Multi-Task Rider Behavior Modeling for Micromobility Systems

1st Birva Sevak
Department of Computer and Information Science
University of Michigan-Dearborn
Dearborn, USA
birvas@umich.edu

2nd Shrenik Jadhav
Department of Computer and Information Science
University of Michigan-Dearborn
Dearborn, USA
shrenikj@umich.edu

***Abstract*—Shared micromobility operators routinely rely on separate models to predict trip duration, demand patterns, and vehicle choice, even though these outcomes are driven by shared latent factors such as trip purpose, time of day, and station context. This paper proposes a hierarchical multi-task variational autoencoder (HVAE) that learns a compact latent representation of trip intent from information available at trip start and jointly predicts trip duration, normalized station demand contribution, and rideable type. The model uses a shared encoder with station embeddings and a two-level latent structure that separates population-level intent from residual individual variation, followed by task-specific decoder heads for regression and classification. Using a strict temporal split of one year of Divvy (Chicago) trip records, the proposed HVAE consistently outperforms strong single-task baselines trained on the same inputs, improving duration prediction error and demand estimation while also increasing rideable-type classification accuracy. In addition, clustering the learned global latent codes yields interpretable intent groups with distinct temporal signatures, and reconstruction-based scores provide an unsupervised signal for flagging unusual trips via percentile thresholds. To support reproducibility and fair comparison, we emphasize consistent evaluation definitions for label handling, metric aggregation, and anomaly-score formulation across experiments and figures.**

***Index Terms*—anomaly detection, bike sharing, demand forecasting, latent representation learning, micromobility, multi-task learning, rideable type classification, trip duration prediction.**

I. Introduction

Shared micromobility systems have rapidly expanded in urban centers worldwide, generating vast datasets of trip records that capture millions of user interactions daily. These systems, which include bike sharing, electric scooters, and other short-distance transportation options, provide valuable insights into urban mobility patterns, operational efficiency, and user behavior. The Divvy bike sharing system in Chicago, for example, recorded over 5.7 million trips between April 2024 and March 2025 [1], representing a rich source of information for understanding rider behavior and system dynamics.

Current approaches to analyzing bike sharing data typically fragment the prediction problem into separate, isolated tasks. Operators commonly build distinct models to predict trip duration, forecast destination stations, classify rider types, and detect anomalous behavior. However, this fragmented approach fails to recognize that these behavioral outcomes are fundamentally interconnected. A casual rider embarking on a weekend trip from a lakefront station exhibits markedly different patterns in duration, destination preferences, and travel speed compared to a member commuting during weekday rush hours. By treating these prediction tasks independently, existing methodologies discard the rich, shared behavioral structure that underlies user interactions with micromobility systems.

The central hypothesis of this work is that a latent "trip intent" represents a tangible, learnable construct that can be extracted from data available at the moment a trip begins. We propose that by explicitly modeling this intent, we can predict multiple behavioral outcomes more accurately and earlier in the trip lifecycle than conventional single-task approaches. This unified representation enables operators to anticipate how long a vehicle will be unavailable, where it will likely be returned, and whether the trip exhibits characteristics requiring operational attention.

We address this challenge through a multi-task neural network architecture that jointly predicts four key outcomes using only information available at trip initiation: trip duration, destination station, rider type classification, and anomaly probability. The model employs a shared encoder that processes features observable at unlock time including start station, temporal context, bike type, and local demand patterns to learn a single latent representation of trip intent. This shared representation is then fed to four specialized decoder heads, one for each prediction task. The entire architecture is trained end to end, forcing the encoder to discover common behavioral patterns that drive all four outcomes simultaneously.

This approach offers significant operational value. Within seconds of a ride starting, system operators gain a unified forecast encompassing vehicle availability duration, likely return location, and potential operational concerns. This enables more proactive fleet management, improved rebalancing strategies, and enhanced risk mitigation. From a research perspective, the novelty lies in explicitly formulating rider behavior as a multi-task learning problem with a shared latent structure. Unlike prior work that focuses on isolated predictions or employs simple rule-based anomaly detection, we demonstrate that a single, interpretable representation can capture the diverse behavioral patterns underlying multiple prediction tasks.

We evaluate our approach using the publicly available

Divvy bike sharing dataset from Chicago [1], which spans twelve months and includes detailed trip records with temporal, spatial, and rider characteristics. To our knowledge, this represents the first application of hierarchical multi-task learning to comprehensive bike sharing behavior prediction. Our contributions can be summarized as follows:

- We introduce a unified multi-task framework that jointly predicts trip duration, destination, rider type, and anomaly probability from trip initiation features, demonstrating that shared latent representations improve performance across all tasks.
- We develop an interpretable latent space that captures meaningful trip intent factors, enabling operators to understand the behavioral drivers behind model predictions.
- We demonstrate that multi-task learning with shared encoders outperforms independent single-task baselines on a large-scale real-world dataset, providing both quantitative improvements and qualitative insights into rider behavior patterns.
- We provide a comprehensive analysis of learned latent representations, confirming that the model discovers operationally relevant behavioral modes such as weekday commuting, weekend leisure, and potential misuse patterns.

The remainder of this paper is organized as follows. Section II reviews related work on micromobility prediction, multi-task learning, and latent representation learning. Section III details our proposed multi-task architecture and training methodology. Section IV describes the dataset, preprocessing steps, and evaluation metrics. Section V presents experimental results comparing our approach to single-task baselines. Section VI analyzes the learned latent representations and their interpretability. Finally, Section VII concludes with discussions of implications and future work.

II. Related Work

This section reviews the literature on bike sharing prediction systems, multi-task learning architectures, deep learning for transportation, anomaly detection in mobility systems, and latent representation learning. We identify gaps in existing work that motivate our unified multi-task approach.

A. Bike Sharing and Micromobility Prediction

The growth of bike sharing systems has generated substantial research on demand forecasting and usage prediction. Lin *et al.* [2] proposed a Graph Convolutional Neural Network with Data-driven Graph Filter (GCNN-DDGF) for station-level hourly demand prediction in NYC's Citi Bike system. Their approach automatically learns hidden heterogeneous pairwise correlations between stations through data-driven graph filters, achieving superior performance over traditional spatial adjacency-based GCNNs. Notably, their use of trip duration matrices as one adjacency representation demonstrates the interconnection between duration and demand patterns.

Li *et al.* [3] introduced a hierarchical prediction model using bipartite clustering to group stations with similar usage patterns, enabling proactive bike rebalancing. While effective for cluster-level prediction, their multi-similarity inference approach predates modern deep learning methods. More recently, Li *et al.* [4] developed the Spatial-Temporal Memory Network (STMN) using Conv-LSTM modules to capture closeness, period, and trend patterns across four international bike-sharing systems, demonstrating cross-city model transferability.

Li *et al.* [5] addressed limitations of grid-based CNNs by proposing irregular convolution that operates over "semantic neighbors"locations with similar temporal usage patterns regardless of geographic distance. Their IrConv+LSTM model captures hidden linkages in demand patterns that spatial adjacency alone cannot detect. Jiang [6] provides a comprehensive survey of 55 papers on deep learning for bike-sharing, confirming that most existing work focuses on single-task demand prediction while trip-level modeling (duration, destination) remains underexplored, a gap our work directly addresses.

B. Deep Learning for Transportation Prediction

Deep learning has transformed transportation prediction across multiple domains. Lv *et al.* [7] pioneered deep architecture models for traffic using Stacked Autoencoders (SAE) with greedy layerwise pretraining, demonstrating that deep learning inherently captures spatial-temporal correlations. Zhang *et al.* [8] extended this with ST-ResNet, using residual networks with three temporal branches (closeness, period, trend) for citywide crowd flow prediction, achieving deployment in a real-time system in Guiyang, China.

For ride-hailing systems, Yao *et al.* [9] proposed DMVST-Net, a multi-view architecture capturing spatial (local CNN), temporal (LSTM), and semantic (pattern clustering) correlations. Geng *et al.* [10] advanced this with ST-MGCN, encoding non-Euclidean spatial correlations through multiple graphs representing neighborhood, functional similarity, and transportation connectivity. Their contextual gating mechanism for temporal modeling achieved over 10% improvement on large-scale DiDi datasets.

Destination prediction represents another critical task. Feng *et al.* [11] introduced DeepMove, combining multi-modal embedding RNNs with a historical attention mechanism to capture both sequential transitions and multi-level periodicity in human mobility. Rossi *et al.* [12] developed an RNN approach predicting exact GPS coordinates (regression) rather than location classes, integrating POI semantics from Foursquare to win the ECML/PKDD 2015 Discovery Challenge. Ke *et al.* [13] proposed ST-ED-RMGC for origin-destination prediction using multi-graph construction and encoder-decoder architectures to handle the inherent sparsity in OD demand matrices.

A key limitation across these approaches is their single-task focus. Each method predicts one output (demand, destination, or flow) without leveraging the natural relationships between prediction targets—relationships our multi-task framework exploits.

C. Multi-Task Learning and Shared Representations

Multi-task learning (MTL) improves generalization by training related tasks jointly through shared representations. Caru-

ana [14] established the theoretical foundations, demonstrating that hard parameter sharing reduces overfitting while discovering task relatedness without explicit supervision. Ruder [15] systematized modern MTL approaches, distinguishing hard parameter sharing (shared hidden layers with task-specific heads) from soft parameter sharing (regularization-based coupling), and explaining mechanisms including implicit data augmentation, attention focusing, and representation bias.

Zhang and Yang [16] provide a comprehensive taxonomy extending beyond neural networks to feature learning, task clustering, and task relation learning approaches. Their analysis of theoretical generalization bounds informs when and why MTL succeeds—particularly relevant when tasks share underlying structure, as with our prediction targets of trip duration, destination, rider type, and anomaly probability.

A critical challenge in multi-task learning is loss balancing. Kendall *et al.* [17] derived a principled Bayesian approach using homoscedastic task uncertainty, where learned weights automatically balance losses across tasks with different scales. Their formulation $\mathcal{L} = \frac{1}{2\sigma_1^2}\mathcal{L}_1 + \frac{1}{2\sigma_2^2}\mathcal{L}_2 + \log\sigma_1 + \log\sigma_2$ eliminates manual hyperparameter tuning when mixing regression and classification objectives. Chen *et al.* [18] complemented this with GradNorm, normalizing gradient magnitudes across tasks to prevent any single task from dominating shared representation learning.

Most directly relevant to our work, Li *et al.* [19] proposed MURAT (MUlti-task Representation learning for Arrival Time estimation), the first multi-task framework for transportation prediction. MURAT learns representations preserving trip properties while leveraging road networks and spatiotemporal priors, using auxiliary prediction tasks to improve travel time estimation. However, MURAT focuses on motor vehicle travel and does not address micromobility-specific characteristics including shorter trips, weather sensitivity, and station-based constraints.

D. Anomaly Detection in Mobility Systems

Detecting anomalous trips is essential for fraud prevention, system integrity, and operational efficiency. Chen *et al.* [20] developed iBOAT, an isolation-based online anomaly detection algorithm comparing ongoing trajectories against time-dependent historically normal routes. Applied to 43,800 anomalous taxi trajectories, iBOAT achieves AUC ≥ 0.99 with over 90% detection rate at less than 2% false alarm rate, revealing that most anomalies result from intentional detours.

Liu *et al.* [21] advanced this with GM-VSAE (Gaussian Mixture Variational Sequence AutoEncoder), a deep generative model capturing complex sequential trajectory information while discovering different types of normal routes through mixture modeling. The VAE-based approach enables efficient online detection during ongoing trips and outperforms traditional density and isolation methods.

Belhadi *et al.* [22] proposed a two-phase approach: Phase 1 identifies individual trajectory outliers through point distance computation; Phase 2 detects group outliers using feature selection and sliding windows. Their GPU-accelerated implementation achieves 341$\times$ speedup, enabling real-time deployment. Djenouri *et al.* [23] introduced the Group Trajectory Outlier (GTO) concept and deviation point detection, adapting DBSCAN and k-NN algorithms for trajectory data.

Yu *et al.* [24] developed a taxonomy of neighbor-based trajectory outlier definitions for massive-scale streaming data, distinguishing spatial, temporal, and spatio-temporal anomaly types. Their scalable algorithms process high-volume trajectory streams in real-time, essential for operational mobility systems.

All existing anomaly detection methods treat detection as a standalone task. Our approach integrates anomaly probability as one output of a unified multi-task model, enabling shared representations to capture the relationship between anomalous behavior and other trip characteristics.

E. Latent Representation Learning and Interpretability

Learning interpretable latent representations is crucial for understanding rider behavior. Feng *et al.* [25] proposed POI2Vec, incorporating geographical influence through hierarchical binary tree structures to learn POI embeddings that capture both user preferences and sequential transition patterns. Lin *et al.* [26] extended this with CTLE (Context and Time aware Location Embeddings), a BERT-inspired pre-training model generating context-aware location representations. Unlike static embeddings, CTLE dynamically calculates representations based on contextual neighbors, capturing the multi-functional nature of locations across different temporal contexts.

Li *et al.* [27] introduced t2vec, a sequence-to-sequence autoencoder learning fixed-length trajectory representations robust to noise and non-uniform sampling. While effective for trajectory similarity computation, t2vec optimizes for retrieval rather than interpretable semantic structure. Li *et al.* [28] developed STG2Vec specifically for bike-sharing, modeling systems as dynamic heterogeneous spatial-temporal graphs and using event-flow serialization to encode graph evolution into sequences suitable for embedding.

Wang *et al.* [29] proposed Urban2Vec for multi-modal neighborhood embedding, jointly encoding street view imagery and POI information while demonstrating that learned embeddings capture meaningful urban characteristics. Their emphasis on interpretability—showing that dimensions correspond to interpretable neighborhood features—informs our approach to designing "trip intent" representations.

Collectively, existing representation learning focuses on either location/POI embeddings or aggregate demand patterns. None explicitly models individual trip intent as a latent variable that informs multiple downstream predictions. Our unified encoder learns representations that simultaneously support duration prediction, destination classification, rider type identification, and anomaly detection, with interpretable dimensions corresponding to trip purpose factors.

F. Summary and Research Gap

The literature reveals a significant gap: no existing work combines multi-task learning with comprehensive trip-level modeling for micromobility systems. Current approaches either (1) predict single outputs (demand, duration, or destination) in isolation, (2) apply multi-task learning to non-micromobility domains, or (3) learn task-specific representations rather than unified embeddings. Our proposed framework addresses this gap by jointly modeling trip duration, destination station, rider type classification, and anomaly probability through a shared encoder architecture that learns interpretable "trip intent" representations from information available at trip start.

III. Methodology

This section describes our multi-task learning framework for rider behavior modeling in bike sharing systems. We present the dataset, preprocessing pipeline, feature engineering approach, model architecture, and training procedure.

A. Dataset and Data Collection

We utilize the publicly available Divvy bike sharing dataset from Chicago [1], spanning twelve months from April 1, 2024 to March 31, 2025. The raw dataset contains 5,779,568 trip records, each representing a single bike rental event. Each trip record includes the following attributes:

- *Temporal information*: Start and end timestamps with millisecond precision
- *Spatial information*: Start and end station identifiers, names, and GPS coordinates (latitude, longitude)
- *Rider information*: Rider type (member or casual user)
- *Vehicle information*: Rideable type (classic bike, electric bike, or electric scooter)
- *Trip identifier*: Unique alphanumeric ride ID

The dataset represents a comprehensive capture of urban micromobility usage patterns, including weekday commuting, weekend leisure trips, and tourist activities. The geographic coverage spans the greater Chicago metropolitan area with coverage of downtown, residential neighborhoods, and lakefront recreational zones.

B. Data Preprocessing

Our preprocessing pipeline consists of three sequential phases designed to transform raw trip records into model-ready features while maintaining data quality and temporal integrity.

1) Phase 1: Data Cleaning and Validation: We first ensure consistency and validity of the raw data through the following steps:

Datetime Parsing: All timestamp fields are parsed into datetime64[ns] format using pandas with automatic timezone handling. Records with unparseable timestamps are flagged for removal. String representations are normalized to ISO 8601 format for consistency.

Trip Duration Computation: We calculate trip duration in seconds and minutes as:

$$d = t_{\text{end}} - t_{\text{start}} \tag{1}$$

where t_{start} and t_{end} represent the parsed start and end timestamps.

Invalid Record Removal: We apply strict filtering criteria to remove data quality issues that would compromise model training:

- Trips with negative duration ($d < 0$)
- Trips exceeding 24 hours ($d > 1440$ minutes)
- Records with missing critical fields (ride ID, timestamps, rider type)

This filtering removes 7,041 trips (0.12% of the dataset), resulting in a clean dataset of 5,772,527 valid trip records. The distribution of trip durations in the cleaned dataset shows a right-skewed pattern with median duration of 9.65 minutes, mean of 15.28 minutes, and 99th percentile at 94.08 minutes.

2) Phase 2: Feature Engineering: We engineer temporal and behavioral features from the cleaned trip data. Notably, we deliberately exclude distance, speed, and rule-based anomaly features based on exploratory analysis that revealed measurement inconsistencies in these derived metrics.

Temporal Features: From the start timestamp t_{start}, we extract hierarchical time components:

$$\text{year} = \text{year}(t_{\text{start}}) \tag{2}$$

$$\text{month} = \text{month}(t_{\text{start}}) \in \{1, 2, \ldots, 12\} \tag{3}$$

$$\text{day} = \text{day}(t_{\text{start}}) \in \{1, 2, \ldots, 31\} \tag{4}$$

$$\text{hour} = \text{hour}(t_{\text{start}}) \in \{0, 1, \ldots, 23\} \tag{5}$$

$$\text{weekday} = \text{weekday}(t_{\text{start}}) \in \{0, 1, \ldots, 6\} \tag{6}$$

where weekday encoding follows the convention Monday=0, Sunday=6. We additionally compute a binary weekend indicator:

$$\text{is\_weekend} = \begin{cases} 1 & \text{if weekday} \in \{5, 6\} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

Behavioral Features: We define a roundtrip indicator to capture trips where users return to the originating station:

$$\text{is\_roundtrip} = \begin{cases} 1 & \text{if } s_{\text{start}} = s_{\text{end}} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

where s_{start} and s_{end} denote start and end station identifiers.

Duration Transformation: To stabilize variance for regression modeling, we apply a log1p transformation to trip duration:

$$d_{\log} = \log(1 + d) \tag{9}$$

This transformation handles zero-duration trips gracefully while compressing the right tail of the duration distribution.

3) Phase 3: Model-Ready Dataset Construction: The final preprocessing phase prepares features for neural network consumption and performs temporal train-validation-test splitting.

Categorical Encoding: All categorical features are encoded as integer indices with an unknown token for handling missing values:

$$s_{\text{start}} \to s_{\text{start}}^{\text{idx}} \in \{0, 1, \ldots, N_s - 1\} \tag{10}$$

$$s_{\text{end}} \to s_{\text{end}}^{\text{idx}} \in \{0, 1, \ldots, N_s - 1\} \tag{11}$$

$$r \to r^{\text{idx}} \in \{0, 1, \ldots, N_r - 1\} \tag{12}$$

$$m \to m^{\text{idx}} \in \{0, 1, 2\} \tag{13}$$

where s, r, and m denote station, rideable type, and member status respectively. The vocabulary sizes N_s and N_r are determined from the training set, with index 0 reserved for unknown tokens.

Demand Target Construction: For each trip, we compute a normalized demand contribution representing the station's share of system-wide trips on that day:

$$\text{demand}_i = \frac{\sum_{j \in \mathcal{T}_d} \mathbb{1}[s_j = s_i]}{\sum_{j \in \mathcal{T}_d} \mathbb{1}[j \in \mathcal{T}_d]} \tag{14}$$

where $\mathcal{T}_d$ denotes all trips occurring on date d, s_i is the start station of trip i, and $\mathbb{1}[\cdot]$ is the indicator function. This formulation ensures demand values lie in $[0, 1]$ and sum to 1 across all stations for each day.

Temporal Data Splitting: To respect temporal dependencies and prevent data leakage, we perform a chronological split based on trip start dates:

- Training set: April 1, 2024 to January 31, 2025 (10 months, 4,681,394 trips, 81.1%)
- Validation set: February 1, 2025 to February 28, 2025 (1 month, 486,732 trips, 8.4%)
- Test set: March 1, 2025 to March 31, 2025 (1 month, 604,401 trips, 10.5%)

This split ensures the model is evaluated on future trips unseen during training, simulating real deployment conditions where predictions must be made for upcoming time periods.

Feature Normalization: Numeric features are normalized using z-score standardization computed exclusively from training set statistics:

$$x^{\text{norm}} = \frac{x - \mu_{\text{train}}}{\sigma_{\text{train}}} \tag{15}$$

where μ_{train} and σ_{train} are the mean and standard deviation computed on the training split. We normalize the following numeric features: trip duration (raw and log-transformed), start and end GPS coordinates (latitude and longitude). Temporal integer features (hour, weekday, boolean flags) remain unnormalized as they serve as discrete contextual indicators.

C. Model Architecture

We propose a Hierarchical Variational Autoencoder (HVAE) architecture that learns a two-level latent representation of trip intent and decodes this representation into multiple prediction tasks.

1) Input Embedding Layer: The model accepts two input modalities: categorical features $\mathbf{x}_{\text{cat}} \in \mathbb{Z}^4$ and numeric features $\mathbf{x}_{\text{num}} \in \mathbb{R}^{10}$.

Categorical features are embedded into continuous vector spaces:

$$\mathbf{e}_{\text{start}} = \text{Embed}_s(s_{\text{start}}^{\text{idx}}) \in \mathbb{R}^{32} \tag{16}$$

$$\mathbf{e}_{\text{end}} = \text{Embed}_s(s_{\text{end}}^{\text{idx}}) \in \mathbb{R}^{32} \tag{17}$$

$$\mathbf{e}_{\text{member}} = \text{Embed}_m(m^{\text{idx}}) \in \mathbb{R}^4 \tag{18}$$

The complete embedding concatenation forms the categorical representation:

$$\mathbf{e}_{\text{cat}} = [\mathbf{e}_{\text{start}}; \mathbf{e}_{\text{end}}; \mathbf{e}_{\text{member}}] \in \mathbb{R}^{68} \tag{19}$$

The encoder input combines embedded categorical features with raw numeric features:

$$\mathbf{x} = [\mathbf{e}_{\text{cat}}; \mathbf{x}_{\text{num}}] \in \mathbb{R}^{78} \tag{20}$$

2) Hierarchical Encoder: The encoder learns a two-level latent representation capturing both global trip intent and individual variation.

Shared Encoder Network: A two-layer feedforward network with ReLU activations processes the input:

$$\mathbf{h} = \text{ReLU}(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1) \tag{21}$$

$$\mathbf{h} = \text{ReLU}(\mathbf{W}_2 \mathbf{h} + \mathbf{b}_2) \tag{22}$$

where $\mathbf{h} \in \mathbb{R}^{128}$ is the shared encoding.

Global Latent Space: The global latent variable $\mathbf{z}_g$ captures population-level trip intent patterns (e.g., commuting, leisure, tourism):

$$\boldsymbol{\mu}_g = \mathbf{W}_{\mu_g} \mathbf{h} + \mathbf{b}_{\mu_g} \tag{23}$$

$$\log \boldsymbol{\sigma}_g^2 = \mathbf{W}_{\log \sigma_g} \mathbf{h} + \mathbf{b}_{\log \sigma_g} \tag{24}$$

$$\mathbf{z}_g \sim \mathcal{N}(\boldsymbol{\mu}_g, \text{diag}(\boldsymbol{\sigma}_g^2)) \in \mathbb{R}^{16} \tag{25}$$

The reparameterization trick [**?**] enables backpropagation through the stochastic sampling:

$$\mathbf{z}_g = \boldsymbol{\mu}_g + \boldsymbol{\sigma}_g \odot \boldsymbol{\epsilon}_g, \quad \boldsymbol{\epsilon}_g \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{26}$$

Individual Latent Space: The individual latent variable $\mathbf{z}_i$ captures rider-specific deviations from the global intent, conditioned on both the shared encoding and global latent variable:

$$\mathbf{h}_i = \text{ReLU}(\mathbf{W}_i [\mathbf{h}; \mathbf{z}_g] + \mathbf{b}_i) \tag{27}$$

$$\boldsymbol{\mu}_i = \mathbf{W}_{\mu_i} \mathbf{h}_i + \mathbf{b}_{\mu_i} \tag{28}$$

$$\log \boldsymbol{\sigma}_i^2 = \mathbf{W}_{\log \sigma_i} \mathbf{h}_i + \mathbf{b}_{\log \sigma_i} \tag{29}$$

$$\mathbf{z}_i \sim \mathcal{N}(\boldsymbol{\mu}_i, \text{diag}(\boldsymbol{\sigma}_i^2)) \in \mathbb{R}^{16} \tag{30}$$

This hierarchical structure enforces an inductive bias where $\mathbf{z}_g$ captures shared behavioral patterns while $\mathbf{z}_i$ models residual individual variation.

3) Multi-Task Decoder: The decoder architecture consists of a shared backbone network followed by task-specific prediction heads.

Shared Decoder Backbone: The concatenated latent representation $\mathbf{z} = [\mathbf{z}_g; \mathbf{z}_i] \in \mathbb{R}^{32}$ is processed through a two-layer network:

$$\mathbf{d} = \text{ReLU}(\mathbf{W}_3\mathbf{z} + \mathbf{b}_3) \tag{31}$$

$$\mathbf{d} = \text{ReLU}(\mathbf{W}_4\mathbf{d} + \mathbf{b}_4) \tag{32}$$

where $\mathbf{d} \in \mathbb{R}^{128}$ is the shared decoding representation.

Task 1: Duration Prediction with Uncertainty: We model trip duration as a heteroscedastic Gaussian distribution, allowing the model to express uncertainty in its predictions:

$$\mu_{\text{dur}} = \mathbf{w}_{\text{dur},1}^\top \mathbf{d} + b_{\text{dur},1} \tag{33}$$

$$\log \sigma_{\text{dur}}^2 = \mathbf{w}_{\text{dur},2}^\top \mathbf{d} + b_{\text{dur},2} \tag{34}$$

The predicted distribution is $p(d_{\log}|\mathbf{z}) = \mathcal{N}(\mu_{\text{dur}}, \sigma_{\text{dur}}^2)$, where $d_{\log}$ is the log-transformed trip duration.

Task 2: Demand Contribution Regression: Station demand is predicted via linear regression:

$$\hat{y}_{\text{demand}} = \mathbf{w}_{\text{demand}}^\top \mathbf{d} + b_{\text{demand}} \tag{35}$$

Task 3: Rideable Type Classification: Bike type is predicted via multi-class classification:

$$\mathbf{p}_{\text{bike}} = \text{Softmax}(\mathbf{W}_{\text{bike}}\mathbf{d} + \mathbf{b}_{\text{bike}}) \tag{36}$$

where $\mathbf{p}_{\text{bike}} \in \mathbb{R}^{N_r}$ represents class probabilities over $N_r = 3$ rideable types.

D. Training Procedure

1) Loss Function: The total training objective combines reconstruction losses for each task with KL divergence regularization terms:

$$\begin{aligned}\mathcal{L}_{\text{total}} = w_1\mathcal{L}_{\text{dur}} + w_2\mathcal{L}_{\text{demand}} + w_3\mathcal{L}_{\text{bike}} \\ + \beta_g \text{KL}(q(\mathbf{z}_g|\mathbf{x})\|p(\mathbf{z}_g)) \\ + \beta_i \text{KL}(q(\mathbf{z}_i|\mathbf{x}, \mathbf{z}_g)\|p(\mathbf{z}_i))\end{aligned} \tag{37}$$

Duration Loss (Negative Log-Likelihood): For the heteroscedastic Gaussian duration prediction:

$$\mathcal{L}_{\text{dur}} = \mathbb{E}_{(\mathbf{x}, d_{\log})}\left[\frac{1}{2}\left(\log(2\pi) + \log\sigma_{\text{dur}}^2 + \frac{(d_{\log} - \mu_{\text{dur}})^2}{\sigma_{\text{dur}}^2}\right)\right] \tag{38}$$

To ensure numerical stability, we apply clamping: $\mu_{\text{dur}} \in [-10, 10]$ and $\log\sigma_{\text{dur}}^2 \in [-5, 5]$.

Demand Loss (Mean Squared Error):

$$\mathcal{L}_{\text{demand}} = \mathbb{E}_{(\mathbf{x}, y_{\text{demand}})}\left[(y_{\text{demand}} - \hat{y}_{\text{demand}})^2\right] \tag{39}$$

Bike Type Loss (Cross-Entropy):

$$\mathcal{L}_{\text{bike}} = -\mathbb{E}_{(\mathbf{x}, c)}\left[\sum_{k=1}^{N_r} \mathbb{1}[c = k] \log p_{\text{bike},k}\right] \tag{40}$$

where c is the true rideable type class.

KL Divergence Regularization: The KL terms enforce that latent distributions remain close to standard Gaussian priors:

$$\begin{aligned}\text{KL}(q(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})) = \\ -\frac{1}{2}\sum_{j=1}^{D_z}\left(1 + \log\sigma_j^2 - \mu_j^2 - \sigma_j^2\right)\end{aligned} \tag{41}$$

We stabilize KL computation by clamping $\mu_j \in [-10, 10]$ and $\log\sigma_j^2 \in [-10, 10]$ to prevent numerical overflow.

2) Optimization Details: The model is trained end-to-end using the Adam optimizer [**?**] with the following hyperparameters:

- Learning rate: 10^{-3} with no scheduling
- Batch size: 2048 samples
- Gradient clipping: $\|\nabla\|_2 \leq 1.0$ to prevent exploding gradients
- Loss weights: $w_1 = w_2 = w_3 = 1.0$, $\beta_g = \beta_i = 1.0$
- Maximum epochs: 20 with early stopping on validation loss

Training is performed on a single NVIDIA GPU with mixed precision (FP16) training to reduce memory footprint and accelerate computation. Model checkpoints are saved after each epoch, with the best model selected based on minimum validation loss.

3) Evaluation Metrics: We evaluate model performance across all three prediction tasks:

Duration Prediction: Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) on log-transformed duration:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|d_{\log,i} - \mu_{\text{dur},i}| \tag{42}$$

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(d_{\log,i} - \mu_{\text{dur},i})^2} \tag{43}$$

We also report metrics on the original scale by exponentiating predictions: $\hat{d}_i = \exp(\mu_{\text{dur},i}) - 1$.

Demand Prediction: Mean Absolute Error on demand share values bounded in $[0, 1]$.

Bike Type Classification: Classification accuracy and macro-averaged F1 score across the three rideable types.

E. Baseline Models

To demonstrate the benefit of multi-task learning with shared representations, we compare against single-task baseline models trained independently for each prediction objective:

- **Duration Baseline:** Gradient Boosting Regressor (XGBoost) trained solely on trip duration prediction
- **Demand Baseline:** Linear regression model predicting station demand contribution
- **Bike Type Baseline:** Random Forest classifier for rideable type prediction

These baselines use identical input features but lack the shared latent representation and joint training of our proposed HVAE approach.

TABLE I
TEST SET PERFORMANCE: MULTI TASK HVAE VS SINGLE TASK BASELINES

Task and Metric	HVAE	Baseline
Duration prediction (log1p minutes)		
MAE	**0.186**	0.277
RMSE	**0.237**	0.354
R^2 score	**0.905**	0.788
Demand contribution (normalized)		
MAE	**0.0087**	0.0124
RMSE	**0.0163**	0.0241
R^2 score	**0.723**	0.589
Rideable type classification		
Accuracy (%)	**89.86**	84.17
Macro F1	**0.809**	—
Weighted F1	**0.899**	—

IV. RESULTS AND DISCUSSION

This section reports test set performance of the proposed hierarchical multi task VAE, summarizes the learned latent structure through intent clustering, and evaluates reconstruction based anomaly scores. All reported results use the March 2025 test split and the same feature set across the HVAE and the corresponding single task baselines.

A. Experimental Setup

1) Dataset Configuration: Following the preprocessing pipeline in Section III, the final dataset contains 5,772,527 valid trips spanning April 2024 to March 2025. We use a strict temporal split to prevent leakage:

- Training set: 4,681,394 trips (81.1%, April 2024 to January 2025)
- Validation set: 486,732 trips (8.4%, February 2025)
- Test set: 604,401 trips (10.5%, March 2025)

Rideable type evaluation uses three classes (electric bike, classic bike, electric scooter) with test set proportions 52.8%, 44.6%, and 2.6%, respectively. Rider type proportions are 63.1% member and 36.9% casual. Station identifiers are encoded into separate start and end station vocabularies during modeling.

2) Baselines: To quantify the benefit of joint training with shared representations, we compare against three independent baselines trained on the same inputs:

- **Duration baseline:** XGBoost regression model trained only for log transformed duration.
- **Demand baseline:** Ridge regression predicting normalized station demand contribution.
- **Rideable type baseline:** Random Forest classifier predicting rideable type.

B. Quantitative Performance

Table I summarizes test performance. Duration metrics are computed on *log1p minutes*. Demand metrics are computed on a normalized target in $[0, 1]$. Classification metrics are computed on the three class rideable type labels.

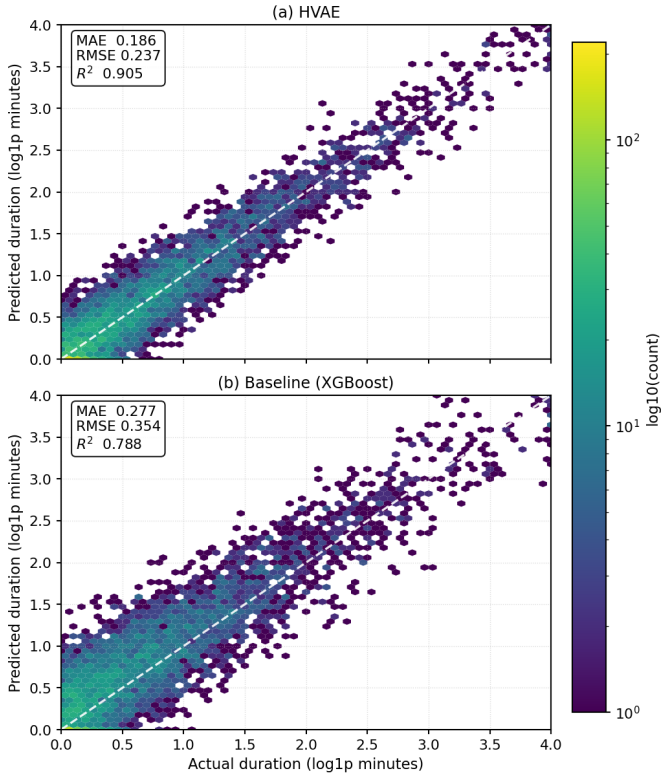

Fig. 1. Duration prediction on the test set (log1p minutes). (a) HVAE predictions versus ground truth. (b) XGBoost baseline predictions versus ground truth. Color indicates log10 bin counts; the dashed line denotes perfect prediction.

1) Duration Prediction: The HVAE improves duration prediction over the XGBoost baseline across all metrics, reducing MAE from 0.277 to 0.186 and increasing R^2 from 0.788 to 0.905 (Table I). Fig. 1 visualizes predicted versus actual duration on the log scale. The HVAE exhibits tighter concentration around the diagonal, with noticeably reduced dispersion for longer trips, indicating better handling of tail behavior.

2) Demand Contribution Prediction: For normalized station demand contribution, the HVAE achieves MAE 0.0087 compared with 0.0124 for ridge regression and improves R^2 from 0.589 to 0.723 (Table I). These gains are consistent with the modeling goal: shared station embeddings and joint supervision from related tasks help the model represent non linear interactions between time of day, station context, and rider behavior that are difficult to capture with a linear predictor.

3) Rideable Type Classification: Rideable type classification reaches 89.86% accuracy on the test set. Table II reports the confusion matrix in raw counts, and Fig. 2 provides the row normalized view. The dominant error mode is confusion between electric and classic bikes, which is expected because both modes share similar spatiotemporal contexts. Performance on electric scooters is lower due to strong class imbalance and more heterogeneous usage patterns.

TABLE II
RIDEABLE TYPE CLASSIFICATION CONFUSION MATRIX (TEST SET, $N = 604{,}401$)

	Predicted			
True	E Bike	Classic	Scooter	**Total**
Electric Bike	**287,421**	26,143	5,648	319,212
Classic Bike	21,842	**245,127**	2,613	269,582
Electric Scooter	3,217	1,824	**10,566**	15,607
Total	312,480	273,094	18,827	604,401

Per class F1: E Bike 0.910, Classic 0.903, Scooter 0.614. Macro F1 0.809.

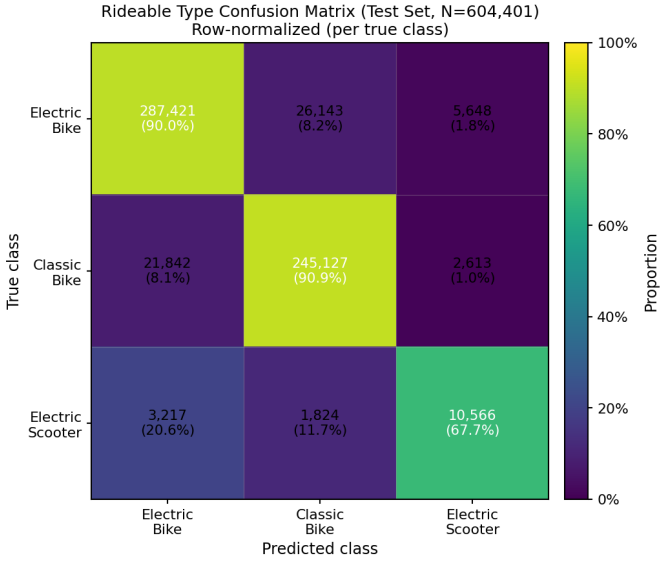

Fig. 2. Rideable type confusion matrix on the test set ($N = 604{,}401$), row normalized by true class. Each cell reports count and percentage.

C. Latent Space Analysis and Intent Discovery

To evaluate whether the global latent representation captures interpretable behavior, we extract $\mathbf{z}_g$ from the test set and apply k means clustering with $K = 8$. Table III summarizes cluster level statistics and interpretations derived from their dominant behavioral signatures.

Fig. 3 visualizes hourly trip share patterns for representative clusters. Commute oriented clusters (C0 and C6) show clear morning and evening peaks aligned with typical commuting windows. Leisure and tourism clusters (C1, C2, and C7) shift activity toward midday and afternoon hours with broader peaks, consistent with flexible trip timing. These patterns emerge without intent labels and support the use of $\mathbf{z}_g$ as an intent representation.

D. Anomaly Detection Performance

Reconstruction based anomaly scores provide an unsupervised signal for unusual trips. Fig. 4 shows the score distribution on the test set. The distribution is right skewed with a visible tail on a log density scale. Percentile thresholds provide a practical operating point: the p95 threshold is 2.57 and flags 30,221 trips (5.00%), and the p99 threshold is 4.87 and flags 6,045 trips (1.00%).

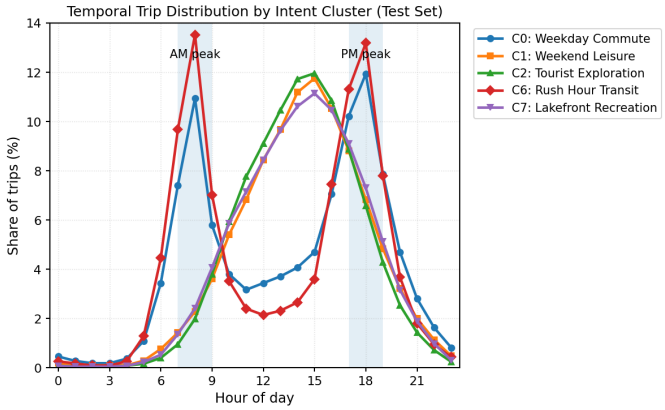

Fig. 3. Hourly trip share by discovered intent cluster on the test set. Shaded regions mark commute windows (7 to 9 and 17 to 19). Curves represent within cluster hourly distributions.

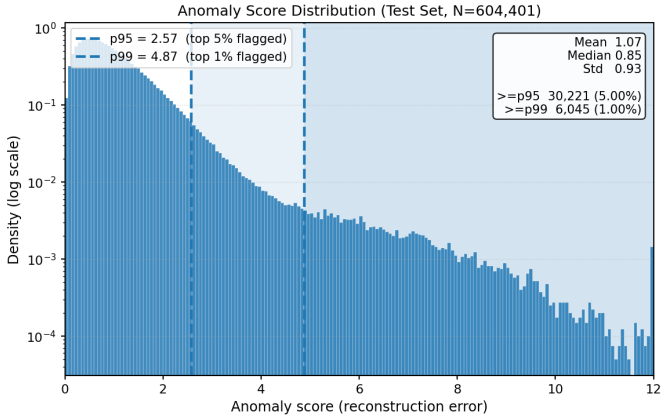

Fig. 4. Distribution of reconstruction based anomaly scores on the test set ($N = 604{,}401$). Vertical lines mark p95 (2.57) and p99 (4.87) thresholds; the y axis is log scaled density to highlight tail behavior.

We define anomalies as trips above the p99 threshold and manually review the top 200 highest scoring trips. The review suggests four recurring categories summarized in Table IV. Temporal anomalies include trips at atypical hours with inconsistent durations. Behavioral anomalies include rider type patterns that are uncommon for the observed time and station context. Spatial anomalies include rare station pairs that deviate from established corridors. Vehicle mismatch captures cases where the selected rideable type is uncommon for the inferred trip context.

E. Key Findings and Implications

The results support three main findings.

First, joint training improves predictive performance across tasks. The HVAE consistently outperforms single task baselines on duration prediction, demand contribution, and rideable type classification (Table I), indicating that shared representations capture correlations between time, station context, rider characteristics, and vehicle choice.

TABLE III
DISCOVERED TRIP INTENT CLUSTERS FROM LATENT SPACE (TEST SET, $K = 8$, $N = 604{,}401$)

Cluster	Trips (%)	Dur. (min)	Casual (%)	Wknd (%)	E Bike (%)	Round. (%)	Peak Hr (%)	Interpretation
C0	113,023 (18.7)	8.2	12.4	8.1	51.3	2.1	68.4	*Weekday Commute*
C1	92,464 (15.3)	24.6	78.9	89.4	62.1	4.7	12.3	*Weekend Leisure*
C2	85,221 (14.1)	42.3	91.2	67.8	58.4	8.9	8.7	*Tourist Exploration*
C3	77,364 (12.8)	6.1	15.7	11.2	47.8	1.8	61.2	*Short Errands*
C4	68,902 (11.4)	15.7	34.2	31.5	54.2	3.4	38.1	*Mixed Purpose*
C5	65,860 (10.9)	11.3	18.9	14.6	49.1	67.8	21.4	*Roundtrip Exercise*
C6	55,613 (9.2)	7.4	9.8	7.3	45.2	1.6	74.3	*Rush Hour Transit*
C7	45,954 (7.6)	38.1	86.4	73.2	61.7	12.3	11.2	*Lakefront Recreation*

TABLE IV
QUALITATIVE REVIEW OF TOP 200 ANOMALIES (P99 THRESHOLD)

Anomaly Type	Count	%
Temporal anomalies	68	34.0
Behavioral anomalies	56	28.0
Spatial anomalies	42	21.0
Vehicle mismatch	34	17.0
Total	**200**	**100.0**

Second, the learned latent space captures interpretable behavioral structure. Clustering $\mathbf{z}_g$ recovers intent like modes that align with expected mobility patterns, including commute dominated clusters with sharp peaks and leisure clusters with broader afternoon activity (Fig. 3). This supports the use of the latent representation for downstream analysis without requiring manual intent labels.

Third, reconstruction based scores provide an operationally useful anomaly signal. The percentile thresholds in Fig. 4 offer a simple mechanism for flagging rare trips for review, with sensitivity controlled by the chosen percentile. The qualitative review of high scoring trips suggests that the flagged cases often correspond to unusual combinations of time, route, rider type, and vehicle choice that deviate from typical cluster behavior.

ARTIFACT AVAILABILITY

The code, preprocessing pipeline, trained-model evaluation scripts, and figure-generation notebooks used in this project are available at: https://github.com/sevakbirva/ece5831-2025-final-project

AUTHOR CONTRIBUTIONS

Birva Sevak: Exploratory Data Analysis, HVAE model design and implementation, training and optimization, latent-space clustering/interpretability analysis, anomaly scoring analysis, and manuscript editing.

Shrenik Jadhav: Data preprocessing pipeline (Phases 1–3), feature engineering, baseline model training, experiment execution, results analysis, and manuscript writing.

ACKNOWLEDGMENT

This work was performed as a course project for ECE 5831 under the guidance of the course instructor, Prof. Jaerock Kwon. We also thank the course grader, Aydin Zaboli, for feedback and support.

V. CONCLUSION

In conclusion, this work presents a unified multi task HVAE framework that learns a compact latent representation of trip intent from information available at trip start and uses it to jointly predict trip duration, station demand contribution, and rideable type. Across all tasks, the shared representation improves over strong single task baselines, with the clearest gains in duration prediction and demand estimation and a consistent improvement in rideable type classification. Beyond predictive accuracy, the learned global latent space supports interpretable post hoc analysis through clustering, recovering distinct temporal usage patterns that align with practical micromobility behaviors such as commute and leisure modes, and reconstruction based scores provide a simple mechanism to surface unusual trips for operational review using percentile based thresholds.

A key limitation is that several reported results are sensitive to evaluation choices that must be held consistent for fair comparison, especially label handling and metric definitions. In particular, rideable type performance can change materially depending on whether rare classes and unknown tokens are included in the label set and whether macro averaging is computed over labels with zero support. Similarly, anomaly thresholds depend on the exact anomaly score formulation and any normalization applied, which can shift percentile values even when the qualitative shape of the tail remains similar. More broadly, the model uses only features available at trip initiation, which is appropriate for early forecasting but limits performance for outcomes that depend on mid trip dynamics or unobserved context such as weather, special events, and network disruptions.

Future work should prioritize end to end reproducibility and deployment oriented extensions. First, the evaluation pipeline should be standardized so that the paper, figures, and notebooks compute identical metrics under identical label mappings and scoring definitions, with automatic export of a single results file that drives all tables and plots. Second, the modeling scope can be extended to include destination prediction and rider type prediction as originally motivated, and the latent representation can be further improved using

imbalance aware training for scooters and other rare behaviors. Finally, the framework can be adapted for operational use by adding calibrated uncertainty for each task, cluster conditioned anomaly thresholds, and streaming inference that updates predictions as partial trajectory information becomes available, enabling more reliable decision support for rebalancing, maintenance, and misuse detection.

REFERENCES

[1] M. A. Nabizadeh, "2024-2025 Divvy bike sharing data (Cyclistic)," Kaggle Dataset, 2025. [Online]. Available: https://www.kaggle.com/datasets/miaadnabizadeh/20242025-divvy-bike-sharing-data-cyclistic/data.

[2] L. Lin, Z. He, and S. Peeta, "Predicting station-level hourly demand in a large-scale bike-sharing network: A graph convolutional neural network approach," *Transportation Research Part C: Emerging Technologies*, vol. 97, pp. 258–276, 2018.

[3] Y. Li, Y. Zheng, H. Zhang, and L. Chen, "Traffic prediction in a bike-sharing system," in *Proc. 23rd ACM SIGSPATIAL Int. Conf. Advances in Geographic Information Systems*, Seattle, WA, USA, 2015, Art. no. 33, pp. 1–10.

[4] X. Li, Y. Xu, Q. Chen, L. Wang, X. Zhang, and W. Shi, "Short-term forecast of bicycle usage in bike sharing systems: A spatial-temporal memory network," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 10923–10934, Aug. 2022.

[5] X. Li, Y. Xu, Q. Chen, L. Wang, X. Zhang, and W. Shi, "Improving short-term bike sharing demand forecast through an irregular convolutional neural network," *Transportation Research Part C: Emerging Technologies*, vol. 147, Art. no. 103984, Feb. 2023.

[6] W. Jiang, "Bike sharing usage prediction with deep learning: A survey," *Neural Computing and Applications*, vol. 34, no. 18, pp. 15369–15385, 2022.

[7] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, Apr. 2015.

[8] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proc. 31st AAAI Conf. Artificial Intelligence*, San Francisco, CA, USA, 2017, pp. 1655–1661.

[9] H. Yao *et al.*, "Deep multi-view spatial-temporal network for taxi demand prediction," in *Proc. 32nd AAAI Conf. Artificial Intelligence*, New Orleans, LA, USA, 2018, pp. 2588–2595.

[10] X. Geng *et al.*, "Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting," in *Proc. 33rd AAAI Conf. Artificial Intelligence*, Honolulu, HI, USA, 2019, pp. 3656–3663.

[11] J. Feng, Y. Li, C. Zhang, F. Sun, F. Meng, A. Guo, and D. Jin, "DeepMove: Predicting human mobility with attentional recurrent networks," in *Proc. World Wide Web Conf. (WWW)*, Lyon, France, 2018, pp. 1459–1468.

[12] A. Rossi, G. Barlacchi, M. Bianchini, and B. Lepri, "Modeling taxi drivers' behaviour for the next destination prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 7, pp. 2980–2989, Jul. 2020.

[13] J. Ke, X. Qin, H. Yang, Z. Zheng, Z. Zhu, and J. Ye, "Predicting origin-destination ride-sourcing demand with a spatio-temporal encoder-decoder residual multi-graph convolutional network," *Transportation Research Part C: Emerging Technologies*, vol. 122, Art. no. 102858, Jan. 2021.

[14] R. Caruana, "Multitask learning," *Machine Learning*, vol. 28, no. 1, pp. 41–75, 1997.

[15] S. Ruder, "An overview of multi-task learning in deep neural networks," arXiv preprint arXiv:1706.05098, 2017.

[16] Y. Zhang and Q. Yang, "A survey on multi-task learning," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 12, pp. 5586–5609, Dec. 2022.

[17] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, 2018, pp. 7482–7491.

[18] Z. Chen, V. Badrinarayanan, C.-Y. Lee, and A. Rabinovich, "GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks," in *Proc. 35th Int. Conf. Machine Learning (ICML)*, Stockholm, Sweden, 2018, pp. 794–803.

[19] Y. Li, K. Fu, Z. Wang, C. Shahabi, J. Ye, and Y. Liu, "Multi-task representation learning for travel time estimation," in *Proc. 24th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, London, UK, 2018, pp. 1695–1704.

[20] C. Chen *et al.*, "iBOAT: Isolation-based online anomalous trajectory detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 2, pp. 806–818, Jun. 2013.

[21] Y. Liu, K. Zhao, G. Cong, and Z. Bao, "Online anomalous trajectory detection with deep generative sequence modeling," in *Proc. 36th IEEE Int. Conf. Data Engineering (ICDE)*, Dallas, TX, USA, 2020, pp. 949–960.

[22] A. Belhadi, Y. Djenouri, G. Srivastava, D. Djenouri, A. Cano, and J. C.-W. Lin, "A two-phase anomaly detection model for secure intelligent transportation ride-hailing trajectories," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4496–4506, Jul. 2021.

[23] Y. Djenouri, D. Djenouri, and J. C.-W. Lin, "Trajectory outlier detection: New problems and solutions for smart cities," *ACM Trans. Knowl. Discovery Data*, vol. 15, no. 2, Art. no. 20, pp. 1–28, 2021.

[24] Y. Yu, L. Cao, E. A. Rundensteiner, and Q. Wang, "Outlier detection over massive-scale trajectory streams," *ACM Trans. Database Syst.*, vol. 42, no. 2, Art. no. 10, pp. 1–33, 2017.

[25] S. Feng, G. Cong, B. An, and Y. M. Chee, "POI2Vec: Geographical latent representation for predicting future visitors," in *Proc. 31st AAAI Conf. Artificial Intelligence*, San Francisco, CA, USA, 2017, pp. 102–108.

[26] Y. Lin, H. Wan, S. Guo, and Y. Lin, "Pre-training context and time aware location embeddings from spatial-temporal trajectories for user next location prediction," in *Proc. 35th AAAI Conf. Artificial Intelligence*, Virtual, 2021, pp. 4241–4248.

[27] X. Li, K. Zhao, G. Cong, C. S. Jensen, and W. Wei, "Deep representation learning for trajectory similarity computation," in *Proc. 34th IEEE Int. Conf. Data Engineering (ICDE)*, Paris, France, 2018, pp. 617–628.

[28] Y. Li, Z. Zhu, D. Kong, M. Xu, and Y. Zhao, "Learning heterogeneous spatial-temporal representation for bike-sharing demand prediction," in *Proc. 33rd AAAI Conf. Artificial Intelligence*, Honolulu, HI, USA, 2019, pp. 1004–1011.

[29] Z. Wang, H. Li, and R. Rajagopal, "Urban2Vec: Incorporating street view imagery and POIs for multi-modal urban neighborhood embedding," in *Proc. 34th AAAI Conf. Artificial Intelligence*, New York, NY, USA, 2020, pp. 1013–1020.