International Journal of Geographical Information Science
Vol. 00, No. 00, Month 2025, 1–35

Detecting Transportation Mode Using Dense Smartphone GPS Trajectories and Transformer Models

Yuandong Zhang[1,†], Othmane Echchabi[2,3,†], Tianshu Feng[4], Wenyi Zhang[5], Hsuai-Kai Liao[5], and Charles Chang[5,*]

[1]University of California, San Diego, USA
[2]Mila - Quebec AI Institute
[3]McGill University, Canada
[4]University of Pennsylvania, USA
[5]Duke Kunshan University, Kunshan, China

[†]These authors contributed equally to this work.
(*Received 00 Month 200x; final version received 00 Month 200x*)

Transportation mode detection is an important topic within GeoAI and transportation research. In this study, we introduce SpeedTransformer, a novel Transformer-based model that relies solely on speed inputs to infer transportation modes from dense smartphone GPS trajectories. In benchmark experiments, SpeedTransformer outperformed traditional deep learning models, such as the Long Short-Term Memory (LSTM) network. Moreover, the model demonstrated strong flexibility in transfer learning, achieving high accuracy across geographical regions after fine-tuning with small sample subsets. Finally, we deployed the model in a real-world experiment, where it consistently outperformed baseline models under complex conditions and high data uncertainty. These findings suggest that Transformer architectures, when combined with dense GPS trajectories, hold substantial potential for advancing transportation mode detection and broader mobility-related research.

Keywords: GeoAI; Transformer; dense GPS trajectory; deep learning; real-world experiment; smartphone app design

[*]Corresponding author. Email: charles.c.chang@dukekunshan.edu.cn

ISSN: 1365-8816 print/ISSN 1362-3087 online

DOI: 10.1080/1365881YYxxxxxxxx
http://www.informaworld.com

1. Introduction

The study of human mobility patterns—how individuals move across space—has become an important focus across geography, transportation science, public health, and climate change science (Gonzalez *et al.* 2008, Schuessler and Axhausen 2009, Yao *et al.* 2020, Shaw *et al.* 2016, Bonaccorsi *et al.* 2020, McMichael 2020, Barbosa *et al.* 2021, Zook *et al.* 2015, Guo *et al.* 2015, Tao *et al.* 2018). One key aspect of human mobility is transportation mode choice, the accurate estimation of which is essential for understanding individual carbon emissions and associated health benefits (Girod *et al.* 2013, Tajalli and Hajbabaie 2017).

Traditionally, transportation surveys were used to estimate individual's choices of transportation mode (Wang *et al.* 2015). However, two major technological advancements in recent years have reshaped transportation mode detection. First, smartphones—equipped with GPS, accelerometers, gyroscopes, and cellular network connectivity—have enabled the creation of detailed mobility datasets derived from mobile applications (Molloy *et al.* 2023), social media platforms (Preotiuc-Pietro and Cohn 2013), travel cards (Gordon *et al.* 2013), and cellphone signals (Lu *et al.* 2012). These datasets substantially surpassed traditional transportation surveys in dimensionality, accuracy, variety, and volume (Barbosa *et al.* 2015, Goodchild 2013).

Second, advancements in machine learning (ML) have greatly improved the ability to extract meaningful information from mobility datasets. ML techniques now range from ensemble-based models such as Random Forests (Breiman 2001) to deep learning architectures including Convolutional Neural Networks (CNNs) (LeCun and Bengio 1998) and Long Short-Term Memory (LSTM) networks (Hochreiter and Schmidhuber 1997). These approaches have leveraged the increasing spatiotemporal granularity of human mobility data, substantially improving the accuracy of transportation mode prediction (Pappalardo *et al.* 2023).

Nevertheless, significant challenges remained. Smartphone-derived mobility datasets, while highly granular, often exhibited inconsistent quality due to the absence of standardized data collection protocols and inaccuracies in geographic information, complicating analytical modeling. Model performance frequently depended on derived features such as acceleration, while complex preprocessing tasks—such as geocoding and geotagging—further increased data uncertainty. Moreover, extensive feature engineering used to aggregate raw GPS trajectories often resulted in the loss of critical sequential information essential for accurate mobility modeling (Jahangiri and Rakha 2015).

Privacy posed another major challenge. Mobility applications commonly collected sensitive information, including geographic locations, trip details (e.g., start and end times), and personal or device identifiers. For example, Xu *et al.* (2017) demonstrated that even anonymized mobility data could be re-identified with 73–91% accuracy, underscoring the difficulty of ensuring privacy protection. Such risks frequently discouraged individuals from sharing their mobility data (Rzeszewski and Luczys 2018), constraining data availability.

Moreover, mobility models have often lacked generalizability across geographic regions, particularly when trained on isolated or geographically homogeneous datasets. Human mobility behaviors vary widely across countries and regions due to differences in infrastructure, speed regulations, and cultural norms. Nevertheless, most studies have assessed model performance using train–test splits from the same dataset (Zhao *et al.* 2016), which fail to capture real-world adaptability. Models fine-tuned on benchmarks such as Geolife (Zheng *et al.* 2010) frequently required substantial recalibration when applied to other

contexts. The proliferation of deep learning frameworks, hyperparameter tuning strategies, and architectural variations has further complicated cross-regional reproducibility in mobility modeling research (Graser *et al.* 2024).

Finally, existing mobility models have often struggled to perform reliably under real-world conditions, which are considerably more unpredictable and complex than curated research datasets suggest. Everyday travel involves nuances such as short trips and multimodal journeys that traditional models frequently misclassified (Pappalardo *et al.* 2023). Moreover, GPS signal quality varied across smartphone models and was highly sensitive to environmental factors, including urban canyons and signal obstructions (Zandbergen 2009, Cui and Ge 2001). These real-world data inconsistencies—typically underrepresented in benchmark datasets—underscore the need to validate mobility models under realistic conditions that reflect the full variability of human movement.

To address these challenges, we introduced a Transformer-based deep learning model that uses a simple input—instantaneous speed—to achieve highly accurate transportation mode detection. We refer to this model as SpeedTransformer. Using transportation mode classification as a case study, we demonstrate the model's capability and versatility. Our contributions are threefold. First, we show that a Transformer-based neural network can achieve state-of-the-art performance using only speed as input. By leveraging positional encoding and multi-head attention, our model captured complex temporal patterns without elaborate feature engineering, thereby reducing both privacy concerns and computational demands. Second, we demonstrate robust cross-regional generalizability through transfer learning: a model pre-trained on data primarily collected in Switzerland maintained exceptional performance when fine-tuned on samples from Beijing. Finally, we validated our approach under real-world conditions by developing a novel smartphone mini-program and recruiting 348 participants for a three-day field experiment. The experimental results confirmed that our model's advantages translated effectively from controlled environments to practical applications characterized by real-world uncertainty and variability.

2. Related Work

Transportation mode detection represents a central dimension of human mobility research, with downstream applications ranging from carbon footprint estimation (Manzoni *et al.* 2010) and tourism recommendations (Xiao *et al.* 2025) to traffic management (Prelipcean *et al.* 2016) and public health interventions (Aleta *et al.* 2022). With the proliferation of GPS-enabled devices and advances in machine learning, the field has undergone substantial progress in recent years, moving beyond traditional social scientific methods such as transportation surveys and adopting data-driven methodologies grounded in machine learning. Notable developments have emerged in both classical machine learning algorithms and deep learning approaches.

2.1. *Machine Learning for Transportation Mode Detection and its Challenges*

Classical machine learning (ML) algorithms formed the foundation of early data-driven approaches to transportation mode detection. These methods typically converted sequential GPS trajectories into statistical, tabular representations, with notable implementations including Decision Trees (Zheng *et al.* 2008), Random Forests (Stenneth *et al.*

2011, Jahangiri and Rakha 2015), and Support Vector Machines (Bolbol *et al.* 2012a). Stenneth *et al.* (2011) demonstrated the effectiveness of these models through systematic evaluation, while Jahangiri and Rakha (2015) highlighted their versatility across different transportation modes, variations, and contexts. The logic underlying classical ML algorithms was conceptually similar to that of rule-based models, which relied on indicative features computed from GPS trajectories and used them to infer travel modes statistically. For example, mobility features derived from dense GPS trajectories—such as average speed and acceleration rate—were often effective in distinguishing between driving, walking, and cycling.

Although computationally efficient, these approaches depended heavily on domain expertise for feature engineering and performed poorly when handling variable-length inputs. In many inner-city trips, for instance, driving was only marginally faster than cycling, rendering average speed an unreliable discriminator. More advanced statistical features could improve performance but required specialized expertise and local contextual knowledge that were seldom available and sometimes unreliable.

Moreover, such methods posed risks to privacy and anonymity, as even a small number of precise and longitudinal GPS points could be sufficient to re-identify individuals (De Montjoye *et al.* 2013). Growing concerns regarding the collection, management, and disclosure of personal GPS data, together with advances in re-identification techniques, have further raised ethical issues surrounding GPS trajectory research (Michael *et al.* 2006, Klasnja *et al.* 2009, Minch 2004). In response, scholars and regulators have increasingly advocated for privacy-preserving techniques in mobility research—including applications such as transportation mode detection—as alternatives to classical machine learning approaches that depend on rich location features and extensive feature engineering (Ng-Kruelle *et al.* 2002, Krumm 2009, Shin *et al.* 2012, Thompson and Warzel 2022, Jiang *et al.* 2021).

Recurrent neural networks (RNNs), particularly Long Short-Term Memory (LSTM) networks (Hochreiter and Schmidhuber 1997), emerged as a dominant approach for transportation mode prediction due to their capacity to capture temporal dependencies in sequential data. Jiang *et al.* (2017) pioneered the application of LSTMs for mobility analysis, and Asci and Guvensan (2019) extended this work by incorporating attention mechanisms that improved performance across varied trip lengths. Hybrid architectures combining LSTMs with other neural components further advanced model accuracy—for example, the Conv-LSTM model proposed by Nawaz *et al.* (2020) leveraged convolutional layers for spatial feature extraction prior to temporal processing. Other innovations included Convolutional Neural Networks (CNNs) (Dabiri and Heaslip 2018), autoregressive flow models (Dutta and Patra 2023), and semi-supervised learning approaches (Dabiri *et al.* 2020). Despite these accuracy gains, such sophisticated architectures often required substantial computational resources and complex pre-processing to structure data according to model specifications, thereby limiting their practicality for real-world deployment (Graser *et al.* 2024).

2.2. *Transformers for Mobility Modeling*

Transformer architectures revolutionized sequence modeling through their reliance on self-attention and multi-head attention mechanisms (Cheng *et al.* 2016, Vaswani *et al.* 2017). Models such as GPT-2 (Radford *et al.* 2019) and BERT (Devlin *et al.* 2019) demonstrated exceptional performance on sequence-based data, significantly surpassing classical machine learning models in capturing complex sequential patterns. This success

motivated researchers to explore whether Transformers' superior capacity for modeling long-range dependencies could similarly advance mobility modeling, where understanding the relationships between distant points within a trajectory is critical (Xue *et al.* 2022).

Recent applications of Transformer architectures to mobility studies have shown promise, although the field remains in its early stages. For example, Hong *et al.* (2022) incorporated Transformer components for next-location prediction while treating mode identification as a secondary task. Liang *et al.* (2022) addressed challenges such as irregular spatiotemporal intervals but focused primarily on spatio-temporal dependencies rather than transportation mode prediction. More recently, Ribeiro *et al.* (2024) achieved strong results using a vision Transformer approach (Dosovitskiy *et al.* 2021), although their implementation required converting trajectories into image-based representations. Similarly, Drosouli *et al.* (2023) converted GPS trajectory points into word tokens to apply the original BERT model (Devlin *et al.* 2019), achieving notable results; however, this approach imposed a linguistic abstraction onto spatial data, making it less suitable for general-purpose mobility modeling.

Despite these advances, existing Transformer-based approaches for mobility modeling typically required extensive pre-processing, multiple input features, or auxiliary contextual information—making them computationally demanding and often impractical for real-world applications where only basic GPS data are available. Moreover, their generalizability across different geographical contexts has remained largely unexplored, as most evaluations have focused on performance within a single dataset rather than testing transferability between regions with distinct transportation infrastructures and mobility behaviors.

3. SpeedTransformer Architecture

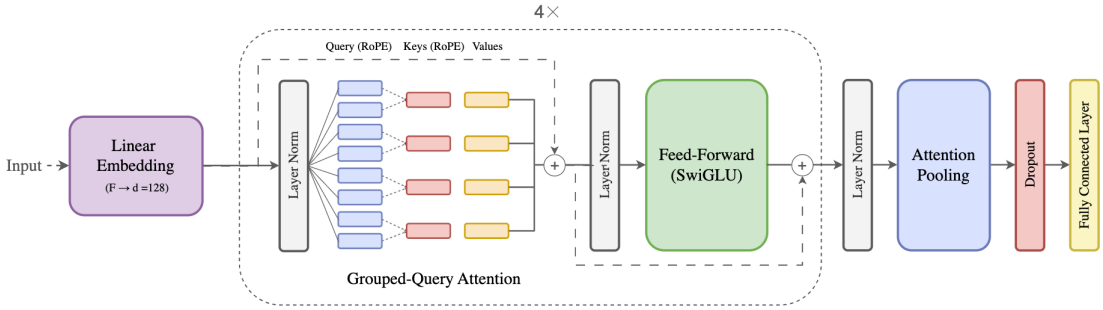

Figure 1. Transformer architecture

Our SpeedTransformer architecture adapted the Transformer encoder framework proposed by Vaswani *et al.* (2017), incorporating several key modifications to the data input, model structure, and output. First, the input sequences consisted of instantaneous velocities computed from dense GPS trajectories collected via transportation applications on smartphones. These velocity sequences were sampled at high frequencies—typically representing distances traveled over five to ten seconds—and therefore implicitly encoded higher-order motion features such as acceleration (the first derivative of velocity) and jerk (the second derivative of velocity). Given a complete velocity sequence from trip start to end, each velocity value served as a token, which was subsequently embedded

and transformed into a query vector representing its relationship to all other velocity positions within the sequence.

Figure 1 illustrated the overall model architecture. To preserve privacy and avoid real-time location data collection, our model required only raw scalar speed sequences as trajectory input. To accommodate variable input lengths, each trajectory was segmented into fixed-length sequences of $T = 200$ consecutive speed samples using a sliding window with a stride of 50 (see Appendix F for further discussion on window size selection). Shorter sequences were zero-padded, and a key-padding mask was applied to ensure that padded tokens were ignored during attention and pooling.

Each scalar speed value s_t was linearly projected into a $d = 128$-dimensional embedding space (see Appendix D for details on the embedding process). The sequence of embedded speed vectors was then processed by a modified Transformer encoder. Owing to its attention mechanism, the model was able to extract sequential dependencies—such as acceleration and jerk—from these speed embeddings, which were critical for differentiating transportation modes. When trained on sufficiently large empirical datasets, the model effectively optimized its capacity to detect transportation modes.

We replaced the standard sinusoidal positional encoding with Rotary Positional Embeddings (RoPE), which were directly applied to the query (Q) and key (K) vectors in the attention mechanism to introduce continuous relative positional awareness (Su *et al.* 2021). This approach allowed the model to encode both absolute and relative spatial dependencies in a continuous, rotation-invariant manner, making it particularly suitable for sequential mobility data such as speed trajectories.

The encoder consisted of $L = 4$ Pre-Norm Transformer blocks. Each block contained two key components: a Grouped-Query Attention (GQA) layer (Ainslie *et al.* 2023), which efficiently computed attention by grouping multiple query vectors per key–value pair, and a SwiGLU-activated feed-forward sublayer (Shazeer 2020) that introduced non-linear transformations with improved gradient flow and expressivity (see Appendix E for a detailed explanation of SwiGLU). The computation within each block was expressed as follows:

$$\mathbf{z}_1 = \mathbf{x} + \mathrm{GQA}(\mathrm{LayerNorm}(\mathbf{x})), \tag{1}$$

$$\mathbf{z}_2 = \mathbf{z}_1 + \mathrm{FFN}_{\mathrm{SwiGLU}}(\mathrm{LayerNorm}(\mathbf{z}_1)), \tag{2}$$

Equation 1 described the attention sub-layer, in which the input representation $\mathbf{x} \in \mathbb{R}^{T\times d}$ (a sequence of T tokens, each of dimension d) was first normalized using layer normalization before being passed to the GQA mechanism. The resulting attention output was then added back to the input through a residual connection. This structure enabled the model to integrate contextual information—such as temporal dependencies between consecutive speeds—without destabilizing training.

Equation 2 represented the feed-forward transformation that followed the attention mechanism. The intermediate output $\mathbf{z}_1$ was normalized again and passed through a SwiGLU-activated feed-forward network, $\mathrm{FFN}_{\mathrm{SwiGLU}}(\cdot)$, which introduced non-linear transformations to enhance representational capacity while maintaining computational efficiency. The residual connection in this step further stabilized gradient propagation across layers.

We employed $h = 8$ query heads and $h_{kv} = 4$ shared key/value heads, following the formulation of Ainslie *et al.* (2023). This design reduced both memory usage and computational cost by allowing multiple query heads to share the same key and value projections

while preserving diversity across query subspaces. RoPE were applied to Q and K before computing scaled dot-product attention, and padding masks were introduced to ensure that zero-padded tokens did not contribute to the attention weights.

After the encoder stack, a final layer normalization was applied. The contextualized sequence representation was then aggregated through attention pooling: each timestep embedding received a learnable scalar attention score, which was normalized using a masked softmax function across valid time-steps. The pooled sequence embedding $\mathbf{c}$ was computed as the weighted sum of the contextual embeddings, capturing salient temporal information. Dropout regularization was applied to mitigate overfitting. Finally, the pooled representation $\mathbf{c}$ was passed through a linear projection layer to produce class logits, followed by a softmax activation to obtain the probability distribution over transportation mode categories.

4. Datasets

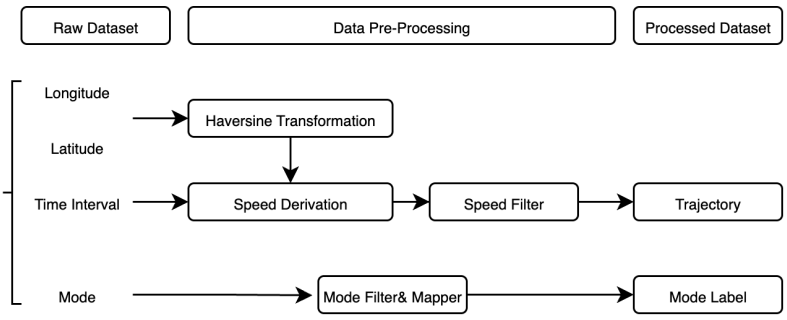

Figure 2. Data Pre-Processing

We utilized two longitudinal tracking datasets—the widely adopted Geolife dataset (Zheng *et al.* 2010) and the Swiss MOBIS dataset (Heimgartner and Axhausen 2024)—which offered complementary strengths for evaluating our model. Summary statistics are presented in Table 1. As illustrated in Figure 2, we standardized their data structures through a unified pre-processing pipeline: we consolidated transportation modes into five consistent categories (Bike, Bus, Car, Train, Walk), unified the trajectory input format across datasets, converted geographic coordinates into speed sequences, removed abnormal trips with erroneous location data, and standardized transportation mode labels. Detailed descriptions of the pre-processing procedures and raw

Table 1. Comparison of MOBIS and Geolife Datasets

Mode	**Data Points**		**Unique Trips**	
	MOBIS	**Geolife**	**MOBIS**	**Geolife**
Bike	4,251,028	746,098	40,171	1,555
Bus	4,606,409	1,061,196	74,324	1,847
Car	88,965,473	627,047	542,078	1,293
Train	8,379,962	788,250	130,964	772
Walk	38,463,266	1,215,054	743,425	3,960
Total	144,666,138	4,437,645	1,530,962	9,427

dataset statistics are provided in Appendix A.

4.1. *MOBIS Dataset*

The MOBIS dataset (Molloy *et al.* 2023) was derived from an eight-week randomized controlled trial (RCT) on transport pricing involving 3,680 participants in Switzerland. Each participant used the *Catch-my-Day* mobile application (available for both iOS and Android), which continuously recorded GPS data via the device's location services. The application captured daily travel patterns, storing raw trajectory data locally before uploading them to the MotionTag analytics platform, where trip segmentation and transportation mode inference were performed. This extensive data collection process produced 255.3 million GPS records and 1.58 million labeled trips in the raw dataset. After applying our standardized pre-processing pipeline described in Appendix A, the resulting MOBIS dataset contained 144.7 million data points across 1.53 million unique trips distributed among five transportation modes, as summarized in Table 1.

4.2. *Geolife Dataset*

Geolife (Zheng *et al.* 2010) was a widely used benchmark in transportation mode research, collected by Microsoft Research Asia from 182 users in Beijing over a five-year period (2007–2012). The dataset captured urban mobility through 17,000 trajectories covering approximately 1.2 million kilometers across Beijing's complex transportation network. Data were recorded using various GPS loggers and GPS-enabled phones with different sampling rates, with 91% of trajectories collected at high density (every 1–5 seconds or every 5–10 meters). In addition to routine commutes, the Geolife dataset included leisure and sports activities such as shopping, sightseeing, and cycling, offering rich contextual diversity in trip purposes. After pre-processing to align with the MOBIS data structure, the Geolife dataset contained 4.44 million data points and 9,427 unique trips—substantially smaller than MOBIS, yet providing valuable geographical and temporal diversity for evaluating our model's performance.

5. Experiments

We evaluated SpeedTransformer under three experimental conditions. First, we benchmarked it against state-of-the-art transportation mode identification models on Geolife (Zheng *et al.* 2010), enabling direct comparison with existing approaches. Second, we examined performance consistency across different geographical contexts by comparing it with classical LSTM baseline models on both the Swiss (MOBIS) (Heimgartner and Axhausen 2024) and Chinese (Geolife) datasets. Finally, we assessed cross-regional transferability through fine-tuning experiments, in which models pretrained on Swiss data were adapted to Chinese mobility patterns using small sample subsets.

5.1. *Benchmarking Performance*

To evaluate SpeedTransformer's ability to achieve high accuracy with minimal input, we benchmarked against several state-of-the-art transportation mode identification models:

- **LSTM-Attention (Baseline)**: Our reconstructed baseline model implementing a

Table 2. Test Accuracy Comparison on Geolife (Ordered by Performance)

Model	**Test Acc. (%)**
SpeedTransformer (Ours)	**95.97**
Deep-ViT (Ribeiro *et al.* 2024)	92.96
LSTM-based DNN (Yu 2020)	92.70
LSTM-Attention (Baseline)	**92.40**
CE-RCRF (Zeng *et al.* 2023)	85.23
SECA (Dabiri *et al.* 2020)	84.80
ConvLSTM (Nawaz *et al.* 2020)	83.81

classical bidirectional LSTM with attention mechanism (Hochreiter and Schmidhuber 1997). This baseline is specifically designed to test whether pure attention-based mechanisms outperform recurrent networks augmented with attention, while maintaining the same minimal input requirement (speed only).

- **Deep-ViT** (Ribeiro *et al.* 2024): A Vision Transformer approach that transforms GPS features (velocity, acceleration, bearing) into image representations using DeepInsight methodology before processing with a Vision Transformer architecture, combining traditional feature engineering with advanced deep learning.
- **CE-RCRF** (Zeng *et al.* 2023): A sequence-to-sequence framework (TaaS) that processes entire trajectories using a Convolutional Encoder to extract high-level features and a Recurrent Conditional Random Field to maintain contextual information at both feature and label levels, with specialized bus-related features to distinguish high-speed modes.
- **LSTM-based DNN** (Yu 2020): An ensemble of four LSTM networks that incorporates both time-domain trajectory attributes and frequency-domain statistics developed through discrete Fourier and wavelet transforms, creating a semi-supervised deep learning approach.
- **SECA** (Dabiri *et al.* 2020): A Semi-supervised Convolutional Autoencoder that integrates a convolutional-deconvolutional autoencoder with a CNN classifier to simultaneously leverage labeled and unlabeled GPS segments, automatically extracting relevant features from 4-channel tensor representations.
- **ConvLSTM** (Nawaz *et al.* 2020): A hybrid architecture that uses convolutional layers to extract spatial features from GPS data, followed by LSTM layers to capture temporal patterns, incorporating both location and weather features to enhance mode detection.

All models were evaluated on the Geolife dataset using identical initial pre-processing and train-test splits. Table 2 presents the test accuracies ranked from highest to lowest.

SpeedTransformer achieved the highest test accuracy of 95.97%, outperforming all competing approaches despite using only speed as input. Deep-ViT (92.96%) and the LSTM-based DNN (92.70%) achieved strong results but required more complex preprocessing or architectural components. Our LSTM-Attention baseline (92.40%) demonstrated that while recurrent networks with attention could capture temporal patterns effectively, they still lagged behind the pure attention-based design of SpeedTransformer in predicting complex mobility patterns.

The lower-ranked models illustrated different architectural trade-offs. The CNN-ensemble extracted localized spatial patterns but struggled with sequential dependencies;

ConvLSTM improved temporal modeling through its hybrid design but faced generalization challenges; and CE-RCRF treated trajectories as continuous sequences yet was hindered by architectural complexity. We also evaluated a simple rule-based model using the same process and found that it performed substantially worse than all machine learning models (Appendix H).

To further assess performance consistency across datasets, we compared SpeedTransformer with our LSTM-Attention baseline on both the Geolife and MOBIS datasets. Figures 3 and 4 show that SpeedTransformer not only converged faster and achieved higher validation accuracy but also maintained superior F1-scores across all transportation modes. The F1-score in Figure 4, which examines accuracy by class, provided a more granular and reliable measure of model performance under class imbalance. Moreover, Table 3 presents a detailed comparison of precision and recall between SpeedTransformer and Deep-ViT—the most competitive alternative—demonstrating that SpeedTransformer achieved a more balanced trade-off between precision and recall across datasets, underscoring its robustness and generalizability.

SpeedTransformer demonstrated two significant advantages over the LSTM-Attention baseline: faster convergence and higher accuracy across both datasets. As

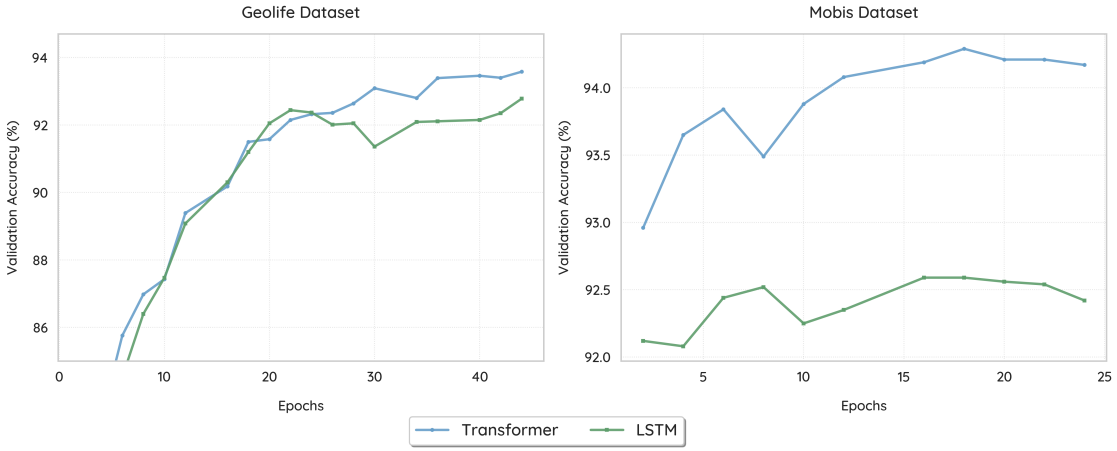

Figure 3. Validation accuracies over epochs for Geolife and MOBIS. The SpeedTransformer consistently converges faster and achieves higher overall accuracy than the LSTM-Attention baseline on both datasets.

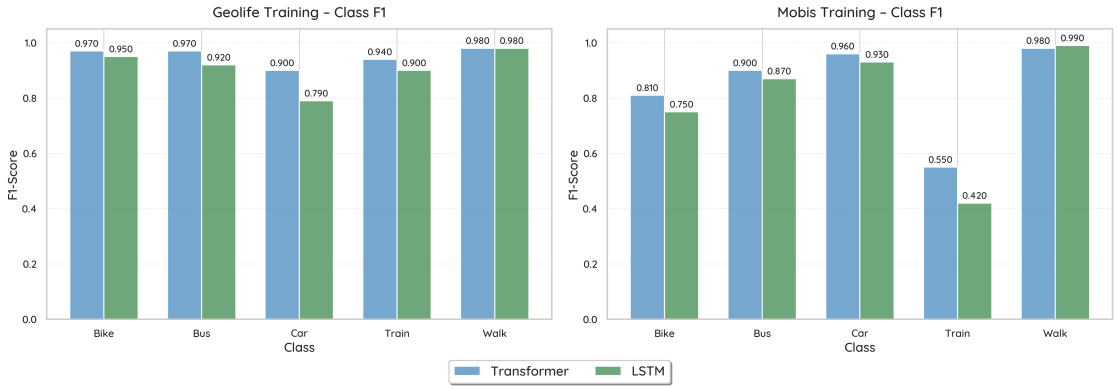

Figure 4. Per class F1-Score for Geolife and MOBIS trainings using SpeedTransformer and LSTM. The SpeedTransformer consistently achieves better results than the LSTM-Attention across all classes on both datasets.

Table 3. Validation Precision and Recall Comparison between Deep-ViT and SpeedTransformer on Geolife

Mode	**Deep-ViT**		**SpeedTransformer**	
	Precision (%)	**Recall (%)**	**Precision (%)**	**Recall (%)**
Walk	95.04	96.12	**96.72**	**99.90**
Bike	93.79	91.40	**98.54**	**95.45**
Bus	89.68	91.13	**98.14**	**95.49**
Car	**88.96**	87.26	87.33	**93.65**
Train	90.00	79.41	**95.66**	**92.93**
Macro Avg.	91.89	89.86	**95.68**	**95.08**

Table 4. Model Accuracy during Cross-Dataset Fine-Tuning from MOBIS to Small Geolife Subsets

Model	**100 trips**	**200 trips**
LSTM-Attention	75.47%	79.15%
SpeedTransformer	80.53%	86.13%

shown in Figure 4, the model consistently achieved higher validation accuracies throughout training, reaching 94.22% accuracy on MOBIS compared to 92.33% for the LSTM-Attention model. The faster convergence was particularly valuable when working with large-scale mobility datasets such as MOBIS, where training efficiency was critical. We also observed that SpeedTransformer achieved superior per-class F1-scores across all transportation modes in both the Geolife and MOBIS datasets. Although the model required GPU resources, it remained relatively efficient, completing both training and inference on a single GPU node (Appendix I).

5.2. *Cross-Regional Transferability*

Learning from raw trajectories was feasible for all models, including rule-based approaches. However, deep learning architectures such as Transformers achieved state-of-the-art accuracy through inductive transfer and fine-tuning of pretrained models (Howard and Ruder 2018). This transfer-learning approach was particularly advantageous when cross-regional differences in transportation networks and traffic dynamics produced distinct patterns of transportation modes. To evaluate model transferability, we fine-tuned the MOBIS-pretrained models on small subsets of the Geolife dataset to simulate low-shot adaptation. Both SpeedTransformer and LSTM-Attention were fine-tuned on 100 and 200 Geolife trips (approximately 1.1% and 2.2% of the dataset, respectively) for 20 epochs and were subsequently evaluated on the remaining samples. Table 4 reports the overall classification accuracy.

Fine-tuning followed a standardized low-shot protocol using MOBIS-pretrained checkpoints. For SpeedTransformer, the encoder and attention layers were frozen, and training was performed with mixed precision for up to 20 epochs. The LSTM-Attention baseline was fine-tuned with smaller batch sizes, a lower learning rate, stronger regularization (dropout = 0.3, weight decay = 5e-3), and gradient clipping at 0.25. Hyperparameters were selected through a grid search to ensure optimal stability under limited supervision (see Appendix C). SpeedTransformer achieved 80.53% accuracy with

only 100 trips, surpassing LSTM-Attention by more than five percentage points, and further improved to 86.13% with 200 trips. These results confirmed its superior capacity to transfer learned mobility representations across regions with minimal labeled data.

6. Real-World Field Experiment

Having established SpeedTransformer's superior accuracy, minimal input requirements, and strong cross-regional transferability, we next evaluated its robustness under real-world conditions, where GPS data were inherently noisy, irregular, and device-dependent. While curated benchmarks provided clean and controlled comparisons, they did not capture the complexities of real-world mobility data, which were subject to signal loss, user heterogeneity, and hardware variability. To address this gap, we conducted a large-scale field experiment to assess SpeedTransformer's reliability in real-world environments characterized by high uncertainty and unpredictability.

6.1. *Smartphone Application and Data Collection*

We developed *CarbonClever*, a WeChat-integrated mini-program designed to estimate individual carbon footprints through continuous mobility tracking.[1] The application provided a streamlined interface for trip initiation, real-time monitoring, and post-trip mode

[1]A WeChat mini-program is a lightweight application embedded within the WeChat ecosystem. Functionally similar to a simplified smartphone app, the mini-program operates on top of the WeChat super-app, which is ubiquitously used in China. This integration enabled efficient participant recruitment and minimized testing costs across different mobile operating systems and device platforms.

(a) Tracking initiation (b) Active tracking (c) Trip completion

Figure 5. User-initiated trip tracking interface in the CarbonClever application: (a) tracking initiation, (b) active tracking with real-time speed and distance updates, and (c) trip completion with mode verification.

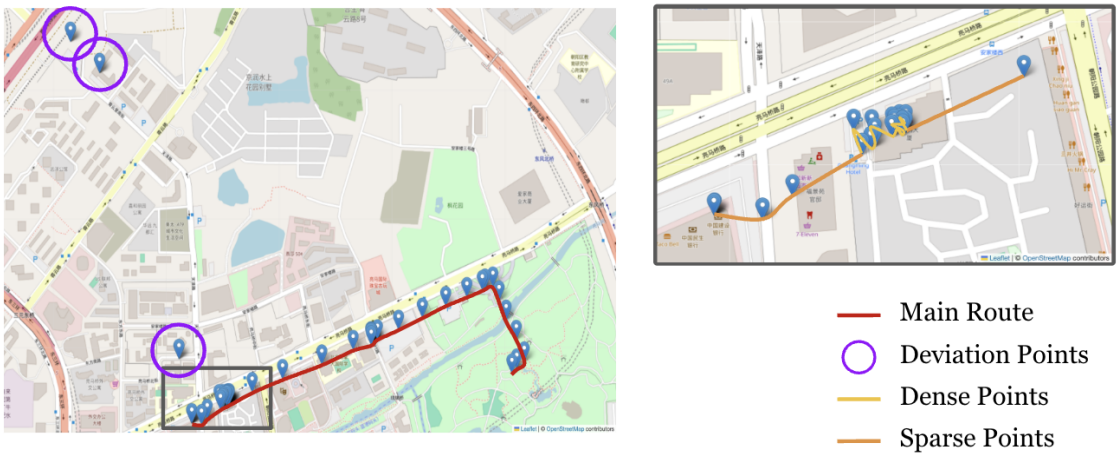

Figure 6. Example of real-world GPS trajectories. The red line indicates the main route, with varying point density reflecting heterogeneous sampling frequencies. Purple circles mark signal interference and positioning noise.

verification (Figure 5). A total of 348 participants from Jiangsu Province, China, were recruited through an environmental organization to record their daily movements.[2]

Real-world GPS traces differed substantially from those collected under laboratory conditions. The sampling frequencies varied widely: older iPhone SE models recorded locations every 30–60 s, whereas newer iPhone 14 Pro devices achieved 5–10 s intervals under identical conditions. As shown in Figure 6, trajectories exhibited irregular sampling densities, signal dropouts, and spurious positional jumps caused by multi-path effects or indoor transitions. Unlike benchmark datasets, we intentionally retained these imperfections to evaluate the model's robustness under realistic deployment conditions.

Table 5. Summary of the Mini-Program Dataset

Mode	Number of Data Points	Number of Unique Trips
Walk	28,985	100
Bus	32,547	259
Car	40,885	205
Bike	4,901	36
Train	1,505	49
Total	**108,823**	**649**

6.2. *Field Experiment Evaluation*

We conducted a three-day field experiment that collected 649 verified trips totaling 108,823 GPS points from heterogeneous devices across both iOS and Android platforms, ensuring representative variability in sampling rates and noise profiles (see Appendix B for detailed data pre-processing). To assess real-world adaptability, we fine-tuned MOBIS-pretrained SpeedTransformer and LSTM-Attention models using progressively larger subsets of the collected data. Each model was trained for up to 20 epochs with

[2]The experiment was approved by the Duke Kunshan University Institutional Review Board (protocol 2022CC073). See Appendix J for protocol summary.

Table 6. Fine-Tuning Accuracies of Models with Real-World Data Subsets

Data Subset (%)	**LSTM-Attention Acc. (%)**	**SpeedTransformer Acc. (%)**
15% (94 trips)	86.62	89.12
20% (125 trips)	88.02	92.53
30% (188 trips)	88.97	91.15
40% (251 trips)	87.45	88.78
50% (314 trips)	87.57	94.22

early stopping and identical hyperparameter configurations derived from the grid search (Appendix C).

SpeedTransformer consistently outperformed the LSTM-Attention model across most data samples, as shown in Table 6. SpeedTransformer achieved a peak accuracy of 94. 22%, while LSTM-Attention plateaued around 88%. Taken together, the real-world experimental results further corroborate our earlier findings, demonstrating that SpeedTransformer not only generalizes effectively across geographical contexts but also remains robust under genuine, noisy, and device-diverse GPS data—bridging the gap between research prototypes and operational smart-mobility applications.

7. Discussion and Conclusion

Our research addressed three fundamental challenges in transportation mode detection: (1) the reliance on extensive feature engineering and input pre-processing, (2) limited model generalizability across distinct geographical contexts, and (3) insufficient and unpredictable real-world validation. By achieving state-of-the-art transportation mode detection, our work contributed to a growing body of literature on transportation mode detection and its downstream applications in transportation research, GIS, urban analytics, and climate change science. Compared to classical and deep ML methods (Ribeiro *et al.* 2024, Zeng *et al.* 2023, Yu 2020, Dabiri *et al.* 2020, Nawaz *et al.* 2020), SpeedTransformer outperformed feature-rich models despite using only speed as input. Across different hyperparameter configurations, our model consistently outperformed the LSTM-Attention baseline by 2–3% (Appendix C) and surpassed several other machine learning models by more than 10% (Table 2). It also outperformed the classical rule-based model, which relied solely on speed, by over 30% (Appendix H).

The model's strong performance was largely attributable to its attention mechanism (Appendix G). Nevertheless, our integration of SwiGLU activation, Grouped-Query Attention, and pre-attention Layer Normalization yielded modest yet consistent performance improvements, aligning with the findings of Shazeer (2020) and Touvron *et al.* (2023). Consistent with Yu and Wang (2023), our results further indicated that deeper neural network architectures were better suited to capturing the semantic and sequential structures embedded within GPS trajectories.

Moreover, this strong performance relied solely on instantaneous speed—a single feature independent of real-time location—without requiring additional engineered features such as acceleration. This architectural choice offered several advantages: it preserved user privacy by avoiding the direct collection of sensitive location data (Thompson and Warzel 2022, De Montjoye *et al.* 2013), and it simplified preprocessing procedures, thereby enhancing reproducibility (Bolbol *et al.* 2012b).

Furthermore, SpeedTransformer exhibited strong cross-regional transferability. In

the out-of-domain evaluation, it surpassed the LSTM-Attention model by 7% (Table 4), demonstrating robust generalizability across distinct geographical and transportation contexts—from Swiss transportation systems (MOBIS) to Chinese urban mobility (Geolife). Travel behaviors varied substantially between these regions, and the two datasets were collected nearly a decade apart, introducing considerable differences in travel patterns across modes such as train, cycling, and automobile use. Remarkably, with only 100 fine-tuning trips, SpeedTransformer achieved satisfactory accuracy (80.53%) on Geolife after pretraining on MOBIS. These findings suggested that SpeedTransformer captured fundamental patterns of human mobility that persisted across infrastructural settings and regional mobility cultures.

Finally, to bridge the gap between research prototypes and real-world implementation, we validated SpeedTransformer in a large-scale deployment via our *CarbonClever* WeChat Mini-Program. This mobile application collected GPS trajectories from 348 participants across diverse smartphones and operating systems. Unlike curated benchmarks, this dataset included irregular sampling intervals, signal interferences, and device-specific noises. SpeedTransformer consistently outperformed the LSTM-Attention baseline across all fine-tuning results, achieving 94.22% accuracy with 50% of the real-world training data, compared to 87.57% for LSTM-Attention. While accuracy decreased slightly due to data irregularities, SpeedTransformer retained high stability across transportation modes, confirming its robustness for practical deployment.

A key limitation of Transformer-based models, including SpeedTransformer, was their sensitivity to training data quality. As shown in Table 3 and Figure 4, the model performed less effectively for the *Train* class in the MOBIS dataset, where data imbalance and under-representation reduced its ability to generalize. As Van Hulse *et al.* (2007) noted, class imbalance posed significant challenges to ML accuracy. Still, by pretraining on the larger, more balanced MOBIS dataset and fine-tuning on the smaller, imbalanced Geolife subsets, SpeedTransformer maintained strong predictive accuracy with minimal performance degradation under distributional shift. The model performed even better using our real-world experimental data (Table 6). These results underscored the model's adaptability to data-sparse and heterogeneous mobility environments.

Another limitation concerned the model's limited interpretability—specifically, the difficulty of attributing learned patterns to interpretable mobility features such as acceleration bursts, stop durations, or route choices. Future research could investigate hybrid architectures that integrate Transformer attention mechanisms with physically interpretable mobility features and extend the framework to incorporate contextual signals—such as weather, traffic density, and land use—for downstream mobility tasks, including origin–destination flow estimation and route-level analysis. In addition, expanding evaluation to regions with different road infrastructures and behavioral norms could further validate model universality.

Acknowledgments

We thank the Institute for Transport Planning and Systems at ETH Zürich for providing the data used in our model training. We are also grateful to Yucen Xiao, Peilin He, Zhixuan Lu, Yili Wen, Shanheng Gu, Ziyue Zhong, and Ni Zheng for their excellent research assistance. This project was supported by the 2024 Kunshan Municipal Key R&D Program under the Science and Technology Special Project (No. KS2477). ChatGPT 5 was used solely for grammar checking. All remaining errors are our own.

8. Data and Codes Availability Statement

Replication data and code can be found at :
`https://github.com/theanonymousresearcher/SpeedTransformer`
The repository provides source code for data processing, model training and evaluation.

We have also prepared two Colab notebooks, one with main results and most of the results in appendices at :
`https://shorturl.at/dzVkb`
and the other with remaining results in the appendices at `https://shorturl.at/XZcB8`

The original datasets of MOBIS and Geolife can be found at their respective project websites: the MOBIS dataset (https://www.research-collection.ethz.ch/handle/20.500.11850/553990) and the Geolife dataset (https://www.microsoft.com/en-us/research/publication/Geolife-gps-trajectory-dataset-user-guide/), both accessible on May 26, 2025.

References

Ainslie, J., *et al.*, 2023. GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints. *arXiv preprint arXiv:2305.13245*.

Aleta, A., *et al.*, 2022. Quantifying the importance and location of SARS-CoV-2 transmission events in large metropolitan areas. *Proceedings of the National Academy of Sciences of the United States of America*, 119 (26), e2112182119.

Asci, G. and Guvensan, M.A., 2019. A Novel Input Set for LSTM-Based Transport Mode Detection. *In*: *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)* IEEE, 107–112.

Barbosa, H., *et al.*, 2015. The effect of recency to human mobility. *EPJ Data Science*, 4, 21.

Barbosa, H., *et al.*, 2021. Uncovering the socioeconomic facets of human mobility. *Scientific reports*, 11 (1), 8616.

Bolbol, A., *et al.*, 2012a. Inferring hybrid transportation modes from sparse GPS data using a moving window SVM classification. *Computers, Environment and Urban Systems*, 36 (6), 526–537.

Bolbol, A., *et al.*, 2012b. Inferring hybrid transportation modes from sparse GPS data using a moving window SVM classification. *Computers, Environment and Urban Systems*, 36 (6), 526–537.

Bonaccorsi, G., *et al.*, 2020. Economic and social consequences of human mobility restrictions under COVID-19. *Proceedings of the national academy of sciences*, 117 (27), 15530–15535.

Breiman, L., 2001. Random Forests. *Machine Learning*, 45, 5–32.

Cheng, J., Dong, L., and Lapata, M., 2016. Long short-term memory-networks for machine reading. *arXiv preprint arXiv:1601.06733*.

Cui, Y.J. and Ge, S.S., 2001. Autonomous vehicle positioning with GPS in urban canyon environments. *In*: *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, Vol. 2, Seoul, Korea (South), 1105–1110.

Dabiri, S. and Heaslip, K., 2018. Inferring transportation modes from GPS trajectories using a convolutional neural network. *Transportation Research Part C: Emerging*

Technologies, 86, 360–371.

Dabiri, S., *et al.*, 2020. Semi-Supervised Deep Learning Approach for Transportation Mode Identification Using GPS Trajectory Data. *IEEE Transactions on Knowledge and Data Engineering*, 32 (5), 1010–1023.

De Montjoye, Y.A., *et al.*, 2013. Unique in the crowd: The privacy bounds of human mobility. *Scientific reports*, 3 (1), 1376.

Devlin, J., *et al.*, 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *In*: J. Burstein, C. Doran and T. Solorio, eds. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Jun.. Minneapolis, Minnesota: Association for Computational Linguistics, 4171–4186.

Dosovitskiy, A., *et al.*, 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv preprint arXiv:2010.11929* ICLR 2021.

Drosouli, I., *et al.*, 2023. TMD-BERT: a transformer-based model for transportation mode detection. *Electronics*, 12 (3), 581.

Dutta, S. and Patra, B.K., 2023. Inferencing transportation mode using unsupervised deep learning approach exploiting GPS point-level characteristics. *Applied Intelligence*, 53 (10), 12489–12503.

Girod, B., van Vuuren, D.P., and de Vries, B., 2013. Influence of travel behavior on global CO2 emissions. *Transportation Research Part A: Policy and Practice*, 50, 183–197.

Gonzalez, M.C., Hidalgo, C.A., and Barabasi, A.L., 2008. Understanding individual human mobility patterns. *nature*, 453 (7196), 779–782.

Goodchild, M.F., 2013. The quality of big (geo) data. *Dialogues in Human Geography*, 3 (3), 280–284.

Gordon, J.B., *et al.*, 2013. Automated Inference of Linked Transit Journeys in London Using Fare-Transaction and Vehicle Location Data. *Transportation Research Record*, 2343, 17 – 24.

Graser, A., *et al.*, 2024. MobilityDL: A Review of Deep Learning From Trajectory Data. *arXiv preprint arXiv:2402.00732* Submitted to Geoinformatica.

Guo, B., *et al.*, 2015. Mobile crowd sensing and computing: The review of an emerging human-powered sensing paradigm. *ACM computing surveys (CSUR)*, 48 (1), 1–31.

Heimgartner, D. and Axhausen, K.W., 2024. Modal splits before, during, and after the pandemic in Switzerland. *Transportation research record*, 2678 (7), 1084–1099.

Hochreiter, S. and Schmidhuber, J., 1997. Long Short-Term Memory. *Neural Comput.*, 9 (8), 1735–1780.

Hong, Y., Martin, H., and Raubal, M., 2022. How do you go where? Improving next location prediction by learning travel mode information using transformers. *In*: *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*, Nov.. ArXiv:2210.04095 [cs], 1–10.

Howard, J. and Ruder, S., 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

Huang, H., Cheng, Y., and Weibel, R., 2019. Transport mode detection based on mobile phone network data: A systematic review. *Transportation Research Part C: Emerging Technologies*, 101, 297–312.

Jahangiri, A. and Rakha, H.A., 2015. Applying Machine Learning Techniques to Transportation Mode Recognition Using Mobile Phone Sensor Data. *IEEE Transactions on Intelligent Transportation Systems*, 16 (5), 2406–2417 Conference Name: IEEE Transactions on Intelligent Transportation Systems.

Jiang, H., *et al.*, 2021. Location privacy-preserving mechanisms in location-based services: A comprehensive survey. *ACM Computing Surveys (CSUR)*, 54 (1), 1–36.

Jiang, X., *et al.*, 2017. TrajectoryNet: An Embedded GPS Trajectory Representation for Point-based Classification Using Recurrent Neural Networks. *arXiv preprint arXiv:1705.02636*.

Klasnja, P., *et al.*, 2009. Exploring privacy concerns about personal sensing. *In*: *International Conference on Pervasive Computing*, 176–183.

Krumm, J., 2009. A survey of computational location privacy. *Personal and Ubiquitous Computing*, 13 (6), 391–399.

LeCun, Y. and Bengio, Y., 1998. *In*: *Convolutional networks for images, speech, and time series.*, p. 255–258 Cambridge, MA, USA: MIT Press.

Liang, Y., *et al.*, 2022. TrajFormer: Efficient Trajectory Classification with Transformers. *In*: *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, Atlanta, GA, USA New York, NY, USA: Association for Computing Machinery, 1229–1237.

Lu, X., Bengtsson, L., and Holme, P., 2012. Predictability of population displacement after the 2010 Haiti earthquake. *Proceedings of the National Academy of Sciences*, 109 (29), 11576–11581.

Manzoni, V., *et al.*, 2010. *Transportation Mode Identification and Real-Time CO2 Emission Estimation Using Smartphones: How CO2GO Works - Technical Report.* Technical report, SENSEable City Lab, Massachusetts Institute of Technology and Politecnico di Milano.

McMichael, C., 2020. Human mobility, climate change, and health: Unpacking the connections. *The Lancet Planetary Health*, 4 (6), e217–e218.

Michael, K., McNamee, A., and Michael, M.G., 2006. The emerging ethics of human-centric GPS tracking and monitoring. *In*: *2006 International Conference on Mobile Business*, 34–34.

Minch, R.P., 2004. Privacy issues in location-aware mobile devices. *In*: *37th Annual Hawaii International Conference on System Sciences, 2004. Proceedings of the*, 10–pp.

Molloy, J., *et al.*, 2023. The MOBIS dataset: a large GPS dataset of mobility behaviour in Switzerland. *Transportation*, 50 (5), 1983–2007.

Nawaz, A., *et al.*, 2020. Convolutional LSTM based transportation mode learning from raw GPS trajectories. *IET Intelligent Transport Systems*, 14 (6), 570–577 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1049/iet-its.2019.0017.

Ng-Kruelle, G., *et al.*, 2002. The price of convenience: Privacy and mobile commerce. *Quarterly journal of electronic commerce*, 3, 273–286.

Pappalardo, L., *et al.*, 2023. Future directions in human mobility science. *Nature Computational Science*, 3 (7), 588–600.

Prelipcean, A.C., Gidófalvi, G., and Susilo, Y.O., 2016. Transportation Mode Detection – an in-Depth Review of Applicability and Reliability. *Transport Reviews*, 37 (4), 442–464.

Preotiuc-Pietro, D. and Cohn, T., 2013. Mining User Behaviours: A Study of Check-in Patterns in Location Based Social Networks. *In*: *Proceedings of the 3rd Annual ACM Web Science Conference (WebSci 2013)*.

Radford, A., *et al.*, 2019. Language Models are Unsupervised Multitask Learners. .

Ribeiro, R., Trifan, A., and Neves, A.J.R., 2024. A deep learning approach for transportation mode identification using a transformation of GPS trajectory data features into an image representation. *International Journal of Data Science and Analytics*.

Rzeszewski, M. and Luczys, P., 2018. Care, Indifference and Anxiety—Attitudes toward Location Data in Everyday Life. *ISPRS International Journal of Geo-Information*, 7 (10), 383.

Schuessler, N. and Axhausen, K.W., 2009. Processing raw data from global positioning systems without additional information. *Transportation Research Record*, 2105 (1), 28–36.

Shaw, S.L., Tsou, M.H., and Ye, X., 2016. Human dynamics in the mobile and big data era. *International Journal of Geographical Information Science*, 30 (9), 1687–1693.

Shazeer, N., 2020. GLU Variants Improve Transformer. *arXiv preprint arXiv:2002.05202*.

Shin, K.G., *et al.*, 2012. Privacy protection for users of location-based services. *IEEE Wireless Communications*, 19 (1), 30–39.

Stenneth, L., *et al.*, 2011. Transportation mode detection using mobile phones and GIS information. *In*: *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, Nov.. Chicago Illinois: ACM, 54–63.

Su, J., *et al.*, 2021. RoFormer: Enhanced Transformer with Rotary Position Embedding. *arXiv preprint arXiv:2104.09864*.

Tajalli, M. and Hajbabaie, A., 2017. On the relationships between commuting mode choice and public health. *Journal of Transport & Health*, 4, 267–277.

Tao, Y., Both, A., and Duckham, M., 2018. Analytics of movement through checkpoints. *International Journal of Geographical Information Science*, 32 (7), 1282–1303.

Thompson, S.A. and Warzel, C., 2022. Twelve million phones, one dataset, zero privacy. *Ethics of data and analytics*. Auerbach Publications, 161–169.

Touvron, H., *et al.*, 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Van Hulse, J., Khoshgoftaar, T.M., and Napolitano, A., 2007. Experimental perspectives on learning from imbalanced data. *In*: *Proceedings of the 24th international conference on Machine learning*, 935–942.

Vaswani, A., *et al.*, 2017. Attention Is All You Need. *arXiv preprint arXiv:1706.03762*.

Wang, Z., Chen, F., and Fujiyama, T., 2015. Carbon emission from urban passenger transportation in Beijing. *Transportation Research Part D: Transport and Environment*, 41, 217–227.

Xiao, X., *et al.*, 2025. Personalized tourism recommendation model based on temporal multilayer sequential neural network. *Scientific Reports*, 15 (1), 382.

Xu, F., *et al.*, 2017. Trajectory recovery from ash: User privacy is not preserved in aggregated mobility data. *In*: *Proceedings of the 26th international conference on world wide web*, 1241–1250.

Xue, H., Voutharoja, B.P., and Salim, F.D., 2022. Leveraging language foundation models for human mobility forecasting. *In*: *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '22, Seattle, Washington New York, NY, USA: Association for Computing Machinery.

Yao, Z., *et al.*, 2020. Understanding human activity and urban mobility patterns from massive cellphone data: Platform design and applications. *IEEE Intelligent Transportation Systems Magazine*, 13 (3), 206–219.

Yu, J.J.Q., 2020. Semi-supervised deep ensemble learning for travel mode identification. *Transportation Research Part C: Emerging Technologies*, 112, 120–135.

Yu, W. and Wang, G., 2023. Graph based embedding learning of trajectory data for transportation mode recognition by fusing sequence and dependency relations. *International Journal of Geographical Information Science*, 37 (12), 2514–2537.

Zandbergen, P.A., 2009. Accuracy of iPhone Locations: A Comparison of Assisted GPS, WiFi and Cellular Positioning. *Transactions in GIS*, 13, 5–25.

Zeng, J., *et al.*, 2023. Trajectory-as-a-Sequence: A novel travel mode identification framework. *Transportation Research Part C: Emerging Technologies*, 146, 103957.

Zhao, Z., *et al.*, 2016. Understanding the bias of call detail records in human mobility research. *International Journal of Geographical Information Science*, 30 (9), 1738–1762.

Zheng, Y., *et al.*, 2008. Understanding mobility based on GPS data. *In*: *Proceedings of the 10th international conference on Ubiquitous computing*, Sep.. Seoul Korea: ACM, 312–321.

Zheng, Y., Xie, X., and Ma, W.Y., 2010. GeoLife: A collaborative social networking service among user, location and trajectory.. *IEEE Data Eng. Bull.*, 33 (2), 32–39 Publisher: Citeseer.

Zook, M., Kraak, M.J., and Ahas, R., 2015. Geographies of mobility: applications of location-based data. *International Journal of Geographical Information Science*, 29 (11), 1935–1940.

Appendix A. Data Pre-Processing

Our pre-processing pipeline transformed raw GPS trajectories from both the Geolife and MOBIS datasets into standardized, analysis-ready data suitable for model training. The process involved several key steps:

Data Consolidation: For Geolife, we first converted the original PLT files into CSV format, while MOBIS data was already structured appropriately. Tables A2 and A3 show the initial distribution of transportation modes in the raw datasets.

Distance Calculation: We computed distances between consecutive GPS points using the Haversine formula, which accounts for Earth's curvature:

$$d = 2r \cdot \arcsin\left(\sqrt{\sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right) \tag{A1}$$

where ϕ represents latitude, λ represents longitude (both in radians), and r is Earth's radius (6,371,000 meters).

Speed Derivation: We computed speeds by dividing the distance by the time difference between consecutive points, converting to km/h:

$$v = \frac{d}{\Delta t} \cdot 3.6 \tag{A2}$$

Label Standardization: We harmonized transportation mode labels across datasets, consolidating similar modes (e.g., merging 'taxi' with 'car', and 'subway' with 'train') to create consistent categories across datasets.

Mode-Specific Filtering: We applied speed thresholds for each transportation mode (Table A1) to remove physically implausible values caused by GPS errors or data anomalies.

Trip Quality Control: We removed trips with fewer than three GPS points, as these would depart from real world scenarios and they provide insufficient sequential information for the model to learn meaningful patterns.

This pre-processing reduced the original datasets to five standardized transportation modes (Bike, Bus, Car, Train, and Walk) suitable for cross-dataset modeling. The final datasets used for model training and evaluation contained 144.7 million data points across 1.53 million trips from MOBIS and 4.44 million data points across 9,427 trips from Geolife, as detailed in Table 1.

Table A1. Speed Thresholds for Each Mode

Mode	Min Speed (km/h)	Max Speed (km/h)
Bike	0.5	50
Bus	1.0	120
Car	3.0	180
Train	3.0	350
Walk	0.1	15

Table A2. Summary of the MOBIS Dataset

Mode	Number of Data Points	Number of Unique Trips
Aerialway	109	7
Airplane	718,156	3,185
Bicycle	6,568,587	41,166
Bus	7,450,378	77,361
Car	160,348,026	553,259
Ferry	228	12
LightRail	5,824,637	39,657
RegionalTrain	3,674,625	22,413
Subway	425,223	6,917
Train	10,166,760	34,064
Tram	3,591,472	32,428
Walk	56,539,324	767,923
Total	**255,307,525**	**1,578,392**

Table A3. Summary of the Labeled Geolife Dataset

Mode	Number of Data Points	Number of Unique Trips
Airplane	9,196	14
Bike	951,350	1,555
Boat	3,565	7
Bus	1,271,062	1,851
Car	512,939	782
Motorcycle	336	2
Run	1,975	4
Subway	309,699	613
Taxi	241,404	513
Train	556,397	177
Walk	1,582,693	3,991
Total	**5,440,616**	**9,509**

Appendix B. GPS Tracking on Smartphone Application

Our smartphone application, *CarbonClever* WeChat Mini-Program, implements real-time GPS tracking to collect high-quality mobility data while minimizing battery consumption. When a participant initiates tracking, the application activates the device's location services using a dynamic sampling strategy designed to balance data density and energy efficiency.

The location tracking module periodically queries the device's location sensors at intervals ranging from 10 to 30 seconds, depending on device capability and movement speed. Although the application can technically record location data at one-second intervals, a sampling rate of 10–30 seconds effectively preserves the accuracy of speed estimation while preventing excessive battery drain on users' smartphones.

Each recorded sample captures the user's current geographic coordinates (ϕ, λ) and timestamp (t). As new coordinates are received, the application computes the great-circle distance between consecutive points using the Haversine formula (B1):

$$d = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right), \tag{B1}$$

where d denotes the great-circle distance in meters, r is Earth's radius (6,371,000 meters), and ϕ_1, λ_1, ϕ_2, and λ_2 are the latitudes and longitudes (in radians) of consecutive GPS points. The average speed between two points is then calculated as:

$$v = \frac{d}{\Delta t}, \tag{A2}$$

where Δt represents the elapsed time (in seconds) between successive samples. This real-time computation allows for efficient on-device speed estimation while maintaining privacy, since only scalar speed values—not precise location traces—are required for model input.

These calculations are performed in real time within the application's frontend, allowing users to receive immediate feedback on their trip statistics. Simultaneously, the raw coordinate data, computed speeds, distances, and timestamps are securely transmitted to a cloud database via encrypted API calls. This cloud infrastructure also hosts our pre-trained SpeedTransformer and LSTM-Attention models, enabling real-time transportation mode prediction.

During our experiments, the application's tracking module collected GPS trajectories from a diverse range of smartphone devices and operating systems. Table 5 summarizes the dataset collected through the mini-program, which exhibits natural variations in sampling frequency and signal quality—characteristics typical of real-world usage. This authentic dataset proved invaluable for validating the SpeedTransformer model under conditions that closely mirror practical deployment environments.

Appendix C. Full Grid Hyper-Parameter Space Search Results

Experimental reporting conventions. The following tables enumerate all training and fine-tuning runs conducted in this experiment. To improve readability, we omit dataset size and run identifiers, and instead report the complete set of hyper-parameters that uniquely define each configuration. Unless otherwise specified, all accuracies correspond to the test split (%).

Column definitions. *LR*, *BS*, *DO*, *WD*, *Ep.*, and *Early* denote learning rate, batch size, dropout, weight decay, maximum epochs, and early-stopping patience, respectively. For the LSTM-Attention model, additional columns include *Hidden* (hidden state dimension) and *Layers* (number of stacked LSTM layers). For the SpeedTransformer, columns include *Heads* (attention heads), *d_model* (embedding dimension), *KV Heads* (shared key/value heads when using grouped-query attention), *Warmup* (number of warmup steps), and *Freeze Policy* (which modules were frozen or reinitialized). In fine-tuning tables, *Subset (%)* indicates the percentage of the target training data used (e.g., 15%, 20%, etc.).

Summary of results. Tables C1, C2, C3, C4, C5, C6, C7, and C8 report the results. Across all training and fine-tuning experiments, SpeedTransformer consistently

Table C1. Summary of LSTM–Attention performance trained on MOBIS. All runs use Ep.=50, WD=1e-4, EarlyStop=7, unless noted.

Configuration	LR	Batch Size	Dropout	Hidden Units	Layers	Accuracy (%)
Base (best)	1×10^{-3}	128	0.1	128	2	**92.40**
Smaller batch	1×10^{-3}	64	0.1	128	2	92.20
Higher dropout	1×10^{-3}	64	0.2	128	2	92.15
Deeper network	1×10^{-3}	64	0.1	128	3	92.26
Larger hidden size	1×10^{-3}	64	0.1	256	2	92.04
Higher learning rate	2×10^{-3}	64	0.1	128	2	92.14
Lower learning rate	5×10^{-4}	128	0.1	256	3	92.33
Smaller batch, lower LR	5×10^{-4}	64	0.1	128	2	92.35

achieved higher accuracy and stronger cross-domain generalization than the LSTM-Attention baseline, particularly under limited-data and transfer learning. When trained directly on the Geolife and MOBIS datasets, both models converged to high accuracy (approximately 93–96%). However, notable differences emerged in transfer learning and fine-tuning scenarios.

For in-domain training, SpeedTransformer exhibited optimal performance with a learning rate of 2×10^{-4}, batch size of 512, dropout of 0.1, and embedding dimension $d_{model} = 128$ with 8 attention heads. Accuracy reached 95.97% on Geolife and 94.22% on MOBIS, with stability across modest parameter variations, suggesting strong generalization capacity without overfitting. LSTM-Attention achieved comparable accuracy (around 92.7%) using a learning rate of 1×10^{-3}, hidden dimension 128–256, and two recurrent layers, but its performance was more sensitive to learning rate changes.

In cross-dataset transfer (MOBIS $\rightarrow$ Geolife), fine-tuned SpeedTransformer models surpassed 85% test accuracy under optimal configurations, while LSTM-Attention plateaued around 84%. Among transformer variants, the best results were obtained when the last block was reinitialized (85.7%) or when no layers were frozen (84.2%). By contrast, freezing attention or feed-forward layers caused performance to drop below 65%, confirming that full end-to-end adaptation is necessary for effective transfer across mobility domains.

Fine-tuning from transfer data (MOBIS $\rightarrow$ Real-World App Experiment) further demonstrated the efficiency of the transformer architecture. Even when using only 15% of the experiment dataset, SpeedTransformer achieved 89.1% accuracy, outperforming LSTM-Attention (86.6%) under comparable settings. As the fine-tuning subset increased, SpeedTransformer's accuracy scaled up smoothly, reaching 94.2% with 50% of the target data. Embedding-freezing strategies yielded intermediate performance (up to 87.8%), while partial freezing or warmup schedules offered no clear benefit. These results indicate that SpeedTransformer effectively leverages prior mobility representations with minimal data, whereas LSTM-Attention requires larger sample sizes to reach similar accuracy.

Overall, the experiments reveal three consistent patterns. First, moderate learning rates (2–5×10^{-4}) and full-layer fine-tuning yield the most robust convergence across datasets. Second, SpeedTransformer's attention-based representations exhibit superior transferability and resilience to dataset heterogeneity compared with recurrent encoders (C1). Third, increasing the fine-tuning subset improves performance approximately monotonically, suggesting that the pre-trained mobility embeddings capture domain-invariant movement structures that generalize across tracking environments.

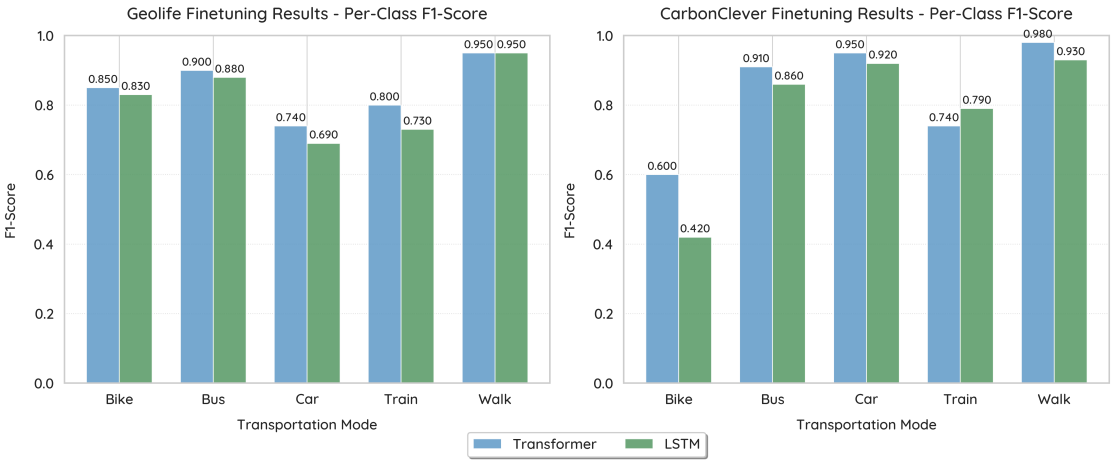

Figure C1. Per class F1-Score for Fine-tuning on Geolife and real-world app data using SpeedTransformer and LSTM. The SpeedTransformer consistently achieves better results than the LSTM-Attention across all classes in both tasks.

Table C2. Summary of SpeedTransformer performance trained on MOBIS. All runs use Ep.=50, WD=1e-4, EarlyStop=7.

Configuration	LR	Batch Size	Dropout	d_model	Heads / KV Heads	Accuracy (%)
Base (best)	1×10^{-4}	512	0.1	128	8 / 4	**94.22**
Higher LR	2×10^{-4}	512	0.1	128	8 / 2	94.20
Larger model	2×10^{-4}	512	0.1	192	12 / 6	93.71
Larger embedding	2×10^{-4}	512	0.1	256	8 / 4	93.54
Smaller batch	2×10^{-4}	1024	0.1	128	8 / 4	93.69
Higher dropout	2×10^{-4}	512	0.2	128	8 / 4	93.54
Higher LR	3×10^{-4}	512	0.1	128	8 / 4	94.09

Table C3. Summary of LSTM–Attention performance trained on Geolife. All runs use Ep.=50, WD=1e-4, EarlyStop=7, unless noted.

Configuration	LR	Batch Size	Dropout	Hidden Units	Layers	Accuracy (%)
Base (best)	1×10^{-3}	128	0.1	128	2	**92.77**
Smaller batch	1×10^{-3}	64	0.1	128	2	91.77
Higher dropout	1×10^{-3}	64	0.2	128	2	92.71
Deeper network	1×10^{-3}	64	0.1	128	3	92.16
Larger hidden size	1×10^{-3}	64	0.1	256	2	92.73
Higher learning rate	2×10^{-3}	64	0.1	128	2	92.56
Lower learning rate	5×10^{-4}	128	0.1	256	3	92.40
Smaller batch, lower LR	5×10^{-4}	64	0.1	128	2	91.63

Appendix D. Input Embeddings

Each scalar speed value s_t is linearly projected into a $d = 128$-dimensional embedding space, producing an input embedding matrix $\mathbf{E} \in \mathbb{R}^{T \times d}$. This projection enables the model to represent one-dimensional scalar speeds within a high-dimensional latent space suitable for Transformer-based sequence modeling:

$$\mathbf{e}_t = \mathbf{W}_e s_t + \mathbf{b}_e, \tag{D1}$$

where s_t denotes the scalar speed at time step t, $\mathbf{W}_e \in \mathbb{R}^{1 \times d}$ is the learnable

Table C4. Summary of SpeedTransformer performance trained on Geolife. All models use Ep.=50, WD=1e-4, EarlyStop=7, unless noted.

Configuration	Learning Rate (LR)	Model Size (d_{model})	Attention Heads (Q/KV)	Accuracy (%)
Base (optimal)	2×10^{-4}	128	8 / 4	**95.97**
Reduced KV heads	2×10^{-4}	128	8 / 2	95.36
Larger model	2×10^{-4}	192	12 / 6	94.72
Larger embedding	2×10^{-4}	256	8 / 4	94.69
Higher LR	3×10^{-4}	128	8 / 4	94.68
Lower LR	1×10^{-4}	128	8 / 4	94.44
Higher batch size	2×10^{-4}	128	8 / 4	92.72
Higher dropout	2×10^{-4}	128	8 / 4	92.70

Table C5. Summary of fine-tuning LSTM-Attention (MOBIS → Geolife). All models use BS=128, DO=0.3, WD=1e-4, Ep.=60, and EarlyStop=7.

Learning Rate (LR)	Hidden Size	Accuracy (%)	Notes
1×10^{-3}	64–256	**84.17**	Optimal configuration
5×10^{-4}	64–256	84.13	Comparable to best
1×10^{-4}	64–256	82.84	Slight underfitting
5×10^{-5}	—	75.47–79.15	Low performance (few epochs)

Table C6. Summary of fine-tuning SpeedTransformer (MOBIS → Geolife). All models use BS=512, WD=1e-4, DO=0.2, Ep.=50, and EarlyStop=7.

Learning Rate (LR)	Best Freeze Policy	Warmup Steps	Accuracy (%)
5×10^{-5}	Reinit last block	0	85.07
1×10^{-4}	Reinit last block	0	85.70
2×10^{-4}	Attention frozen	0	**86.38**
2×10^{-4}	Reinit last block	0	86.30
1×10^{-4}	Embeddings frozen	0	84.22
1×10^{-4}	None (all trainable)	0	84.08
2×10^{-4}	None (all trainable)	0	85.01
5×10^{-5} to 2×10^{-4}	Any freeze w/ warmup (100–500)	—	63–67 (no gain)

Table C7. Summary of LSTM-Attention fine-tuning results from MOBIS → Real-World App Experiment. All runs use 50 epochs, dropout = 0.3, and weight decay = 1e-4.

Subset (%)	LR	Hidden Units	Batch Size	Early Stop	Pat.	Accuracy (%)
15	5×10^{-4}	128	128	10	Pat.10	86.62
20	5×10^{-4}	128	128	10	Pat.10	88.02
30	5×10^{-4}	128	128	10	Pat.10	**88.97**
40	5×10^{-4}	128	128	10	Pat.10	87.45
50	5×10^{-4}	128	128	10	Pat.10	87.57
15	1×10^{-3}	128–256	128	10	Pat.10	86.19
15	1×10^{-4}	128–256	128	10	Pat.10	85.18

weight matrix, and $\mathbf{b}_e \in \mathbb{R}^d$ is the learnable bias vector. The resulting embeddings $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_T]$ serve as the model's input sequence to subsequent positional encoding and self-attention layers.

Table C8. Summary of SpeedTransformer fine-tuning results from MOBIS → Real-World App Experiment. All runs used 50 epochs, dropout = 0.2, and weight decay = 1e-4.

Subset (%)	Learning Rate	Warmup Steps	Freeze Policy	Batch Size	Early Stop	Accuracy (%)
15	5×10^{-4}	0	None	512	Pat.10	89.12
20	5×10^{-4}	0	None	512	Pat.10	82.53
30	5×10^{-4}	0	None	512	Pat.10	**91.15**
40	5×10^{-4}	0	None	512	Pat.10	88.78
50	5×10^{-4}	0	None	512	Pat.10	**94.22**
15	1×10^{-4}	0–100	Attention/Embedding frozen	512	Pat.10	69.3–72.9
15	2×10^{-4}	0–100	Embeddings frozen	512	Pat.10	71.6–80.6
15	5×10^{-4}	0–100	Embeddings frozen	512	Pat.10	77.2–87.8

Appendix E. SwiGLU Activation

Each feed-forward subnetwork adopts the SwiGLU activation (Shazeer 2020), defined as:

$$\text{SwiGLU}(\mathbf{x}) = (\mathbf{x}\mathbf{W}_1) \odot \text{Swish}(\mathbf{x}\mathbf{W}_2)\mathbf{W}_3, \tag{E1}$$

where $\odot$ denotes element-wise multiplication, $\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3$ are learnable weight matrices, and $\text{Swish}(\mathbf{x}) = \mathbf{x}\sigma(\mathbf{x})$ with $\sigma(\cdot)$ being the sigmoid activation function. SwiGLU enhances representational expressivity and improves training stability relative to standard ReLU-based feed-forward layers.

Appendix F. Model Window Size

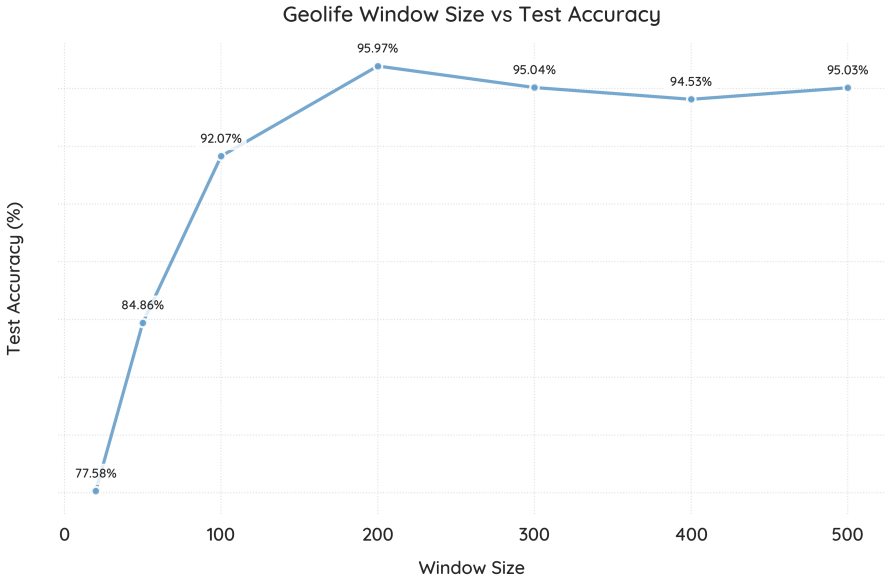

Figure F1. Effect of window size on test accuracy for the Geolife dataset. A window size of 200 provides the optimal balance between temporal context and computational efficiency, achieving the highest accuracy (95.97%).

Figure F1 supports our choice of $T = 200$, showing that performance improves steadily with larger window sizes up to 200, beyond which accuracy plateaus and slightly declines. The 500-sample configuration also required a reduced batch size due to GPU memory

constraints, potentially impacting training stability. Overall, $T = 200$ represents the optional size between accuracy, computational efficiency, and training stability, capturing sufficient motion dynamics without introducing unnecessary redundancy or overfitting.

Table F1. Test accuracy and loss across different window sizes (Geolife dataset).

Window Size	Test Accuracy (%)	Test Loss	Run Name
20	77.58	0.5778	Geolife_ws20_lr2e-4_bs512_h8_d128_kv4_do0.1
50	84.86	0.4043	Geolife_ws50_lr2e-4_bs512_h8_d128_kv4_do0.1
100	92.07	0.2600	Geolife_ws100_lr2e-4_bs512_h8_d128_kv4_do0.1
200	95.97	0.1525	Geolife_ws200_lr2e-4_bs512_h8_d128_kv4_do0.1
300	95.04	0.1753	Geolife_ws300_lr2e-4_bs512_h8_d128_kv4_do0.1
400	94.53	0.1703	Geolife_ws400_lr2e-4_bs256_h8_d128_kv4_do0.1
500	95.03	0.1800	Geolife_ws500_lr2e-4_bs256_h8_d128_kv4_do0.1

Appendix G. SpeedTransformer in Original Transformer Model Architecture

We also conduct experiments using the original Transformer architecture, proposed by Vaswani *et al.* (2017), for a comparison. Although the core architecture shares the transformer encoder backbone with the final model presented in Section 3, this initial version employed sinusoidal positional encodings and a standard feed-forward block instead of Rotary Positional Embeddings (RoPE) and SwiGLU activations. The updated design described in the main text improved both training stability, computational efficiency, and cross-dataset transferability.

The following subsections summarize the original architecture, input encoding strategy, and optimization setup for the model. We provide experiment results, which serves as a reference for ablation comparisons.

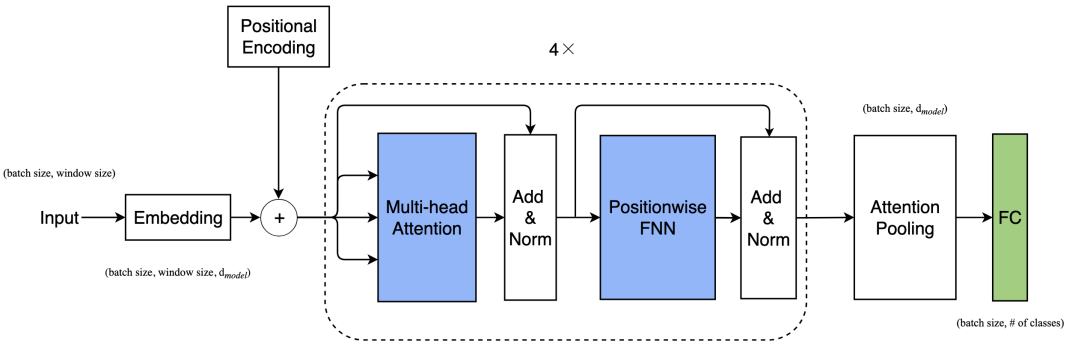

Figure G1. Transformer architecture for transportation mode classification.

The model architecture adapts the transformer encoder framework (Vaswani *et al.* 2017). Figure G1 presents the overall architecture of our model. To incorporate sequential order information, we add sinusoidal positional encoding to these embeddings, shown in Equation G1.

$$\mathbf{PE}(pos, 2i) = \sin\Big(\frac{pos}{10000^{2i/d_{\text{model}}}}\Big), \quad \mathbf{PE}(pos, 2i+1) = \cos\Big(\frac{pos}{10000^{2i/d_{\text{model}}}}\Big) \tag{G1}$$

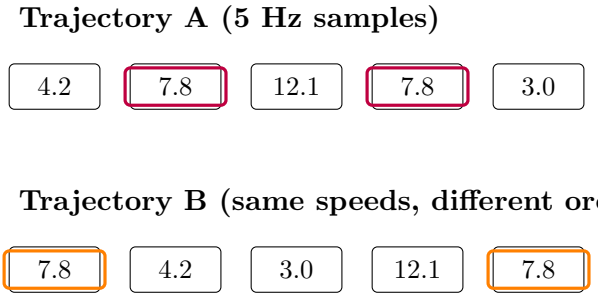

Figure G2. Two GPS velocity sequences share the same multiset of speeds but in different orders. Without position information the embeddings for repeated values (e.g., 7.8 m/s) are identical. With positional encoding we form $\mathbf{z}_t = E(v_t) + P(t)$, allowing attention to model order-dependent patterns.

where *pos* is the position in the sequence, i is the dimension index, and d_{model} is the embedding dimension.

We use the same 200 variable-length sliding window, in consistent with the main model. We projects scalar speed value into high-dimensional vector representations through linear transformation followed by non-linear activation in Equation G2.

$$\mathbf{E} = \text{ReLU}(\mathbf{W}e \cdot \mathbf{S} + \mathbf{b}_e) \tag{G2}$$

where $\mathbf{S}$ represents input speed values and $\mathbf{E}$ the resulting embeddings.

These encoding enables the model to differentiate positions and captures temporal variations that are critical to distinguish transportation modes over time. For example, information such as acceleration patterns can be inferred from sequences of trajectories.

After the positional encoding, the inputs are fed into a self-attention layer, which helps the encoder to check velocity at other positions in the input sequence as it encodes a velocity at a specific location, as shown in Figure G2. Vaswani *et al.* (2017) established that attention mechanisms compute weighted aggregations of value vectors where weights are determined by compatibility scores between query and key vectors, which we adopt the scaled dot-product attention formulation as defined in Equation (1) of Vaswani et al. (2017) and use their definition of scaled dot-product attention in Equation G3.

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\Big(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\Big)\mathbf{V} \tag{G3}$$

In this formulation, $\mathbf{Q}$ represents queries that seek information, $\mathbf{K}$ encodes keys that store information, and $\mathbf{V}$ contains values that are aggregated according to query-key compatibility. For self-attention, all three matrices derive from the same input sequence, which enables each position to attend to all positions within the sequence.

The multi-head attention mechanism extends this concept by computing attention in parallel across different representation subspaces. We follow Vaswani *et al.* (2017) and use their definition of multihead representation in Equation G5.

$$\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\text{head1}, \ldots, \text{head}_h)\mathbf{W}^O \tag{G4}$$

$$\text{head}_i = \text{Attention}(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V) \tag{G5}$$

Here, $\mathbf{W}_i^Q$, $\mathbf{W}_i^K$, and $\mathbf{W}_i^V$ are learned projection matrices that transform the original embeddings into different subspaces, while $\mathbf{W}^O$ projects the concatenated outputs back to the model dimension.

In line with standard deep learning practice, each encoder layer includes a position-wise feed-forward network (FFN) applied independently to every position. The input to the FFN is formed by summing the token embeddings and positional encodings, as shown in Equation G6:

$$\mathbf{z} = \mathbf{x} + \mathbf{PE}, \tag{G6}$$

where $\mathbf{x}$ denotes the token embedding and $\mathbf{PE}$ represents the positional encoding.

Residual connections and layer normalization are applied around each sub-layer to facilitate gradient flow and enhance training stability. To generate a fixed-length representation from variable-length sequences, we employ attention-based pooling. The position-aware input $\mathbf{z}$ is then passed into the feed-forward network (FFN), defined in Equation G7:

$$\mathrm{FFN}(\mathbf{x} + \mathbf{PE}) = \max\left(0, (\mathbf{x} + \mathbf{PE})\mathbf{W}_1 + \mathbf{b}_1\right)\mathbf{W}_2 + \mathbf{b}_2, \tag{G7}$$

where $\mathbf{W}_1$ and $\mathbf{W}_2$ are weight matrices, and $\mathbf{b}_1$ and $\mathbf{b}_2$ are bias terms. To incorporate sequential order information, token embeddings are first augmented with sinusoidal positional encoding through the attention mechanism and softmax normalization. These position-aware inputs are then processed by the FFN in Equation G7, enabling the model to capture complex non-linear transformations of input representations at each position.

The model is trained using the cross-entropy loss function, which quantifies the divergence between the predicted and ground-truth transportation modes, as shown in Equation G8:

$$\mathcal{L} = -\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\log(p_{i,c}), \tag{G8}$$

where N is the number of samples, C is the number of classes, $y_{i,c} \in \{0, 1\}$ denotes the ground-truth label (1 if sample i belongs to class c), and $p_{i,c}$ is the predicted probability of class c for sample i.

Final classification is performed through a linear projection of the pooled representation, followed by a softmax activation to generate a probability distribution over transportation modes, as defined in Equation G9:

$$p(y = c \mid \mathbf{c}) = \frac{\exp(\mathbf{w}_c^\top \mathbf{c} + b_c)}{\sum_{j=1}^{C}\exp(\mathbf{w}_j^\top \mathbf{c} + b_j)}, \tag{G9}$$

where $\mathbf{c}$ denotes the pooled feature vector, and $\mathbf{w}_c$ and b_c represent the weight vector and bias term for class c, respectively.

We implement AdamW optimization with weight decay, learning rate scheduling, dropout regularization, and gradient clipping. This architecture enables efficient dis-

covery of complex temporal patterns in speed data, capturing the distinctive signatures of different transportation modes without requiring additional input features or preprocessing steps.

Table G1. Per-class accuracy (recall) of LSTM and Transformer models across datasets. Each model was trained on Geolife and MOBIS, and fine-tuned from MOBIS to Geolife and Mini-Program datasets.

Model	**Training Setup**	**Bike**	**Bus**	**Car**	**Train**	**Walk**
LSTM	Geolife	0.92	0.86	0.86	0.92	0.99
	MOBIS	0.61	0.84	0.99	0.28	0.99
	MOBIS → Geolife	0.59	0.84	0.57	0.79	0.72
	MOBIS → Mini-Program	0.39	0.69	0.93	0.79	0.86
Transformer	Geolife	0.93	0.96	0.92	0.87	0.99
	MOBIS	0.78	0.88	0.98	0.43	0.99
	MOBIS → Geolife	0.75	0.88	0.75	0.80	0.99
	MOBIS → Mini-Program	0.40	0.70	1.00	0.43	0.99

As shown in Table G1, the Transformer consistently outperforms the LSTM across all datasets and transfer setups, confirming its superior ability to capture temporal dependencies and generalize across domains.

Appendix H. Rule-based Model

Many existing studies on transportation mode detection employ simple, rule-based models combined with dense GPS trajectory data (Huang *et al.* 2019). A rule-based model relies only on heuristic rules and typically does not require auxiliary GIS data or long GPS sequences. As a result, such models are computationally efficient and simple to implement. However, their performance is highly sensitive to data outliers, which frequently occur in urban environments. For instance, a slow-moving car in heavy traffic may be indistinguishable from a fast-walking pedestrian when classification depends exclusively on speed-based heuristics.

To illustrate this limitation, we designed a simple rule-based baseline model in consultation with a local transportation expert prior to our real-world experiment. The heuristic rules, summarized in Table H1, are expressed in meters per second (m/s). The heuristic rule operates hierarchically: it first uses the 95th-percentile window speed to differentiate between slower (walk/bike) and faster (motorized/rail) modes, and then applies stop ratio and acceleration variability to distinguish buses from cars.

We evaluated this rule-based model on both the Geolife and MOBIS datasets, using the same train test split used in other models in the main text. As shown in Table H2, the rule-based model performs poorly on the Geolife data but achieves relatively better results on the MOBIS dataset. A closer inspection reveals that the model's apparent success on MOBIS is largely driven by the dataset's high class imbalance and the heuristic rule's effectiveness in distinguishing between *car* and *walk* modes—two categories that together account for 89.2% of all records. As a result, the overall accuracy appears relatively high (0.6030). In contrast, the same heuristic rule performs dismally on the Geolife data, where mobility patterns differ substantially, yielding an overall accuracy of only 0.0154. While a more carefully calibrated heuristic rule could certainly improve performance, our broader argument still holds: rule-based models are inherently rigid and ill-suited for

Rule	Threshold (m/s)	Description / Condition for Mode Assignment
`walk_p95_max`	≤ 1.75	Windows with 95th-percentile speed below 1.75 m/s (≈ 6.3 km/h) are classified as *walk*.
`bike_p95_max`	≤ 2.08	Windows with 95th-percentile speed between 1.75–2.08 m/s (≈ 6.3–7.5 km/h) are classified as *bike*.
`road_p95_min`	≥ 2.08	Windows exceeding 2.08 m/s (≈ 7.5 km/h) are treated as motorized road transport (*bus* or *car*).
`rail_p95_min`	≥ 41.7	Windows with 95th-percentile speed above 41.7 m/s (≈ 150 km/h) are classified as *rail*.
`stop_thresh`	< 0.3	Speeds below 0.3 m/s are considered "stopped" for computing the stop ratio.
`bus_stop_ratio_min`	≥ 0.20	Within the motorized range, if $\geq 20\%$ of time is spent stopped and acceleration variability is low, classify as *bus*.
`accel_std_split`	< 0.6	Within the motorized range, smoother acceleration (< 0.6 m/s^2 std.) indicates *bus*; higher variability indicates *car*.

Table H1. Heuristic Rules for Transportation Mode Classification.

transportation mode detection in settings where mobility data are both abundant and behaviorally complex.

Dataset	Mode	Precision	Recall	F1-score	Support	Accuracy
Geolife	Bike	0.0062	0.0169	0.0091	3,200	
	Bus	0.0000	0.0000	0.0000	4,863	
	Car	0.0226	0.1106	0.0376	2,251	
	Train	0.0000	0.0000	0.0000	4,119	
	Walk	0.0000	0.0000	0.0000	5,225	
	Macro Avg.	0.0058	0.0255	0.0093		
	Weighted Avg.	0.0036	0.0154	0.0058		
	Overall Accuracy				19,658	**0.0154**
MOBIS	Bike	0.0015	0.0050	0.0023	17,389	
	Bus	0.0000	0.0000	0.0000	14,474	
	Car	0.6561	0.9136	0.7638	336,860	
	Train	0.0000	0.0000	0.0000	27,607	
	Walk	0.9951	0.1603	0.2762	155,581	
	Macro Avg.	0.3305	0.2158	0.2085		
	Weighted Avg.	0.6810	0.6030	0.5441		
	Overall Accuracy				551,911	**0.6030**

Table H2. Rule-Based Model on Geolife and MOBIS Datasets.

Appendix I. SpeedTransformer's Computational Efficiency

Like most deep learning architectures, SpeedTransformer requires substantially higher computational resources than traditional, non–deep learning models. Nevertheless, through our adaptation of the Grouped-Query Attention (GQA) structure, the model achieves comparatively high computational efficiency.

As illustrated in Figure I1, during a full training session with four processes one a single A100 GPU node, the model maintained an average GPU utilization of approximately 67% (peaking at 96%) across all devices. Meanwhile, host memory usage averaged around 52.9 GB (with a peak of 61.9 GB) over a training period of roughly 126.5 minutes. These measurements suggest effective multi-GPU-process utilization and stable memory management, consistent with a compute-bound workload characteristic of Transformer-based architectures.

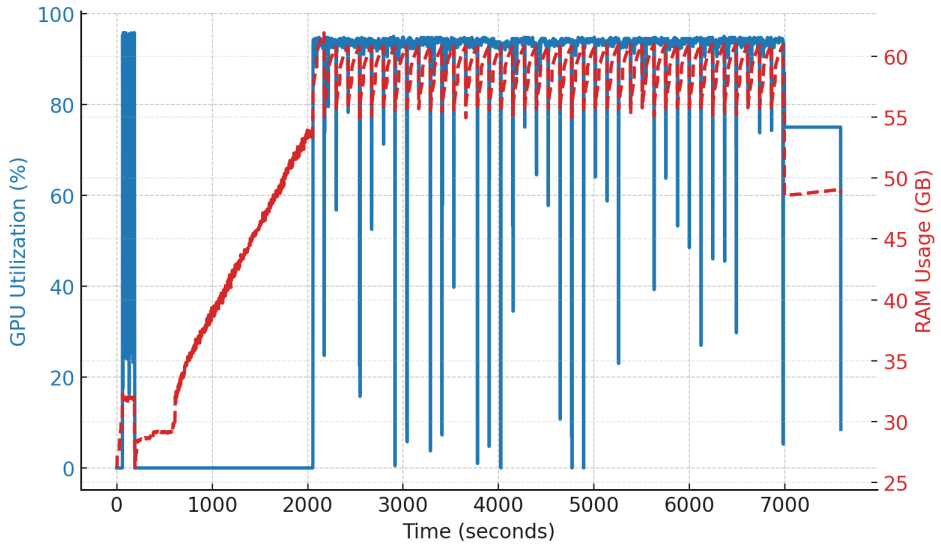

Figure I1. GPU and memory utilization of the Transformer model during training.

Appendix J. IRB Protocol and Ethics Approval

Following the Committee on Publication Ethics (COPE) guidance, we include our IRB protocol and ethical statement here, concerning our real-world experiment. In respect for conciseness, we summarize our IRB protocol and its approval below.

This study was approved by the Institutional Review Board (IRB) at Duke Kunshan University (DKU) shortly before the carried out experiment. The protocol (version dated June 23, 2022) received clearance to conduct randomized experiments using a WeChat-based mini-program developed by the research team. Participants were recruited from cities in China, through both online advertisements and QR-code posters in focus groups.

Eligible participants (aged 18 and above) provided brief informed consent, consistent with standard practices in the industry, before engaging in the study. The consent process was integrated into the mini-program and clearly stated that no personal identifiers (e.g., name, phone number, WeChat ID) would be collected. Instead, a pseudonymous device ID was used to generate a non-identifiable case ID for analysis. Demographic information is purposefully not collected to protect human subjects' privacy. Minimal geographic and

behavioral data were collected for the purpose of estimating participants' daily carbon footprints.

The study involved daily interaction with the mini-program over a one-month period, during which participants received varying forms of informational stimulus—ranging from government policy content to scientific facts and social cues. Weekly questionnaires measured participants' willingness to pay for carbon reduction.

Data were stored securely on DKU-managed servers, and all members of the research team were Chinese citizens, in compliance with China's data sovereignty regulations. No audio, video, or photographic data were collected. Participants had the option to donate or receive a small monetary compensation for their time.

This research posed minimal risk to participants and involved no clinical interventions. The IRB ensured that appropriate data security and privacy measures were in place.

We include the full approval letter in the journal's manuscript portal.