science*, 48(5):1228–1244.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533.

Newman, M. E. (2005). A measure of betweenness centrality based on random walks. *Social networks*, 27(1):39–54.

OpenStreetMap contributors (2022). OpenStreetMap. `https://www.openstreetmap.org`.

Paluch, R., Łukasz G. Gajewski, Hołyst, J. A., and Szymanski, B. K. (2020). Optimizing sensors placement in complex networks for localization of hidden signal source: A review. *Future Generation Computer Systems*, 112:1070–1092.

Pan, W. and Liu, S. Q. (2023). Deep reinforcement learning for the dynamic and uncertain vehicle routing problem. *Applied Intelligence*, 53(1):405–422.

Pucher, J. and Buehler, R. (2012). *City cycling*. MIT press.

Salon, D. (2016). Estimating pedestrian and cyclist activity at the neighborhood scale. *Journal of transport geography*, 55:11–21.

Senturk, I. F. and Akkaya, K. (2014). Connectivity restoration in disjoint wireless sensor networks using centrality measures. In *39th Annual IEEE Conference on Local Computer Networks Workshops*, pages 616–622.

Strava Metro contributors (2022). Strava Metro. https://metro.strava.com.

Tokic, M. (2010). Adaptive ε-greedy exploration in reinforcement learning based on value differences. In *Annual conference on artificial intelligence*, pages 203–210. Springer.

Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y., et al. (2017). Graph attention networks. *stat*, 1050(20):10–48550.

Wang, T., Zhu, Z., Zhang, J., Tian, J., and Zhang, W. (2024). A large-scale traffic signal control algorithm based on multi-layer graph deep reinforcement learning. *Transportation Research Part C: Emerging Technologies*, 162:104582.

Winters, M., Teschke, K., Brauer, M., and Fuller, D. (2016). Bike score®: Associations between urban bikeability and cycling behavior in 24 cities. *International journal of behavioral nutrition and physical activity*, 13:1–10.

Zhao, P. and Li, S. (2017). Bicycle-metro integration in a growing city: The determinants of cycling as a transfer mode in metro station areas in beijing. *Transportation research part A: policy and practice*, 99:46–60.

Instructions for Paper Submissions to AISTATS 2025: Supplementary Materials

1 Theorem

Theorem 1. *Let $G = (V, E)$ represents an undirected graph of the bicycling network, where V is the set of nodes (road/path segments) and E is the set of edges (connections between nodes). The existing sensor placements are denoted by $\mathcal{S}_{existing} \subset V$. The set of unlabeled nodes (without sensors) is $\mathcal{S}_{unlabeled} = V \setminus \mathcal{S}_{existing}$. The prediction model Φ trained on nodes $\mathcal{S}_{train} = \mathcal{S}_{existing} \cup \mathcal{S}_{new}$ minimizes the Mean Squared Error (MSE) loss $L_{MSE}(\Phi, \mathcal{S}_{val})$, where $\mathcal{S}_{val} \subseteq \mathcal{S}_{unlabeled}$.*

If: Φ is a sufficiently expressive model to approximate the underlying function $f : V \to \mathbb{R}$, the RL policy π^* ensures sufficient exploration of the action space $S_{unlabeled}$, and the reward function is aligned with $-L_{MSE}$, then the RL agent's optimal policy π^* selects the sensor placement $S_{new} \subseteq S_{unlabeled}$ such that:

$$S_{new} = \underset{S \subseteq S_{unlabeled}, |S|=K}{\arg\min} \mathcal{L}_{MSE}(\Phi, S_{val})$$

2 Proof of Theorem 1

To establish the optimality of the INSPIRE-GNN framework in minimizing the prediction loss through optimal sensor placement, we proceed through the following steps:

2.1 Modeling Sensor Placement as a Markov Decision Process (MDP)

We model the sensor placement task as a finite-horizon Markov Decision Process (MDP) defined by the tuple (S, A, P, R, T), where:

- **States (S):** Each state s_t at time t represents the current set of sensor locations up to step t, i.e., $s_t = \mathcal{S}_{\text{existing}} \cup \mathcal{S}_{\text{new}}^{(t)}$, where $\mathcal{S}_{\text{new}}^{(t)}$ contains the t sensors placed so far.
- **Actions (A):** The action a_t at time t involves selecting a node $v_t \in \mathcal{S}_{\text{unlabeled}} \setminus \mathcal{S}_{\text{new}}^{(t-1)}$ to place a new sensor.
- **Transition Function (P):** The transition from state s_t to s_{t+1} is deterministic and defined as:

$$s_{t+1} = s_t \cup \{v_t\}$$

- **Reward Function (R):** The immediate reward r_t received after placing a sensor at node v_t is defined as the reduction in validation loss:

$$r_t = L_{\text{val}}^{(t-1)} - L_{\text{val}}^{(t)}$$

  where $L_{\text{val}}^{(t)} = L_{\text{MSE}}(\Phi^{(t)}, \mathcal{S}_{\text{val}})$ is the validation loss after placing t sensors and retraining the model $\Phi^{(t)}$.
- **Horizon (T):** The process continues for K steps, corresponding to the budget K of additional sensors to be placed.

2.2 Alignment of Reward Function with the Objective

The cumulative reward G over K steps is given by:

$$G = \sum_{t=1}^{K} r_t = \sum_{t=1}^{K} \left(L_{\text{val}}^{(t-1)} - L_{\text{val}}^{(t)} \right) = L_{\text{val}}^{(0)} - L_{\text{val}}^{(K)}$$

Maximizing the cumulative reward G is equivalent to minimizing the final validation loss $L_{\text{val}}^{(K)}$ since $L_{\text{val}}^{(0)}$ is a constant initial loss before any additional sensors are placed. The RL agent's objective of maximizing G aligns directly with the goal of minimizing the validation MSE loss after placing K new sensors.

2.3 Convergence to the Optimal Policy

Under the given assumptions:

1. **Sufficient Exploration:** Achieved through an ϵ-greedy strategy with decaying ϵ, ensuring that the RL agent explores the action space $\mathcal{S}_{\text{unlabeled}}$ adequately.
2. **Model Expressiveness and Consistency:** The prediction model Φ is capable of accurately learning from the data provided by the chosen sensor placements, ensuring that L_{MSE} effectively captures prediction errors.
3. **Reward Alignment:** The reward function accurately reflects the reduction in validation loss, aligning the agent's incentives with the objective of minimizing L_{MSE}.

Given these, the RL agent employs Q-learning to estimate the optimal action-value function $Q^*(s, a)$. According to the convergence properties of Q-learning (Watkins & Dayan, 1992), under the following conditions:

- The agent explores all state-action pairs infinitely often.
- The learning rate satisfies $\sum_{t=1}^{\infty} \alpha_t = \infty$ and $\sum_{t=1}^{\infty} \alpha_t^2 < \infty$.
- The MDP satisfies the Markov property.

The estimated Q-values converge to the optimal action-value function $Q^*(s, a)$. Consequently, the policy $\pi(s) = \arg\max_a Q^*(s, a)$ converges to the optimal policy π^* that maximizes the expected cumulative reward G, thereby minimizing $L_{\text{val}}^{(K)}$.

2.4 Optimality of Sensor Placement

By converging to π^* (optimal policy), the RL agent selects actions (sensor placements) that maximize G, which corresponds to minimizing $L_{\text{val}}^{(K)}$.

$$\mathcal{S}_{\text{new}} = \underset{S \subseteq \mathcal{S}_{\text{unlabeled}},\, |S|=K}{\arg\min} L_{\text{MSE}}(\Phi^{(K)}, \mathcal{S}_{\text{val}})$$

where $\Phi^{(K)}$ is the model trained on $\mathcal{S}_{\text{train}} = \mathcal{S}_{\text{existing}} \cup \mathcal{S}_{\text{new}}$.

2.5 Effectiveness of the GNN Model

- **Model Capacity:** Given that Φ is sufficiently expressive (e.g., the GNN has enough layers and hidden units), it can accurately approximate the true function f, ensuring that L_{MSE} effectively measures prediction errors.
- **Impact of Sensor Placement:** Strategically placing sensors at nodes that significantly contribute to reducing L_{val} enables Φ to learn a better approximation of f, thereby improving predictions on $\mathcal{S}_{\text{val}}$.
- **Assumption of Stationarity:** The underlying distribution of the data does not change significantly during the learning process, ensuring the model's learning remains valid and the sensor placements remain relevant.

2.6 Conclusion

Combining the above points, we establish that the RL agent, under the optimal policy π^*, selects a set of K additional sensor locations $\mathcal{S}_{\text{new}} \subseteq \mathcal{S}_{\text{unlabeled}}$ that minimizes the validation loss $L_{\text{val}}^{(K)}$. This is formally expressed as:

$$\mathcal{S}_{\text{new}} = \underset{S \subseteq \mathcal{S}_{\text{unlabeled}},\, |S|=K}{\arg\min} L_{\text{MSE}}(\Phi^{(K)}, \mathcal{S}_{\text{val}})$$

Thus, the INSPIRE-GNN framework effectively optimizes sensor placement by leveraging the strengths of both Graph Neural Networks (GNNs) and Reinforcement Learning (RL), ensuring improved accuracy in bicycling volume estimation within sparse sensor networks.

3 Experiment Results

3.1 Initial Baseline Model Comparison - Model Trained with Original Counters

Table 1 presents the performance of various baseline models when trained solely on the existing 141 counters, without additional sensor placements. We compare traditional statistical models (Linear Regression), machine learning approaches (Decision Tree, Random Forest, Gradient Boosting), and a deep learning method (Multi-Layer Perceptron, MLP) against our GAT-GCN hybrid architecture. The metrics reported include MSE Loss, RMSE, MAPE, and MAE. Notably, the GAT-GCN Hybrid model achieves a substantially lower MSE Loss (1976.30) compared to all other baselines, indicating a stronger predictive capability despite the extremely sparse sensor coverage. This initial evaluation demonstrates that even without RL-based sensor placement optimization, leveraging GNN-based architectures offers a significant advantage over conventional methods in estimating bicycling volumes.

Table 1: Initial Baseline Model Comparison - Model Trained with Original Counters

Model	**MSE Loss**	**RMSE**	**MAPE (%)**	**MAE**
Linear Regression	3643.40	60.36	19.97	46.85
Decision Tree	4766.10	69.04	25.69	52.39
Random Forest	4671.40	68.35	26.67	54.56
Graident Boosting	3551.55	59.59	22.34	47.29
MLP	4450.28	66.71	23.33	52.13
GAT-GCN Hybrid	**1976.30**	**44.46**	**18.49**	**33.21**

3.2 Baseline Model Comparison - with Initial 141 Counters and 50 RL Counter Placements

Table 2: Baseline Model Comparison - with Initial 141 counter and 50 RL counter placement

Model	**MSE Loss**	**RMSE**	**MAPE (%)**	**MAE**
Linear Regression	2240.10	47.33	15.08	28.65
Decision Tree	2239.50	47.52	13.66	21.52
Random Forest	1792.79	42.34	11.14	21.63
Graident Boosting	1815.42	42.61	12.06	21.82
MLP	1898.26	43.57	14.49	22.58
GAT-GCN Hybrid	**1020.48**	**32.00**	**4.31**	**20.94**

Table 2 shows the results after deploying an additional 50 sensors selected by our RL-enhanced INSPIRE-GNN framework. We evaluate Linear Regression, Decision Tree, Random Forest, Gradient Boosting, and MLP again under these improved conditions. With the strategic placement of these sensors, all models benefit from enhanced data coverage, reducing their errors relative to the initial scenario. However, our GAT-GCN Hybrid model achieves a markedly lower MSE Loss (1020.48) compared to all tested baselines, reaffirming that the RL-driven sensor placement optimization combined with a GNN-based architecture yields significant gains in accuracy. This table highlights how INSPIRE-GNN outperforms established methods and underscores the importance of informed sensor placement decisions.

3.3 Hyperparameter Sensitivity Analysis of INSPIRE-GNN

In Table 3, we detail the hyperparameter sensitivity analysis performed on INSPIRE-GNN. We vary key parameters such as the hidden dimension sizes of the GNN, the learning rate, and the epsilon decay rate used in the RL component. The performance, assessed by MSE Loss, RMSE, MAPE, and MAE, reveals that certain configurations (e.g., a hidden dimension of 128, a learning rate of 0.001, and an epsilon decay rate of 500) consistently yield the best results (MSE Loss = 1020.48). This analysis provides insights into which hyperparameters have the most significant impact on the model's predictive accuracy and stability, guiding future refinements and assisting practitioners in tuning the framework effectively.

Table 3: Hyperparameter Sensitivity Analysis of INSPIRE-GNN

Hyperparameters	**Value**	**MSE Loss**	**RMSE**	**MAPE (%)**	**MAE**
Hidden Dimensions	32	1431.65	37.46	9.73	23.71
	64	1174.3	34.53	5.05	22.68
	128	**1020.48**	**32**	**4.31**	**20.94**
	256	1144.65	33.2	4.53	21.3
	512	1252.8	35.36	5.72	22.8
Learning Rate	0.0001	1203.52	34.64	5.5	23.11
	0.001	**1020.48**	**32**	**4.31**	**20.94**
	0.005	1146.75	34.11	4.53	21.54
	0.01	1333.91	36.06	5.86	23.59
Epsilon Decay Rate	200	1127.65	33.2	4.55	21.37
	300	1055.4	33.37	4.42	21.25
	500	**1020.48**	**32**	**4.31**	**20.94**
	700	1080.79	32.85	4.6	20.67
	1000	1150.33	34.12	5.17	21.38

3.4 Ablation Study of INSPIRE-GNN (Number of Additional Nodes: 50)

Table 4 presents an ablation study evaluating the importance of each component in our framework. We compare the full INSPIRE-GNN model (Hybrid GAT-GCN + RL with Curiosity-Driven Exploration) against three variants: (1) Without RL, using the hybrid GAT-GCN architecture but selecting new sensors randomly; (2) Using Only GCN with RL; and (3) Using Only GAT with RL. The full INSPIRE-GNN model achieves a drastically lower MSE Loss (1020.48) than any ablated version, demonstrating the necessity of both the hybrid GAT-GCN construction and the RL-driven sensor placement strategy. These results confirm the synergistic effect of combining GCN and GAT layers, as well as employing curiosity-driven RL exploration, to achieve optimal sensor placement and superior bicycling volume estimation.

Table 4: Ablation Study of INSPIRE-GNN (Number of Additional Nodes: 50)

Model Variant	**Components Included**	**MSE Loss**	**RMSE**	**MAPE (%)**	**MAE**
Full INSPIRE-GNN	Hybrid GAT-GCN + RL with Curiosity-Driven Exploration	1020.48	32	4.31	20.94
Without RL	Hybrid GAT-GCN + Random Sensor Placement	2217.36	47.09	14.58	32.08
Using Only GCN	GCN + RL with Curiosity-Driven Exploration	1610.25	39.73	10.5	25
Using Only GAT	GAT + RL with Curiosity-Driven Exploration	1483.51	37.45	9.8	23.5