MobTA: Bus-Conditioned Zero-Shot Trajectory Generation via Task Arithmetic

Shuai Liu
Nanyang Technological University
Singapore
Shuai004@e.ntu.edu.sg

Ning Cao
Nanyang Technological University
Singapore
ning.cao@ntu.edu.sg

Yue Jiang
Nanyang Technological University
Singapore
yue013@e.ntu.edu.sg

Gao Cong
Nanyang Technological University
Singapore
gaocong@ntu.edu.sg

Abstract

Mobility trajectory data provide essential support for smart city applications. However, such data are often difficult to obtain. Meanwhile, most existing trajectory generation methods implicitly assume that at least a subset of real mobility data from target city is available, which limits their applicability in data-inaccessible scenarios. In this work, we propose a new problem setting, called bus-conditioned zero-shot trajectory generation, where no target city mobility trajectories are accessible. Trajectory generation relies solely on source city mobility data and publicly available bus timetables from both the source and target city. Under this setting, we propose MobTA, the first approach to introduce task arithmetic into trajectory generation. MobTA models the parameter shift from bus-timetable-based trajectory generation to mobility trajectory generation in source city, and applies this shift to target city through arithmetic operations on task vectors. This enables trajectory generation that reflects target-city mobility patterns without requiring any real mobility data from it. Furthermore, we theoretically analyze MobTA's stability across base and instruction-tuned LLMs. Extensive experiments show that MobTA significantly outperforms existing methods, and achieves performance close to models finetuned using target city mobility trajectories.

1 Introduction

Human mobility trajectories characterize the spatiotemporal behavior of urban residents Liu et al. [2024], Wei et al. [2025], supporting various smart city applications such as urban planning Zhou et al. [2024b], traffic management Lu et al. [2019], and disease control Pollington et al. [2025]. In practice, however, mobility trajectories are often difficult to access. First, such data are typically collected by ride-hailing platforms or telecommunication operators, and are rarely publicly accessible due to commercial restrictions. Second, mobility trajectories contain sensitive personal information, and their sharing are subject to strict constraints. These challenges motivate trajectory generation, which aims to model the spatiotemporal patterns of human mobility based on available trajectory data, and to generate synthetic trajectories that closely match real-world mobility Chen et al. [2025a].

Although existing trajectory generation models have made notable progress, most of them rely on an implicit assumption: at least a subset of target-city mobility trajectories is accessible. However, satisfying this prerequisite is often impractical. Beyond the aforementioned commercial restrictions and privacy concerns, many cities are unable to collect and maintain mobility trajectory data due

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

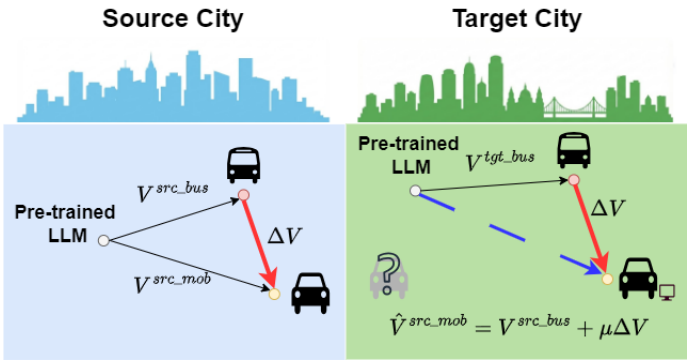


Figure 1: Conceptual overview of MobTA. It uses task arithmetic to model the parameter shift from bus-timetable-based to mobility trajectory generation in source city, and applies this shift to target city, where only bus timetables are accessible, to enable mobility trajectory generation for target city.

to the substantial costs of sensor deployment and data management Kong et al. [2023]. In contrast, bus timetable data offer clear advantages in terms of accessibility. As non-sensitive public data, bus timetables are widely available across mapping platforms like Google Maps, Amap, and Naver Map. Moovit [1] collects bus timetable data across multiple countries. Each bus timetable consists of an ordered sequence of bus stops and their scheduled arrival times, which can be regarded as a structured yet coarse-grained form of trajectory data. This accessibility creates a practical class of cities where bus timetables are publicly available while mobility trajectories are not, such as Shenyang, Doha, Riyadh, and Cape Town. For such cities, existing methods that require target-city mobility trajectories are difficult to apply directly. Nevertheless, bus timetables alone cannot effectively substitute for mobility trajectories. Zhang et al. [2018] has shown that while the two types of trajectories exhibit certain correlations, gaps remain in terms of spatial coverage and distribution. Constrained by fixed routes, bus timetables fail to capture the flexible travel behaviors that occur beyond bus routes. This motivates our core research question: in the absence of real mobility trajectories from target city, how can we leverage mobility trajectories from other cities, together with publicly available bus timetables, to generate synthetic trajectories that reflect the mobility patterns of target city? We define this challenging yet practical problem setting as **bus-conditioned zero-shot trajectory generation**.

To address this problem, our core intuition is that bus routes are designed to cover major urban travel demands, aiming to improve overall mobility efficiency and alleviate traffic congestion Vuchic [2017]. Although bus timetables cannot fully capture urban mobility behaviors, they represent a mobility backbone in terms of spatial layout and temporal regularity, reflecting activity hotspots and dominant travel directions. Mobility trajectories, in turn, capture flexible and fine-grained mobility patterns that complement this backbone. We posit that the discrepancy between bus timetables and mobility trajectories is a structured deviation primarily arising from first-and-last-mile travel and personalized short-range movements Lu et al. [2024]. As such behaviors often share structural regularities across cities Gonzalez et al. [2008], this discrepancy exhibits a certain degree of transferability, motivating a mechanism that explicitly models and transferring such structured deviations.

Motivated by this intuition, we draw inspiration from task arithmetic Ilharco et al. [2022], Li et al. [2025b].Its core concept is task vector, which is defined as the parameter difference between a fine-tuned model and its original pre-trained model. Prior studies Bhardwaj et al. [2024], Zhang et al. [2024] have shown that linear arithmetic operations on task vectors exhibits great effectiveness in out-of-domain generalization and has achieved success across a variety of tasks Parović et al. [2024], Braga et al. [2025]. These findings suggest that appropriate combinations of task vectors can induce task-relevant behavior even when the target task or domain is not directly optimized.

Based on these insights, we propose **Mob**ility **T**ask **A**rithmetic (MobTA), a novel method for bus-conditioned zero-shot trajectory generation. It constructs three task vectors corresponding to mobility trajectory generation task in source city, and bus-timetable-based trajectory generation task in both source and target city. As shown in Fig. 1, MobTA models the parameter shift from bus-timetable-based trajectory generation to mobility trajectory generation in source city. It then transfers this shift to the target city's bus-timetable-based trajectory generation task vector. The resulting task

[1] `https://moovitapp.com/`

vector enables the model to approximate target-city mobility patterns without accessing any target city mobility trajectories. Furthermore, we provide a theoretical analysis showing why MobTA remains stable across base and instruction-tuned LLMs. Experiments show that MobTA consistently outperforms existing baselines, and achieves performance close to that obtained by fine-tuning models with target city mobility trajectories. In summary, our contributions are:

- We introduce bus-conditioned zero-shot trajectory generation, a challenging yet practical problem setting where target city mobility trajectories are entirely inaccessible, and trajectory generation relies solely on mobility trajectories from other cities, and public bus timetables. This offers a new perspective for trajectory generation in data-inaccessible cities.
- We propose MobTA, which introduces task arithmetic into this new problem setting. It captures the parameter shift from bus-timetable-based to mobility trajectory generation in the source city, and applies this shift to the target city for mobility trajectory generation. We further theoretically analyze MobTA's stability across base and instruction-tuned LLMs.
- Extensive experiments show that MobTA outperforms existing baselines, and approaches the performance of models fine-tuned with target-city mobility trajectories.

2 Related Work

2.1 Trajectory Generation

Trajectory generation aims to synthesize mobility data by learning spatiotemporal patterns from real trajectories. Early methods model human mobility using Markov processes Mathew et al. [2012], Baratchi et al. [2014], gravity model Pappalardo et al. [2015], or social interaction graphs Borrel et al. [2008]. While such methods provide interpretable formulations, they struggle to capture the complex patterns and heterogeneous behaviors in real mobility data. In recent years, learning-based approaches have become the dominant paradigm. Methods based on generative adversarial networks (GANs) leverage adversarial training to align generated distributions with real data Rao et al. [2020], Cao and Li [2021], while variational autoencoders (VAEs) model trajectories through structured latent representations Huang et al. [2019], Ding et al. [2019]. Recently, diffusion models have been introduced to trajectory generation. They produce high-fidelity trajectories through iterative denoising Chu et al. [2024], Zhu et al. [2024]. For a detailed review of trajectory generation models, please refer to Appendix C.1. Despite their effectiveness, most existing methods implicitly assume that at least a subset of target city mobility trajectories is accessible. In practice, however, real mobility trajectory data are often difficult to obtain due to privacy concerns or commercial restrictions. As a result, existing approaches cannot be directly applied to such data-inaccessible cities.

2.2 Task Arithmetic

Task arithmetic is a parameter-efficient paradigm for model editing, enabling the injection of new abilities or the mitigation of undesirable behaviors without costly retraining Yadav et al. [2023], Bhardwaj et al. [2024]. Its core concept is the task vector, which is defined as the parameter difference between a fine-tuned model and its pre-trained counterpart Ilharco et al. [2022], Zhang et al. [2023]. More importantly, task arithmetic exhibits remarkable potential for out-of-distribution (OOD) generalization, enabling models to adapt to novel tasks without additional training Chronopoulou et al. [2024], Parović et al. [2024] or narrowing distribution gaps between synthetic and real data Su et al. [2024], Braga et al. [2025]. Theoretically, Li et al. [2025b] validate that linear compositions of task vectors can provably guide models toward such generalization, providing a rigorous foundation for this paradigm. A detailed review of task arithmetic is provided in Appendix. C.2.

3 Preliminary

3.1 Problem Formulation

*Definition 1 (**Mobility Trajectory**)* A mobility trajectory is an ordered sequence of location-time pairs, denoted as $\tau^{mob} = \{(l_1, t_1), (l_2, t_2), \ldots, (l_n, t_n)\}$, where l_i is the spatial location at time t_i. Given a city c, its mobility trajectory data is denoted as $\mathcal{D}_c^{mob} = \{\tau_1^{mob}, \tau_2^{mob}, \ldots, \tau_N^{mob}\}$.

*Definition 2 (**Bus Timetable**)* A bus timetable is an ordered sequence of station-arrival time pairs $\tau^{bus} = \{(s_1, \tilde{t}_1), (s_2, \tilde{t}_2), \ldots, (s_m, \tilde{t}_m)\}$, where s_i is the i-th bus stop on the route and $\tilde{t}_i$ is its scheduled arrival time. Given a city c, its bus timetable data is denoted as $\mathcal{D}_c^{bus} = \{\tau_1^{bus}, \tau_2^{bus}, \ldots, \tau_N^{bus}\}$.

*Definition 3 (**Bus Station Sequence**)* Given a bus timetable τ^{bus}, its corresponding bus station sequence is defined as the ordered list of bus stations $r^{bus} = (s_1, s_2, \ldots, s_m)$.

*Definition 4 (**Bus-Conditioned Zero-Shot Trajectory Generation**)* Bus-conditioned zero-shot trajectory generation considers a problem setting where no mobility trajectories from target city c_{tgt} are accessible during training. Available data consist of source city data $(\mathcal{D}_{c_{src}}^{mob}, \mathcal{D}_{c_{src}}^{bus})$ and target city bus timetables $\mathcal{D}_{c_{tgt}}^{bus}$. The objective is to obtain a trajectory generation model $\mathcal{G}$ that produces synthetic trajectories $\hat{\mathcal{D}}_{c_{tgt}}^{mob}$ that reflect the mobility patterns of c_{tgt}. Formally,

$$\hat{\mathcal{D}}_{c_{tgt}}^{mob} = \mathcal{G}(\mathcal{D}_{c_{src}}^{mob}, \mathcal{D}_{c_{src}}^{bus}, \mathcal{D}_{c_{tgt}}^{bus}) \tag{1}$$

3.2 Empirical Observations on Bus Timetables and Mobility Trajectories

We empirically analyze bus timetables and their relationship with mobility trajectories. We focus on two aspects: (1) the long-term comparability of bus systems, and (2) the spatiotemporal correlations between bus timetables and mobility trajectories. These findings provide practical motivation for transferring the parameter shift from bus-timetable-based to mobility trajectory generation in MobTA.

3.2.1 Long-Term Comparability of Bus Systems

We conduct a comparative analysis using Shanghai bus data in 2015 and 2025. Specifically, we first evaluate the spatial stability of bus station sequences. We then examine operational-time comparability based on service time windows and peak-hour headway statistics.

Table 1: Spatial Stability of Shanghai bus station sequences in 2015 and 2025.

Category	LCS(↑)	Jac(↑)	EDR(↓)	DTW(↓)	Dice(↑)
Entire City	0.8109	0.6975	0.2522	0.1355	0.7937
Within Outer Ring Road	0.8546	0.7459	0.2011	0.1062	0.8383

As shown in Table 1, bus stop sequences exhibit a high degree of consistency over time. When the analysis is restricted to area within Outer Ring Road (the study area of this work), the similarity becomes even more pronounced. These results indicate that bus station sequences preserve a stable spatial structure over long time span. Detailed experimental settings are provided in Sec. E.1.

We further examine operational-time comparability of Shanghai bus systems between 2015 and 2025. The 2015 bus data provide first and last service times for each route, allowing us to compare service windows for matched routes. Following the same route-matching protocol as in Table 1, the average service window overlap is 96.86%. In addition, since the 2015 bus data do not contain bus stop arrival times, we cannot compute 2015 headways. Therefore, we use public transport service requirements in Shanghai's 12th Five-Year Plan [2] as a proxy for 2015 bus service frequency: the peak-hour headway for regular buses should not exceed 8 minutes in the central city and 10 minutes in suburban built-up areas. For comparison, the average peak-hour headway in 2025 data is 10.76 minutes. Considering that the 2025 data cover a broader spatial scope and include more peripheral routes, this value is on the same order of magnitude as the 2015 planning standard. These results indicate that Shanghai's bus system is stable not only in station sequences, but also in operational-time characteristics.

3.2.2 Spatiotemporal Correlations Between Bus Timetables and Mobility Trajectories

Long-term comparability alone does not show whether bus timetables reflect mobility patterns. We therefore examine the spatiotemporal correlations between bus timetables and mobility trajectories.

From a spatial perspective, Table 2 first compares bus timetables with mobility trajectories across cities. The results show that bus timetables and mobility trajectories exhibit relatively high consistency,

[2] `https://www.shanghai.gov.cn/nw30688/20200820/0001-30688_32660.html`

indicating overlap in major regions. However, this overlap is substantial but still partial, suggesting that bus timetables and mobility trajectories do not share identical spatial patterns. Regarding point and route coverage, while a large proportion of mobility trajectories spatially overlap with bus timetable coverage, a non-negligible fraction of mobility activities occur outside this coverage.

From a temporal perspective, Table 2 further reports the temporal Pearson correlation between bus timetables and mobility trajectories for each city, with their specific hourly distributions depicted in Fig.5-7. The results indicate that bus timetables and mobility trajectories exhibit similar flow rhythms, particularly during daytime hours. In late-night periods when bus services are sparse, mobility trajectories remain relatively active, leading to noticeable deviations.

Table 2: Spatiotemporal Relationship Between Bus Timetables and Mobility Trajectories

City	$JSD_c(\downarrow)$	Cos(↑)	Pear-S(↑)	PCov(↑)	RCov(↑)	Pear-T(↑)
SH	0.1142	0.7595	0.6592	86.86	86.87	0.8180
WX	0.2064	0.6595	0.6077	55.71	51.42	0.9180
SG	0.0802	0.8726	0.8457	80.08	63.82	0.9702

Overall, bus timetables are strongly correlated with real mobility trajectories in both spatial and temporal dimensions, but they cannot fully capture mobility patterns. These findings suggest that bus timetables can serve as an effective structural reference for trajectory generation when real target-city mobility data are inaccessible. Meanwhile, Table 2 shows that the three cities are not homogeneous bus-support cases, but represent different levels of bus coverage for mobility trajectories. In particular, Wuxi serves as a representative city with weaker bus coverage, where a substantial fraction of mobility activities falls outside bus timetable coverage. Detailed experimental settings are in Appendix. E.2.

4 Method

4.1 Framework Overview

In light of the observations in Sec.3.2, our key intuition is that bus timetables can serve as a stable mobility backbone in terms of spatial layout and temporal regularity, while mobility trajectories capture flexible and fine-grained mobility patterns that complement this backbone. Their discrepancy exhibits a certain degree of transferability across cities. When target city mobility trajectories are inaccessible, directly learning a mobility trajectory generation model is infeasible. Moreover, training solely on bus timetables tends to overfit bus-specific patterns and fails to generalize. A more viable perspective is to explicitly model the systematic discrepancy between bus-timetable-based trajectory generation and mobility trajectory generation, and to treat this discrepancy as transferable knowledge. Based on this idea, MobTA, as shown in Fig. 2, leverages task arithmetic to transfer such knowledge across cities, enabling mobility trajectory generation under the bus-conditioned zero-shot setting.

4.2 Task Vectors for Trajectory Generation

To apply effective task arithmetic to trajectory generation, it is critical to ensure that task vectors are aligned in parameter space, such that vector arithmetic corresponds to consistent functional edits rather than representation mismatches. Therefore, we derive task vectors for each trajectory generation task by fine-tuning under a unified trajectory representation and serialization scheme.

4.2.1 Unified Trajectory Representation and Serialization

We adopt a unified spatial discretization and tokenization strategy by partitioning the study region of each city into regular grids with the same spatial resolution. We designate the grid containing geometric center of the study region as the anchor grid, indexed as `<LOC_0>`. Grid indices are then assigned to surrounding grids using a spiral indexing strategy that expands outward from the center in a clockwise manner (see Appendix F). This indexing strategy assigns city-specific locations to a shared spatial-token vocabulary, allowing task vectors from different cities to be defined in the same parameter space, which is a prerequisite for task arithmetic. Importantly, the same token index is not assumed to carry identical urban semantics across cities. MobTA transfers the parameter shift from bus-timetable-based to mobility-trajectory generation, rather than source-city spatial semantics.

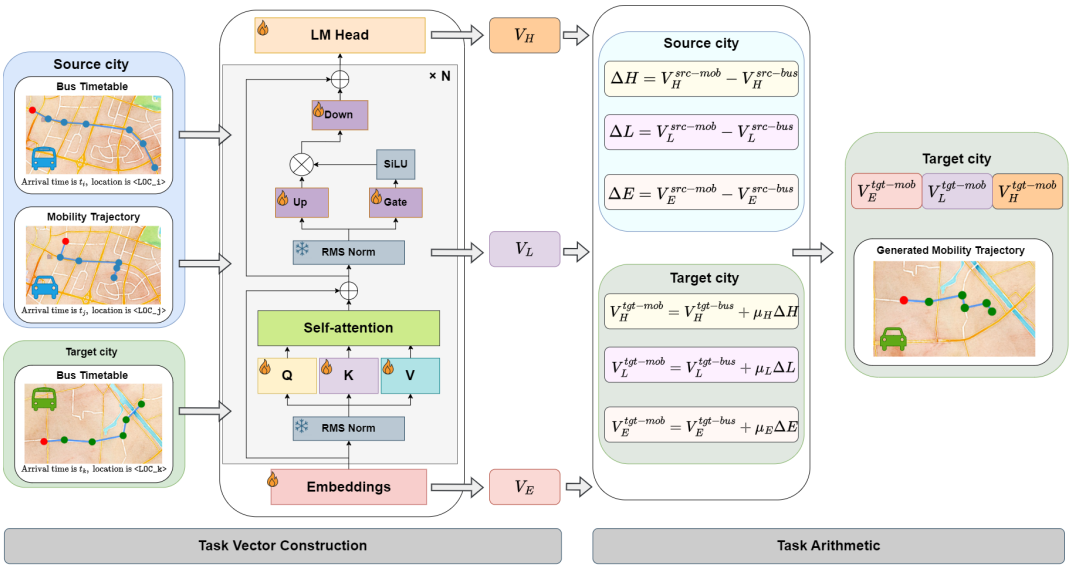


Figure 2: Overall framework of MobTA. MobTA constructs three task vectors corresponding to mobility trajectory generation in the source city, as well as bus-timetable-based trajectory generation in both the source and target cities. It then models the parameter shift from bus-timetable-based to mobility trajectory generation in the source city and applies this shift to the target city's bus-timetable-based trajectory generation task vector, enabling mobility trajectory generation for the target city.

To enable trajectory generation, we serialize each trajectory into a textual sequence. Each record is formatted as "arrival time is t_i, location is <LOC_i> #", and the entire trajectory is concatenated sequentially. For example, a two-record trajectory is serialized as "arrival time is 498, location is <LOC_244> # arrival time is 499, location is <LOC_124> # <EOS>". Here t_i is a numerical token representing the time-of-day. All three training tasks use this same chronological sequence format and the same causal next-token prediction objective; only the training dataset differs. The complete sequence is the language-modeling text, without a separate instruction–response template. This allows the model to learn spatiotemporal patterns directly into its parameters.

4.2.2 Task Vector Construction

Under the unified trajectory representation and serialization, we obtain task vectors by fine-tuning LLMs on trajectory generation tasks over bus timetables and mobility trajectories. For a given pretrained backbone, we independently fine-tune three checkpoints from the same initialization: one on source-city mobility trajectories, one on source-city bus-timetable trajectories, and one on target-city bus-timetable trajectories. These are independent task-specific training runs, not sequential intermediate models; none uses target-city mobility trajectories. We jointly consider parameter shifts in the embedding layer, LLM backbone, and the output head as components of task vector. Together, these parameter shifts capture complementary aspects of trajectory generation behavior and form a unified representation of task-specific adaptations.

For embedding and output layers, we initialize the parameters of newly added spatial tokens using the global mean of pre-trained vocabulary embeddings (see Appendix P.3 and P.4 for details), and perform full-parameter fine-tuning. Unlike Geollama Li et al. [2025c], which encodes locations as numerical identifiers that are tokenized into multiple sub-tokens, our approach assigns a dedicated token to each spatial grid. This keeps the parameter updates for each spatial grid localized to its corresponding token, yielding clearer task vectors for arithmetic operations.

For LLM backbone, we employ LoRA to adapt both the Q, K, and V projections in attention layers and the up, down, and gate projections in feed-forward networks. This design stems from the insight that trajectory generation relies not only on modelling sequential dependencies, but also on the nonlinear transformations to shape spatiotemporal patterns Dong et al. [2021], Geva et al. [2022].

For each task, we compute the parameter difference between its fine-tuned checkpoint and the shared pretrained initialization, and decompose this task vector into $\{\mathbf{V}_E, \mathbf{V}_L, \mathbf{V}_H\}$, corresponding to the embedding layer, the effective backbone weight updates induced by LoRA, and the output head, respectively.

4.3 Task Arithmetic for Bus-Guided Zero-shot Trajectory Generation

Given three task vectors derived from source-city bus-timetable-based generation, source-city mobility trajectory generation, and target-city bus-timetable-based generation, MobTA constructs the target-city mobility-generation task vector via arithmetic operations. Formally, let $\mathbf{V}^{src-bus}$, $\mathbf{V}^{src-mob}$, and $\mathbf{V}^{tgt-bus}$ denote these three task vectors, the resulting task vector $\hat{\mathbf{V}}^{\text{tgt-mob}}$ is computed as:

$$\hat{\mathbf{V}}_E^{\text{tgt-mob}} = \mathbf{V}_E^{\text{tgt-bus}} + \mu_E \left(\mathbf{V}_E^{\text{src-mob}} - \mathbf{V}_E^{\text{src-bus}}\right), \tag{2}$$

$$\hat{\mathbf{V}}_L^{\text{tgt-mob}} = \mathbf{V}_L^{\text{tgt-bus}} + \mu_L \left(\mathbf{V}_L^{\text{src-mob}} - \mathbf{V}_L^{\text{src-bus}}\right), \tag{3}$$

$$\hat{\mathbf{V}}_H^{\text{tgt-mob}} = \mathbf{V}_H^{\text{tgt-bus}} + \mu_H \left(\mathbf{V}_H^{\text{src-mob}} - \mathbf{V}_H^{\text{src-bus}}\right), \tag{4}$$

where μ_E, μ_L, and μ_H control the magnitude of transferred parameter shift within each subspace. The resulting sub task vectors are added to the pre-trained model parameters to obtain a model for target-city mobility trajectory generation. This process does not involve any further gradient based optimization and does not require access to real mobility trajectories from target city, thereby providing an effective solution for bus-guided zero-shot trajectory generation. At inference time, the starting time, starting location, and target trajectory length are sampled from empirical distributions derived only from target-city bus timetables. We serialize the initial time–location record as a prompt, for example, "arrival time is 580, location is <LOC_50> #". The composed model then generates subsequent time–location records autoregressively until it reaches the sampled target length or generates <EOS>.

4.4 Theoretical Analysis of MobTA's Stability under Base and Instruction-Tuned LLMs

Prior task-arithmetic studies mainly derive task vectors from base LLMs Kawamoto et al. [2025], leaving instruction-tuned models less explored. A common concern is that instruction tuning introduces additional parameter shifts for instruction following and safety alignment, which may interfere with task-specific parameter changes and undermine task-vector additivity Chen et al. [2025b].

In MobTA, these concerns are mitigated, as trajectory generation in MobTA relies on fixed structured sequences and newly introduced spatial tokens, rather than instruction understanding or multi-turn interactions. This suggests that task-specific parameter changes are concentrated in a constrained task-relevant subspace, while the parameter shift introduced by instruction tuning has only a small projection onto this subspace. Below, we formalize this intuition to explain the similar generation behavior of task vectors derived from base and instruction-tuned LLMs.

4.4.1 Task-Relevant Parameter Subspace

We first formalize the parameter directions that influence trajectory generation in MobTA. Let θ denote the model parameters. Given a serialized trajectory sequence x, the logit for predicting token T is denoted as $z_T(\theta; x)$. In MobTA, trajectory generation is formulated as predicting trajectory-related tokens $T \in \mathcal{V}_{\text{traj}}$, consisting of newly introduced spatial location tokens and time-related numerical tokens. We define the task-relevant parameter subspace for MobTA's trajectory generation, denoted as $\mathcal{T}_{\text{traj}}$, as parameter directions spanned by the gradients associated with trajectory-related tokens:

$$\mathcal{T}_{\text{traj}} := \text{span}\left\{\nabla_\theta z_T(\theta; x) \;\middle|\; x \sim \mathcal{X}_{\text{traj}},\ T \in \mathcal{V}_{\text{traj}}, \theta = \theta_0^{(B)}\right\}, \tag{5}$$

where $\mathcal{X}_{\text{traj}}$ is the distribution of serialized trajectory sequences, and $\theta_0^{(B)}$ is the base LLM initialization used as the reference point for the subspace. We further denote $\mathbf{P}_{\text{traj}}$ as the orthogonal projection operator onto $\mathcal{T}_{\text{traj}}$. Although $\mathcal{T}_{\text{traj}}$ lies in the full parameter space, trajectory generation in MobTA induces sparse and structured gradients: directions associated with instruction following or safety alignment are negligible, while significant gradients concentrate around trajectory-specific tokens. Thus, $\mathcal{T}_{\text{traj}}$ is expected to be low-dimensional and focused on trajectory generation.

4.4.2 Projection of base and instruction-tuned LLMs onto Task-Relevant Subspace

Let $\theta_0^{(B)}$ and $\theta_0^{(I)}$ denote the initialization parameters of base and instruction-tuned LLM, and define $\delta_0 := \theta_0^{(I)} - \theta_0^{(B)}$. Although δ_0 may be substantial in the full parameter space, instruction tuning primarily affects instruction-following and safety-alignment behavior, whereas MobTA's trajectory generation is concentrated in $\mathcal{T}_{\mathrm{traj}}$. We therefore assume that δ_0 has a small projection onto $\mathcal{T}_{\mathrm{traj}}$, so base and instruction-tuned models remain close in this subspace:

$$\frac{\|P_{\mathrm{traj}}\,\delta_0\|}{\|\delta_0\|} \le \epsilon, \quad \epsilon \ll 1. \tag{6}$$

Thus, base and instruction-tuned models remain close in $\mathcal{T}_{\mathrm{traj}}$, supporting stable task arithmetic in MobTA. Empirically validation of this assumption can be seen in Appendix G.3.

4.4.3 Projection Stability of Task Vectors

For a trajectory dataset $\mathcal{D}$ and a given model's initialization $\theta_0^{(\cdot)}, (\cdot) \in \{B, I\}$, we define the corresponding task vector as the parameter shift obtained via fine-tuning:

$$\mathbf{V}(\theta_0^{(\cdot)}; \mathcal{D}) = \theta^*(\theta_0^{(\cdot)}; \mathcal{D}) - \theta_0^{(\cdot)}, \tag{7}$$

where $\theta^*(\theta_0^{(\cdot)}; \mathcal{D})$ is the parameters after finetuning. Under a small-step, parameter-efficient fine-tuning strategy, this parameter shift admits a standard first-order approximation:

$$\mathbf{V}(\theta_0; \mathcal{D}) \approx -\eta \nabla_\theta \mathcal{L}(\theta_0; \mathcal{D}), \tag{8}$$

where $\mathcal{L}(\theta_0; \mathcal{D})$ is the training loss for trajectory generation and η denotes the effective step size. For dataset $\mathcal{D}$, we project the base–instruction task-vector difference onto $\mathcal{T}_{\mathrm{traj}}$:

$$\mathbf{P}_{\mathrm{traj}}\big(\mathbf{V}_B(\mathcal{D}) - \mathbf{V}_I(\mathcal{D})\big) \approx -\eta\, \mathbf{P}_{\mathrm{traj}}\big(\nabla_\theta \mathcal{L}(\theta_0^{(B)}; \mathcal{D}) - \nabla_\theta \mathcal{L}(\theta_0^{(I)}; \mathcal{D})\big) \tag{9}$$

Under the first-order approximation, this projected difference remains small due to the small projection of δ_0 onto $\mathcal{T}_{\mathrm{traj}}$ and weak interaction with its orthogonal complement; see Appendix G for details.

4.4.4 Stability of MobTA Task Arithmetic

Within $\mathcal{T}_{\mathrm{traj}}$, the difference between $\hat{\mathbf{V}}^{\text{tgt-mob}}$ from base and instruction-tuned LLMs satisfies

$$\begin{aligned}\mathbf{P}_{\mathrm{traj}}\big(\hat{\mathbf{V}}_B^{\text{tgt-mob}} - \hat{\mathbf{V}}_I^{\text{tgt-mob}}\big) = \mathbf{P}_{\mathrm{traj}}\big(\mathbf{V}_B(\mathcal{D}_{c_{tgt}}^{bus}) - \mathbf{V}_I(\mathcal{D}_{c_{tgt}}^{bus})\big) \\ + \mu\Big(\mathbf{P}_{\mathrm{traj}}\big(\mathbf{V}_B(\mathcal{D}_{c_{src}}^{mob}) - \mathbf{V}_I(\mathcal{D}_{c_{src}}^{mob})\big) \\ - \mathbf{P}_{\mathrm{traj}}\big(\mathbf{V}_B(\mathcal{D}_{c_{src}}^{bus}) - \mathbf{V}_I(\mathcal{D}_{c_{src}}^{bus})\big)\Big).\end{aligned} \tag{10}$$

As each right term is small in $\mathcal{T}_{\mathrm{traj}}$, their linear combination remains small in $\mathcal{T}_{\mathrm{traj}}$. Thus, $\hat{\mathbf{V}}_B^{\text{tgt-mob}}$ and $\hat{\mathbf{V}}_I^{\text{tgt-mob}}$ induce similar trajectory-token logits. This step-wise closeness propagates through autoregressive generation, explaining the stability of MobTA across base and instruction-tuned LLMs.

5 Experiment

We evaluate MobTA in the bus-conditioned zero-shot setting through generation performance and robustness across LLMs. Additional experiments, including weak-bus robustness experiments, behavior under varying spatial coverage, efficiency comparison, and ablation studies are detailed in Appendix K, L, O and M. To examine progressively weaker bus signals, we also vary the fraction of retained target-city bus routes from 100% to 1% (Appendix K.1). MobTA's distributional errors generally increase as route coverage decreases, but remain below COLA's at every reported retention level. Point coverage alone can be misleading: for SH→SG at 25% retained routes, PCov is 60.43% while mean RCov is 43.97%, alongside substantial degradation in spatial JSD. Code is available at `https://github.com/liuwj2000/MobTA`. Practical implementation considerations and technical nuances are in Appendix P.

5.1 Experimental Setup

We conduct experiments on Shanghai (SH), Wuxi (WX), and Singapore (SG), using LLaMA-3.2-3B as the default LLM. Experiments are conducted on four NVIDIA RTX A5000 GPUs. Dataset details, baseline details, evaluation metrics, hyperparameter settings are in Appendix H, I, J, and P.

Table 3: Distributional discrepancy (JSD) between bus timetables and real mobility trajectories.

City	Dist.	Rad.	Dur.	Loc.
Shanghai (SH)	0.1765	0.1777	0.0374	0.1643
Wuxi (WX)	0.3055	0.2841	0.0379	0.2483
Singapore (SG)	0.1747	0.1356	0.0361	0.0930

5.2 Trajectory Generation Performance Comparison

Table 3 shows clear distributional gaps between bus timetables and real mobility trajectories, especially in distance, radius, and TrajLoc, further indicating that bus timetables alone are insufficient to capture mobility patterns. Table 4 then shows that non-transfer baselines trained only on bus timetables remain constrained by bus coverage and fail to close these gaps. COLA also struggles on location-related metrics because it can only generate locations observed during training (i.e., those covered by bus timetables). These results show that existing baselines are not effective to handle bus-conditioned zero-shot trajectory generation. In contrast, MobTA achieves consistently better performance, and approaches the performance of models fine-tuned with target-city mobility trajectories. Its low Dist. and Rad. indicate that MobTA matches the physical scale of mobility trajectories, which suggests that the generated trajectories are less likely to be dominated by unrealistic long-range spatial transitions.

5.3 Robustness across LLM Architectures

Table 5 compares the performance of MobTA across different LLMs under the WX $\rightarrow$ SH setting. MobTA maintains consistent generation quality on both LLaMA and Qwen, indicating that the proposed task arithmetic approach is not tied to a specific LLM architecture. Furthermore, within each model family, base and instruction-tuned variants exhibit only negligible performance differences. This empirical observation well support the theoretical analysis in Sec. 4.4

6 Conclusion

This paper introduces bus-conditioned zero-shot trajectory generation, a challenging and practical problem setting in which no real mobility trajectories from the target city are accessible. Under this setting, we propose MobTA, the first approach to apply task arithmetic to trajectory generation. By explicitly transferring the parameter shift from bus-timetable-based trajectory generation to mobility trajectory generation, MobTA enables trajectory generation that closely match real mobility patterns, relying solely on target-city bus timetables and source-city data. From a theoretical perspective, we analyze the stability of MobTA across base and instruction-tuned LLMs. Extensive experiments show MobTA's effectiveness.

Limitations. The route-removal experiment is a controlled proxy for weakened bus signals, not a validation in a car-dominant city. Its PCov and RCov diagnostics use real target-city mobility trajectories for evaluation and cannot be computed in the intended zero-shot deployment setting. The observed performance degradation around low RCov values should therefore not be interpreted as a universal deployment threshold. Meanwhile, MobTA is not intended to cover all cities. For private-vehicle-dominant cities, bus timetables may not sufficiently characterize the dominant urban mobility. Thus, an important future direction is to extend MobTA to other public structural signals, such as road networks, traffic sensor statistics, or POI distributions. Another direction is to extend MobTA to substantially larger metropolitan cities by expanding the shared spatial-token vocabulary or developing hierarchical discretization schemes, so that task vectors can still be constructed within a consistent parameter space. In addition, it would be valuable to explore whether bus stop attributes like station names or topological relationships can further enhance trajectory generation performance remains an interesting direction. Finally, incorporating explicit physical feasibility constraints, such as speed limits or road-network reachability, could further improve per-trajectory plausibility.

Table 4: Trajectory generation performance measured by JSD between generated and real mobility trajectories. Smaller values indicate better performance. Geollama$^+$ (mob) here stands for the upper bound which is finetuned using real target city mobility trajectories.

	SH				**WX**				**SG**			
Method	Dist.	Rad.	Dur.	Loc.	Dist.	Rad.	Dur.	Loc.	Dist.	Rad.	Dur.	Loc.
TimeGeo	0.3555	0.4009	0.3023	0.3150	0.2955	0.2881	0.2260	0.2062	0.4325	0.4817	0.1810	0.3856
MoveSim	0.2789	0.3954	0.1161	0.1408	0.3857	0.3552	0.1596	0.1118	0.2892	0.3171	0.0700	0.2317
TSG	0.3149	0.2763	0.0912	0.1800	0.3969	0.2656	0.1506	0.2059	0.3306	0.3928	0.0633	0.3357
TrajGDM	0.2238	0.1913	0.0512	0.1067	0.2591	0.1844	0.0837	0.0901	0.2355	0.2769	0.0351	0.1864
DiffTraj	0.2763	0.2124	0.0673	0.1156	0.3209	0.2149	0.0748	0.1742	0.2951	0.3080	0.0477	0.2156
TrajGen	0.2880	0.2381	0.0692	0.1824	0.3701	0.3029	0.1155	0.2093	0.3074	0.3347	0.0489	0.3389
Traveller	0.2167	0.1746	0.0615	0.0980	0.2726	0.1730	0.1021	0.1366	0.2325	0.2476	0.0433	0.2120
Geollama$^+$ (bus)	0.2058	0.1493	0.0342	0.1175	0.2936	0.1684	0.0917	0.0997	0.2193	0.2692	0.0471	0.2416
COLA (SH→)	–	–	–	–	0.1651	0.1308	0.0529	0.1056	0.2170	0.1844	0.0436	0.1193
COLA (WX→)	0.1831	0.1564	0.0344	0.1544	–	–	–	–	0.2539	0.2161	0.0713	0.1842
COLA (SG→)	0.1610	0.1497	0.0423	0.1364	0.1920	0.1544	0.0587	0.1101	–	–	–	–
MobTA (SH→)	–	–	–	–	0.0952	0.0576	**0.0472**	0.0664	**0.0855**	**0.0469**	**0.0387**	**0.0961**
MobTA (WX→)	0.0969	0.0639	**0.0206**	**0.0487**	–	–	–	–	0.1144	0.0656	0.0417	0.1289
MobTA (SG→)	**0.0749**	**0.0374**	0.0311	0.0552	**0.0847**	**0.0539**	0.0508	**0.0597**	–	–	–	–
Geollama$^+$ (mob)	0.0515	0.0284	0.0192	0.0293	0.0487	0.0372	0.0286	0.0387	0.0715	0.0877	0.0312	0.0577

Table 5: Performance of different LLMs on WX → SH.

Model	Dist.	Rad.	Dur.	Loc.
LLaMA-3.2-3B	0.0969	0.0639	0.0206	0.0487
Qwen3-1.7B-Base	0.0846	0.0645	0.0208	0.0662
LLaMA-3.2-3B-Instruct	0.0926	0.0601	0.0231	0.0496
Qwen3-1.7B	0.0830	0.0628	0.0198	0.0532

References

Cuauhtemoc Anda, Ning Cao, Shuai Liu, Shaowei Ying, and Gao Cong. Personalized and on-device trajectory mobility prediction. In *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Human Mobility Prediction Challenge*, pages 42–45, 2024.

Mitra Baratchi, Nirvana Meratnia, Paul JM Havinga, Andrew K Skidmore, and Bert AKG Toxopeus. A hierarchical hidden semi-markov model for modeling mobility data. In *Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing*, pages 401–412, 2014.

Rishabh Bhardwaj, Duc Anh Do, and Soujanya Poria. Language models are homer simpson! safety re-alignment of fine-tuned language models through task arithmetic. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14138–14149, 2024.

Vincent Bindschaedler and Reza Shokri. Synthesizing plausible privacy-preserving location traces. In *2016 IEEE symposium on security and privacy (SP)*, pages 546–563. IEEE, 2016.

Vincent Borrel, Franck Legendre, Marcelo Dias De Amorim, and Serge Fdida. Simps: Using sociology for personal mobility. *IEEE/ACM transactions on networking*, 17(3):831–842, 2008.

Marco Braga, Pranav Kasela, Alessandro Raganato, and Gabriella Pasi. Investigating task arithmetic for zero-shot information retrieval. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2738–2743, 2025.

Chu Cao and Mo Li. Generating mobility trajectories with retained data utility. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 2610–2620, 2021.

Ji Cao, Tongya Zheng, Qinghong Guo, Yu Wang, Junshu Dai, Shunyu Liu, Jie Yang, Jie Song, and Mingli Song. Holistic semantic representation for navigational trajectory generation. *arXiv preprint arXiv:2501.02737*, 2025a.

Ning Cao, Yuanlong Chen, Yuxiao Dou, Shuai Liu, and Gao Cong. Multiple calibration for post-disaster origin–destination reconstruction. Accepted for presentation at the 4th ACM SIGSPATIAL International Workshop on Human Mobility Prediction (HuMob 2026), 2026. URL `https://takayabe0505.github.io/humob-2026/`.

Zhongcai Cao, Kang Liu, Xin Jin, Li Ning, Ling Yin, and Feng Lu. Stage: a spatiotemporal-knowledge enhanced multi-task generative adversarial network (gan) for trajectory generation. *International Journal of Geographical Information Science*, 39(5):1100–1127, 2025b.

Wei Chen, Yuxuan Liang, Yuanshao Zhu, Yanchuan Chang, Kang Luo, Haomin Wen, Lei Li, Yanwei Yu, Qingsong Wen, Chao Chen, et al. Deep learning for trajectory data management and mining: A survey and beyond. *arXiv preprint arXiv:2403.14151*, 2024.

Xin Chen, Chengrui Huang, Chenhao Wang, and Lisi Chen. Trajectory generation: a survey on methods and techniques. *GeoInformatica*, pages 1–26, 2025a.

Yan-Lun Chen, Yi-Ru Wei, Chia-Yi Hsu, Chia-Mu Yu, Chun-Ying Huang, Ying-Dar Lin, Yu-Sung Wu, and Wei-Bin Lee. Layer-aware task arithmetic: disentangling task-specific and instruction-following knowledge. *arXiv preprint arXiv:2502.20186*, 3, 2025b.

Yile Chen, Yicheng Tao, Yue Jiang, Shuai Liu, Han Yu, and Gao Cong. Enhancing large language models for mobility analytics with semantic location tokenization. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 262–273, 2025c.

Alexandra Chronopoulou, Jonas Pfeiffer, Joshua Maynez, Xinyi Wang, Sebastian Ruder, and Priyanka Agrawal. Language and task arithmetic with parameter-efficient layers for zero-shot summarization. In *Proceedings of the Fourth Workshop on Multilingual Representation Learning (MRL 2024)*, pages 114–126, 2024.

Chen Chu, Hengcai Zhang, and Feng Lu. Trajgdm: A new trajectory foundation model for simulating human mobility. In *Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems*, pages 1–2, 2023.

Chen Chu, Hengcai Zhang, Peixiao Wang, and Feng Lu. Simulating human mobility with a trajectory generation framework based on diffusion model. *International Journal of Geographical Information Science*, 38(5):847–878, 2024.

Rhonda Daniels and Corinne Mulley. Explaining walking distance to public transport: The dominance of public transport supply. *Journal of Transport and Land Use*, 6(2):5–20, 2013.

Wenhao Ding, Wenshuo Wang, and Ding Zhao. A multi-vehicle trajectories generator to simulate vehicle-to-vehicle encountering scenarios. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 4255–4261. IEEE, 2019.

Yihe Dong, Jean-Baptiste Cordonnier, and Andreas Loukas. Attention is not all you need: Pure attention loses rank doubly exponentially with depth. In *International conference on machine learning*, pages 2793–2803. PMLR, 2021.

Justin Elarde, Joon-Seok Kim, Hamdi Kavak, Andreas Züfle, and Taylor Anderson. Change of human mobility during covid-19: A united states case study. *PloS one*, 16(11):e0259031, 2021.

Jie Feng, Zeyu Yang, Fengli Xu, Haisu Yu, Mudan Wang, and Yong Li. Learning to simulate human mobility. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 3426–3433, 2020.

Lei Gao, Yue Niu, Tingting Tang, Salman Avestimehr, and Murali Annavaram. Ethos: Rectifying language models in orthogonal parameter space. *arXiv preprint arXiv:2403.08994*, 2024.

Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *Proceedings of the 2022 conference on empirical methods in natural language processing*, pages 30–45, 2022.

Marta C Gonzalez, Cesar A Hidalgo, and Albert-Laszlo Barabasi. Understanding individual human mobility patterns. *nature*, 453(7196):779–782, 2008.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

Tao Hu, Siqin Wang, Bing She, Mengxi Zhang, Xiao Huang, Yunhe Cui, Jacob Khuri, Yaxin Hu, Xiaokang Fu, Xiaoyue Wang, et al. Human mobility data in the covid-19 pandemic: characteristics, applications, and challenges. *International Journal of Digital Earth*, 14(9):1126–1147, 2021.

Dou Huang, Xuan Song, Zipei Fan, Renhe Jiang, Ryosuke Shibasaki, Yu Zhang, Haizhong Wang, and Yugo Kato. A variational autoencoder based generative model of urban human mobility. In *2019 IEEE conference on multimedia information processing and retrieval (MIPR)*, pages 425–430. IEEE, 2019.

Shih-Cheng Huang, Pin-Zu Li, Yu-Chi Hsu, Kuang-Ming Chen, Yu Tung Lin, Shih-Kai Hsiao, Richard Tsai, and Hung-Yi Lee. Chat vector: A simple approach to equip llms with instruction following and model alignment in new languages. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10943–10959, 2024.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*, 2022.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Shan Jiang, Yingxiang Yang, Siddharth Gupta, Daniele Veneziano, Shounak Athavale, and Marta C González. The timegeo modeling framework for urban mobility without travel surveys. *Proceedings of the National Academy of Sciences*, 113(37):E5370–E5378, 2016.

Chenlu Ju, Jiaxin Liu, Shobhit Sinha, Hao Xue, and Flora Salim. Trajllm: A modular llm-enhanced agent-based framework for realistic human trajectory simulation. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 2847–2850, 2025.

Kazuhiko Kawamoto, Atsuhiro Endo, and Hiroshi Kera. Cross-model transfer of task vectors via few-shot orthogonal alignment. *arXiv preprint arXiv:2505.12021*, 2025.

Rihong Ke, Kuang Wang, Qi Liu, Ling Liu, and Pu Wang. Uncovering urban tourist mobility patterns based on large-scale mobile phone data. *Journal of Transport Geography*, 130:104496, 2026.

Junghwan Kim and Mei-Po Kwan. The impact of the covid-19 pandemic on people's mobility: A longitudinal study of the us from march to september of 2020. *Journal of transport geography*, 93: 103039, 2021.

Akihiro Kobayashi, Naoto Takeda, Yudai Yamazaki, and Daisuke Kamisaka. Modeling and generating human mobility trajectories using transformer with day encoding. In *Proceedings of the 1st International Workshop on the Human Mobility Prediction Challenge*, pages 7–10, 2023.

Xiangjie Kong, Qiao Chen, Mingliang Hou, Hui Wang, and Feng Xia. Mobility trajectory generation: a survey. *Artificial Intelligence Review*, 56(Suppl 3):3057–3098, 2023.

Vaibhav Kulkarni and Benoît Garbinato. Generating synthetic mobility traffic using rnns. In *Proceedings of the 1st Workshop on Artificial Intelligence and Deep Learning for Geographic Knowledge Discovery*, pages 1–4, 2017.

Kyunghan Lee, Seongik Hong, Seong Joon Kim, Injong Rhee, and Song Chong. Slaw: A new mobility model for human walks. In *IEEE INFOCOM 2009*, pages 855–863. IEEE, 2009.

Bobo Li, Xudong Han, Jiang Liu, Yuzhe Ding, Liqiang Jing, Zhaoqi Zhang, Jinheng Li, Xinya Du, Fei Li, Meishan Zhang, et al. Event extraction in large language model. *arXiv preprint arXiv:2512.19537*, 2025a.

Hongkang Li, Yihua Zhang, Shuai Zhang, Meng Wang, Sijia Liu, and Pin-Yu Chen. When is task vector provably effective for model editing? a generalization analysis of nonlinear transformers. *arXiv preprint arXiv:2504.10957*, 2025b.

Siyu Li, Toan Tran, Haowen Lin, John Krumm, Cyrus Shahabi, Lingyi Zhao, Khurram Shafique, and Li Xiong. Geo-llama: Leveraging llms for human mobility trajectory generation with constraints. In *2025 26th IEEE International Conference on Mobile Data Management (MDM)*, pages 20–31. IEEE, 2025c.

Shuai Liu, Ning Cao, Yile Chen, Yue Jiang, and Gao Cong. nextlocllm: next location prediction using llms. *arXiv preprint arXiv:2410.09129*, 2024.

Shuai Liu, Ning Cao, Yile Chen, Yue Jiang, and Gao Cong. Mixture-of-experts for personalized and semantic-aware next location prediction. *arXiv preprint arXiv:2505.24597*, 2025.

Shuai Liu, Ning Cao, and Gao Cong. Post-disaster od flow reconstruction via calendar-aware bridging and flow-difference completion. Accepted for presentation at the 4th ACM SIGSPATIAL International Workshop on Human Mobility Prediction (HuMob 2026), 2026. URL `https://takayabe0505.github.io/humob-2026/`.

Gongyuan Lu, Yu Marco Nie, Xiaobo Liu, and Denghui Li. Trajectory-based traffic management inside an autonomous vehicle zone. *Transportation Research Part B: Methodological*, 120:76–98, 2019.

Ying Lu, Anthony Kimpton, Carlo G Prato, Neil Sipe, and Jonathan Corcoran. First and last mile travel mode choice: A systematic review of the empirical literature. *International Journal of Sustainable Transportation*, 18(1):1–14, 2024.

Yuxiao Luo, Songming Zhang, Kang Liu, Yang Xu, and Ling Yin. Traveller: Travel-pattern aware trajectory generation via autoregressive diffusion models. *Information Fusion*, page 103766, 2025.

Sadhika Malladi, Alexander Wettig, Dingli Yu, Danqi Chen, and Sanjeev Arora. A kernel-based view of language model fine-tuning. In *International Conference on Machine Learning*, pages 23610–23641. PMLR, 2023.

Wesley Mathew, Ruben Raposo, and Bruno Martins. Predicting future locations with hidden markov models. In *Proceedings of the 2012 ACM conference on ubiquitous computing*, pages 911–918, 2012.

Alessandro Mei and Julinda Stefa. Swim: A simple model to generate small mobile worlds. In *IEEE INFOCOM 2009*, pages 2106–2113. IEEE, 2009.

Kun Ouyang, Reza Shokri, David S Rosenblum, and Wenzhuo Yang. A non-parametric generative model for human trajectories. In *IJCAI*, volume 18, pages 3812–3817, 2018.

Luca Pappalardo, Filippo Simini, Salvatore Rinzivillo, Dino Pedreschi, Fosca Giannotti, and Albert-László Barabási. Returners and explorers dichotomy in human mobility. *Nature communications*, 6(1):8166, 2015.

Marinela Parović, Ivan Vulić, and Anna Korhonen. Investigating the potential of task arithmetic for cross-lingual transfer. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 124–137, 2024.

Freya Pollington, Spiros C Denaxas, Kezhi Li, Johan H Thygesen, Georgios Lyratzopoulos, and Becky White. Evaluation of trajectory analysis for disease risk assessment: a scoping review. *Journal of the American Medical Informatics Association*, page ocaf208, 2025.

Jinmeng Rao, Song Gao, Yuhao Kang, and Qunying Huang. Lstm-trajgan: A deep learning approach to trajectory privacy protection. *arXiv preprint arXiv:2006.10521*, 2020.

Xuan Rao, Shuo Shang, Renhe Jiang, Peng Han, and Lisi Chen. Seed: Bridging sequence and diffusion models for road trajectory generation. In *Proceedings of the ACM on Web Conference 2025*, pages 2007–2017, 2025.

Chenyang Shao, Fengli Xu, Bingbing Fan, Jingtao Ding, Yuan Yuan, Meng Wang, and Yong Li. Beyond imitation: Generating human mobility from context-aware reasoning with large language models. *CoRR*, 2024.

Chaoming Song, Tal Koren, Pu Wang, and Albert-László Barabási. Modelling the scaling properties of human mobility. *Nature physics*, 6(10):818–823, 2010.

Yiwen Song, Jingtao Ding, Jian Yuan, Qingmin Liao, and Yong Li. Controllable human trajectory generation using profile-guided latent diffusion. *ACM Transactions on Knowledge Discovery from Data*, 19(1):1–25, 2024.

Hsuan Su, Hua Farn, Fan-Yun Sun, Shang-Tse Chen, and Hung-yi Lee. Task arithmetic can mitigate synthetic-to-real gap in automatic speech recognition. *arXiv preprint arXiv:2406.02925*, 2024.

Peizhi Tang, Chuang Yang, Tong Xing, Xiaohang Xu, Renhe Jiang, and Kaoru Sezaki. Instruction-tuning llama-3-8b excels in city-scale mobility prediction. In *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Human Mobility Prediction Challenge*, pages 1–4, 2024.

Zhenyu Tao, Wei Xu, and Xiaohu You. Map2traj: Street map piloted zero-shot trajectory generation with diffusion model. *arXiv preprint arXiv:2407.19765*, 2024.

Nikhil Tetali, Stefan Popa, Tomas Horak, and Miroslav Svitek. Developing a geometric logistics network for efficient goods distribution in prague. In *2025 Smart City Symposium Prague (SCSP)*, pages 1–7. IEEE, 2025.

Vukan R Vuchic. *Urban transit: operations, planning, and economics*. John Wiley & Sons, 2017.

Jiawei Wang, Renhe Jiang, Chuang Yang, Zengqing Wu, Makoto Onizuka, Ryosuke Shibasaki, Noboru Koshizuka, and Chuan Xiao. Large language models as urban residents: An llm agent framework for personal mobility generation. *Advances in Neural Information Processing Systems*, 37:124547–124574, 2024a.

Jingyuan Wang, Yujing Lin, and Yudong Li. Gtg: Generalizable trajectory generation model for urban mobility. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 834–842, 2025.

Xingrui Wang, Xinyu Liu, Ziteng Lu, and Hanfang Yang. Large scale gps trajectory generation using map based on two stage gan. *Journal of Data Science*, 19(1):126–141, 2021.

Yu Wang, Tongya Zheng, Yuxuan Liang, Shunyu Liu, and Mingli Song. Cola: Cross-city mobility transformer for human trajectory simulation. In *Proceedings of the ACM Web Conference 2024*, pages 3509–3520, 2024b.

Tonglong Wei, Youfang Lin, Shengnan Guo, Yan Lin, Yiheng Huang, Chenyang Xiang, Yuqing Bai, and Huaiyu Wan. Diff-rntraj: A structure-aware diffusion model for road network-constrained trajectory generation. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

Tonglong Wei, Yan Lin, Zeyu Zhou, Haomin Wen, Jilin Hu, Shengnan Guo, Youfang Lin, Gao Cong, and Huaiyu Wan. Transfertraj: A vehicle trajectory learning model for region and task transferability. *arXiv preprint arXiv:2505.12672*, 2025.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. *Advances in Neural Information Processing Systems*, 36:7093–7115, 2023.

Frederic Z Zhang, Paul Albert, Cristian Rodriguez-Opazo, Anton van den Hengel, and Ehsan Abbasnejad. Knowledge composition using task vectors with learned anisotropic scaling. *Advances in Neural Information Processing Systems*, 37:67319–67354, 2024.

Jinghan Zhang, Junteng Liu, Junxian He, et al. Composing parameter-efficient modules with arithmetic operation. *Advances in Neural Information Processing Systems*, 36:12589–12610, 2023.

Xiaohu Zhang, Yang Xu, Wei Tu, and Carlo Ratti. Do different datasets tell the same story about urban mobility—a comparative study of public transit and taxi usage. *Journal of Transport Geography*, 70:78–90, 2018.

Zhaoqi Zhang, Miao Xie, Pasquale Balsebre, Weiming Huang, Siqiang Luo, and Gao Cong. Urban-mfm: Spatial graph-based multiscale foundation models for learning generalized urban representation. *IEEE Transactions on Knowledge and Data Engineering*, 2026a.

Zhaoqi Zhang, Miao Xie, Yi Li, Linyou Cai, Siqiang Luo, and Gao Cong. Causalpoi: Spatio-temporal graph-based causal modeling for cold-start poi check-in forecasting. *arXiv preprint arXiv:2606.05413*, 2026b.

Fan Zhou, Xin Liu, Kunpeng Zhang, and Goce Trajcevski. Toward discriminating and synthesizing motion traces using deep probabilistic generative models. *IEEE Transactions on Neural Networks and Learning Systems*, 32(6):2401–2414, 2020.

Yuyan Zhou, Liang Song, Bingning Wang, and Weipeng Chen. Metagpt: Merging large language models using model exclusive task arithmetic. *arXiv preprint arXiv:2406.11385*, 2024a.

Zhilun Zhou, Yuming Lin, Depeng Jin, and Yong Li. Large language model for participatory urban planning. *arXiv preprint arXiv:2402.17161*, 2024b.

Weiping Zhu, Renhe Jiang, Dongyuan Li, Xiangjie Kong, and Xuan Song. Trajectory generative models: a survey from unconditional and conditional perspectives. *GeoInformatica*, 29(4):1033–1065, 2025.

Yuanshao Zhu, Yongchao Ye, Shiyao Zhang, Xiangyu Zhao, and James Yu. Difftraj: Generating gps trajectory with diffusion probabilistic model. *Advances in Neural Information Processing Systems*, 36:65168–65188, 2023.

Yuanshao Zhu, James Jianqiao Yu, Xiangyu Zhao, Qidong Liu, Yongchao Ye, Wei Chen, Zijian Zhang, Xuetao Wei, and Yuxuan Liang. Controltraj: Controllable trajectory generation with topology-constrained diffusion model. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4676–4687, 2024.

A Broader Impact Statement

Our proposed MobTA framework aims to alleviate the scarcity of real human mobility trajectory data by generating high-quality synthetic trajectories using publicly accessible bus timetables, particularly in regions where data access is limited or constrained by commercial barriers and privacy concerns. By enabling trajectory generation without accessing real user-level mobility data from the target city, this approach has the potential to support important applications such as urban planning, transportation management, and epidemic control, while effectively reducing privacy risks and legal or ethical concerns associated with the use of real mobility data. However, as MobTA builds upon large language models, it may inherit implicit biases present in the underlying models or pretraining data. Therefore, when applying the generated synthetic trajectories to high-stakes or policy-related decision-making processes, it is important to carefully assess their applicability and limitations, and to ensure fairness, transparency, and responsible use of the technology.

B Additional Limitations

The controlled route-removal experiments show that generation quality deteriorates as bus coverage weakens, but they do not establish performance in a real car-dominant city. PCov and RCov use held-out target-city mobility trajectories and therefore serve as retrospective diagnostics rather than deployment-time selection criteria. Our evaluation covers only three Asian cities, so generalization to other transport systems remains untested. Despite manual validation, geocoding errors in bus-stop locations may affect bus–mobility alignment. Grid-level speed checks can flag unusually fast transitions but cannot establish road-network feasibility or other aspects of individual-trip plausibility. Finally, we have not evaluated whether the generated trajectories improve any downstream application.

C Detailed related work

C.1 Trajectory Generation

Human mobility trajectory data encapsulate rich spatiotemporal semantic information and serve as a fundamental data source for understanding urban dynamics and human interactions Chen et al. [2024], Li et al. [2025a], Zhang et al. [2026b,a], Cao et al. [2026], Liu et al. [2026]. Such data play a critical role across a wide range of smart city applications Liu et al. [2024], Ju et al. [2025]. In urban planning, analyses of large-scale mobility trajectories enable the characterization of commuting patterns and travel demands, which supports decisions on infrastructure service arrangement, land-use planning, and establishment of many other urban-related policies Zhou et al. [2024b], Ke et al. [2026]. In mobility prediction, historical trajectory data constitute a key input for traffic flow modeling and are widely used for congestion forecasting, estimated time of arrival prediction, and personalized route recommendation Tang et al. [2024], Anda et al. [2024], Liu et al. [2025], Chen et al. [2025c]. Moreover, for epidemic control, individual mobility trajectories provide essential information for tracing transmission chains, quantifying contact risks, and evaluating the effectiveness of intervention measures, thereby playing a crucial role in mitigating disease spread Kim and Kwan [2021], Hu et al. [2021], Elarde et al. [2021].

Despite their substantial practical value, the acquisition and sharing of real-world mobility trajectory data are highly challenging Chen et al. [2025a]. On the one hand, trajectory data are typically collected by companies like ride-hailing platforms and telecommunication operators. Due to commercial restrictions, they are rarely open to the public. On the other hand, trajectory data inherently contain highly identifiable spatiotemporal behavior patterns. Releasing such data can result in severe privacy risks. These challenges jointly limit the accessibility of high-quality mobility trajectories.

Consequently, trajectory generation has emerged as an effective solution Kulkarni and Garbinato [2017], Ouyang et al. [2018]. Its core objective is to capture the underlying spatiotemporal patterns and human mobility behaviors from real mobility trajectory data, and to synthesize trajectories accordingly. These synthetic trajectories aim to preserve the statistical properties and behavioral characteristics of real mobility data while not corresponding to any real individual, thereby reducing commercial constraints and privacy risks, as well as enabling downstream urban tasks. Existing trajectory generation approaches can majorly be categorized into model-based methods and learning-based methods Chen et al. [2025a], Zhu et al. [2025].

C.1.1 Model-based Trajectory Generation

Model-based trajectory generation approaches are built on the assumption that human mobility behaviors can be characterized by predefined statistical patterns or behavioral rules. These methods typically employ explicit probabilistic models or handcrafted mechanisms to describe transitions between mobility states. Because of their interpretability and low data requirements, model-based approaches were widely studied in the early era. SLAW Lee et al. [2009] assumes that people prefer to visit spatially proximate locations. It models both travel distances and dwell times using power-law distributions, thereby reproducing the heavy-tailed statistics commonly observed in real trajectories. SWIM Mei and Stefa [2009] adopts a social behavior perspective and models mobility decisions as a weighted selection over locations, where the weights are influenced by both location popularity and distance to residents' home. SIMPS Borrel et al. [2008] incorporates sociological theories by distinguishing between social and isolated behaviors. It models interpersonal relationships through social graphs, leading to power-law distributions in contact duration. Song et al. [2010] combines exploration of new locations with preferential return to previously visited ones, so as to capture the dynamic balance between new place seeking and habitual behavior. Building upon this framework, Pappalardo et al. [2015] incorporates gravity models that jointly consider spatial distance and location importance, enabling fine-grained modeling of decision preferences.

In addition, several methods use hidden markov models. Mathew et al. [2012] and Bindschaedler and Shokri [2016] employ hidden Markov models to capture latent state transitions in mobility trajectories, enabling next location prediction and trajectory generation. Baratchi et al. [2014] introduces hierarchical hidden semi-Markov models, which enhance the expressiveness of temporal dynamics while preserving model interpretability.

C.1.2 Learning-based Trajectory Generation

Learning-based trajectory generation methods aim to model and synthesize complex human mobility patterns by directly learning underlying spatiotemporal patterns from real trajectory data. In contrast to model-based approaches, these methods do not rely on strong predefined assumptions, but instead leverage deep learning models to automatically capture high-dimensional correlations and nonlinear dependencies inherent in mobility trajectory data. As a result, learning-based methods generally achieve superior generation quality. Existing learning-based methods are primarily based on generative adversarial networks, variational autoencoders, diffusion models, and more recently, large language models.

Generative adversarial networks (GANs) are among the earliest deep generative models adopted for trajectory generation. In a typical GAN framework, a generator is trained to generate trajectories that resemble real mobility data, while a discriminator aims to distinguish between real and generated trajectories. The two components are jointly optimized through adversarial training. Feng et al. [2020] incorporates human mobility priors into the GAN framework by combining self-attention-based temporal modeling modules with region-level spatial networks. Cao and Li [2021] explicitly balances the realism of generated trajectories with their utility for downstream tasks during adversarial training. LSTM-TrajGAN Rao et al. [2020] integrates LSTMs into the GAN framework to preserve both temporal consistency and privacy. In scenarios with strict road network constraints, Wang et al. [2021] proposes a two-stage generation strategy, where global trajectory distributions are first learned at a coarse grid level and then refined into fine-grained paths under road network constraints. STAGE Cao et al. [2025b] incorporates Transformers into the GAN framework and adopts multi-task learning to better capture long-range spatiotemporal dependencies.

Variational autoencoders (VAEs) typically assume that observed trajectories are generated from underlying latent variables and learn mappings between trajectory observations and latent representations through encoder–decoder architectures. Ding et al. [2019] proposes a generative model for multi-vehicle trajectory generation. It employs a bidirectional GRU encoder and a multi-branch decoder to jointly model the motion states of interacting vehicles in complex traffic scenarios. Huang et al. [2019] combines VAEs with a Seq2Seq framework and introduces latent variable modeling at each time step, enabling continuous trajectory generation. STULIG Zhou et al. [2020] incorporates a Gaussian mixture prior to capture the multimodal nature of trajectory data, and adopts convolutional encoders and decoders to improve the representation of complex spatiotemporal patterns. Compared to GAN-based models, VAE-based approaches typically exhibit more stable training behavior and clearer probabilistic semantics, but often face a trade-off when generating fine-grained trajectories.

In recent years, diffusion models have been introduced to trajectory generation. They leverage iterative noise injection and denoising processes to synthesize high-quality trajectories. Compared to GAN-based and VAE-based approaches, diffusion models typically offer more stable training dynamics and demonstrate strong performance, making them an increasingly popular paradigm for trajectory generation. DiffTraj Zhu et al. [2023] applies diffusion probabilistic models to GPS trajectory generation and adopts a multi-scale feature fusion denoising network to preserve spatiotemporal structures. Chu et al. [2024] further incorporates road topology, enabling generated trajectories to satisfy road network constraints. ControlTraj Zhu et al. [2024] introduces explicit conditioning on road structures and trip attributes within the diffusion process for controllable trajectory generation. To further improve feasibility within road networks, Diff-RNTraj Wei et al. [2024] represents trajectories as road segments with relative proportions and introduces a spatial reachability loss to enforce network-level reachability. Song et al. [2024] incorporates user profile information during denoising, enabling personalized trajectory generation. SEED Rao et al. [2025] explores the integration of sequence modeling techniques with diffusion models to improve modeling efficiency while maintaining high generation quality.

With the rapid development of large language models (LLMs), trajectory generation approaches built on LLMs have attracted increasing attention. The core challenge of these methods lies in tokenizing the spatial and temporal attributes of trajectories into sequential representations that can be processed by large language models, and leveraging the strong sequence modeling and contextual reasoning capabilities of LLMs to generate trajectories. Kobayashi et al. [2023] uses a hierarchical spatial encoding scheme that maps location and temporal information into discrete tokens, and performs autoregressive trajectory generation using GPT-2. Wang et al. [2024a] treats LLMs as decision-making agents, generating personalized mobility trajectories through activity pattern recognition and travel motivation reasoning. Shao et al. [2024] exploits the reasoning abilities of LLMs by formulating trajectory generation as a context-aware commonsense reasoning process.

Despite the substantial progress achieved by all these trajectory generation methods, most of these approaches rely on a common implicit assumption that at least a subset of real mobility trajectories from the target city is accessible during model training or model construction. In many real-world scenarios, however, this assumption is difficult to satisfy. Besides commercial barriers and privacy concerns, the high costs associated with sensor deployment, as well as the long-term burden of data management, prevent many cities from collecting and maintaining large-scale, high-resolution mobility trajectory datasets. As a result, the applicability of these existing trajectory generation methods is substantially limited in such data-inaccessible target cities.

C.2 Task Arithmetic

Task arithmetic has recently been a powerful approach for efficient model editing and generalization. It is based on task vectors, which are defined as the difference between parameters of a fine-tuned model and parameters of its corresponding pre-trained model. Task vectors serve as compact representations of specific tasks in parameter space, enabling direct and efficient modifications to a model's behavior without full retraining Ilharco et al. [2022], Zhang et al. [2023]. Their effectiveness has been supported by several works. Zhou et al. [2024a] conducts a detailed analysis of loss differences under vector operations. They derive upper bounds for both individual and average task loss differences, and further provide a closed-form solution for the optimal scaling coefficient that minimizes the average loss discrepancy between composed tasks. Li et al. [2025b] provides theoretical validation of task arithmetic in nonlinear Transformer architectures. They prove that, when tasks are independent, vector addition enables successful multi-task generalization. They also show that subtractive composition allows for precise forgetting in the presence of conflicting tasks. Moreover, this paper proves that, by linearly combining vectors from a set of unrelated tasks with appropriate coefficients, a model that generalizes to entirely new tasks can be constructed.

Beyond theoretical works, task arithmetic has proven to be a practical tool for both injecting new capabilities and eliminating harmful behaviors. Huang et al. [2024] demonstrates how instruction-following capabilities learned via fine-tuning can be transferred to a base model through vector addition. TIES-Merging Yadav et al. [2023] resolves parameter sign conflicts and pruning insignificant updates, enabling multi-task abilities to be merged without performance loss. In behavior removal and safety alignment, task arithmetic offers a lightweight yet effective solution. Bhardwaj et al. [2024] shows that safety-related behavior lost during fine-tuning can be restored by directly adding a "safety

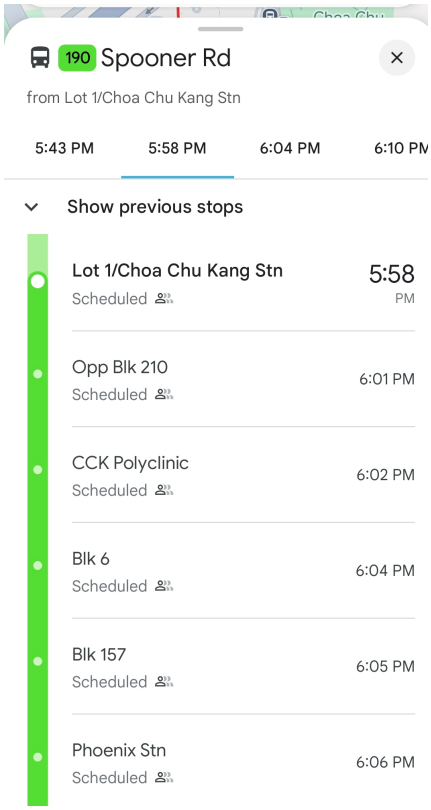


Figure 3: Screenshot of bus timetable in Google Map APP.

vector." Ethos Gao et al. [2024] introduces an orthogonal subspace projection method, decomposing task vectors into general-purpose and harmful components. By removing projection along harmful directions, Ethos enables targeted suppression of toxicity and bias, achieving precise and interpretable model correction.

Task arithmetic has also demonstrated strong potential in addressing out-of-distribution tasks and enabling cross-domain generalization, especially in low-resource or zero-shot settings. Parović et al. [2024] decouples language vectors and task vectors, and recombines them to support zero-shot transfer across languages. Chronopoulou et al. [2024] leverages compositional task vectors to enable summarization in languages with no labeled data. In domain-specific settings, task arithmetic helps bridge gaps between synthetic and real-world data. In information retrieval, domain-specific vectors significantly enhance model performance in unseen fields such as medicine and science Braga et al. [2025]. In automatic speech recognition, task vector differences between synthetic and real data guide the model toward realistic behavior distributions, improving performance in real-world environments Su et al. [2024].

D Screenshot of Bus Timetable Provided by Different Map Apps

Fig. 3 and 4 show the screenshot of bus timetable provided by Google Map and Naver Map. The former is widely used in numerous countries and the latter is used in Korea.

E Detailed Experimental Details of Empirical Observations on Bus Timetables and Mobility Trajectories

E.1 Detailed Experimental Setup for Spatial Stability of Bus Station Sequences

Here we provide a detailed explanation of the experiment which analyze the spatial consistency of bus station sequences in Shanghai between 2015 and 2025.

E.1.1 Data and Preprocessing

The bus station sequence data in 2015 is collected from a public historical dataset [3], which retains at most the first 30 bus stops per route. In contrast, the 2025 bus station sequence data is from its corresponding bus timetable data, which is crawled from the Moovit platform [4]. For consistency, the 2025 bus station sequence data is also truncated to the first 30 bus stops per route, ensuring that route comparisons across years are performed on sequences of equal maximum length. To reduce noise caused by inconsistencies in bus station naming, we apply lightweight normalization to all bus station names, including the removal of whitespace and any parenthetical annotations. These normalized bus station names are used for all similarity computations in this subsection. To enable alignment across years, we group bus routes by their route IDs, with routes sharing the same ID treated as belonging to the same group. Since the same route ID may correspond to different travel directions or terminal stations across years, a one-to-one correspondence between 2015 and 2025 bus lines does not always exist. Within each route group, we enumerate all feasible route pairings and select the pairing with the highest station-sequence similarity. Across the ten-year span from 2015 to 2025, approximately 88% of bus routes in the 2025 dataset can be successfully matched to a corresponding route group from 2015. Similarity results are then computed based on all successfully matched route pairs, reflecting the spatial consistency of the majority of bus routes that persist across the ten-year span. After route matching, we separately compute summary statistics for (a) routes within the entire city, and (b) routes within Outer Ring Road, which is our study region for Shanghai.

E.1.2 Similarity Metrics for Evaluating Spatial Stability of Bus Station Sequences

Given the full set of bus station sequences in 2015:

$$\begin{aligned} R^{\text{bus}}_{2015} &= \left\{ r^{\text{bus}}_{2015,1}, r^{\text{bus}}_{2015,2}, \ldots, r^{\text{bus}}_{2015,N_{2015}} \right\}, \\ r^{\text{bus}}_{2015,i} &= \left(s^{2015}_{i,1}, s^{2015}_{i,2}, \ldots, s^{2015}_{i,n_i} \right) \end{aligned} \tag{11}$$

and the full set of bus station sequences in 2025:

$$\begin{aligned} R^{\text{bus}}_{2025} &= \left\{ r^{\text{bus}}_{2025,1}, r^{\text{bus}}_{2025,2}, \ldots, r^{\text{bus}}_{2025,N_{2025}} \right\}, \\ r^{\text{bus}}_{2025,j} &= \left(s^{2025}_{j,1}, s^{2025}_{j,2}, \ldots, s^{2025}_{j,m_j} \right). \end{aligned} \tag{12}$$

we compute similarity metrics for each matched pair $\left(r^{\text{bus}}_{2015,i}, r^{\text{bus}}_{2025,j}\right)$. Final statistics are aggregated over all such matched pairs' results.

Longest Common Subsequence (LCS) measures the maximum length of a subsequence that is common to two ordered bus station sequences while preserving relative order. Formally, let $||LCS(x, y)||$ denote the length of the longest common subsequence between sequences x and y. The normalized LCS similarity between a matched pair of bus station sequences $r^{bus}_{2015,i}$ and $r^{bus}_{2025,j}$ is defined as Its normalized form is:

$$\text{LCS}_{\text{ratio}}(i,j) = \frac{||\text{LCS}(r^{\text{bus}}_{2015,i}, r^{\text{bus}}_{2025,j})||}{\min(n_i, m_j)}, \tag{13}$$

where n_i and m_j are the number of bus stops in the respective bus station sequence. This metric is sensitive to order but robust to insertions and deletions.

Jaccard Similarity (Jac) measures the overlap of bus stops between two bus station sequences. Let $S(r) = \{s \mid s \in r\}$ be the set of bus stops of bus station sequence r, we have the following definition

[3] https://msittig.wubi.org/bus/

[4] https://moovitapp.com/

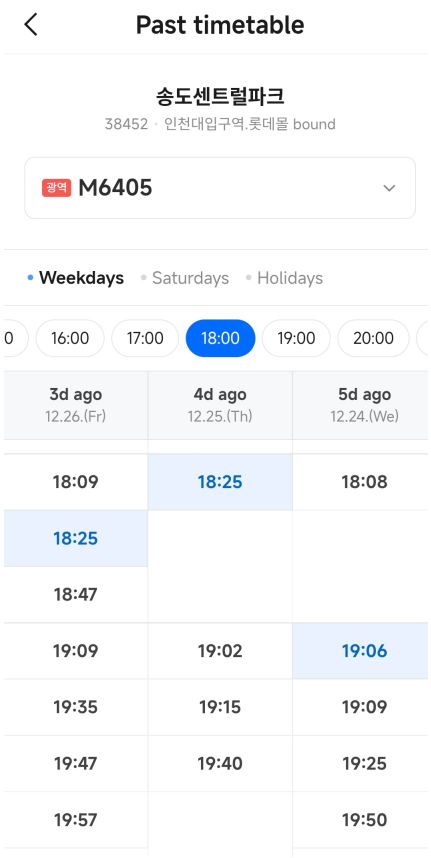


Figure 4: Screenshot of bus timetable in Naver Map APP.

of Jaccard Similarity:

$$\text{Jaccard}(i,j) = \frac{\left|S\left(r^{\text{bus}}_{2015,i}\right) \cap S\left(r^{\text{bus}}_{2025,j}\right)\right|}{\left|S\left(r^{\text{bus}}_{2015,i}\right) \cup S\left(r^{\text{bus}}_{2025,j}\right)\right|} \tag{14}$$

Dice Similarity is another set-based metric, which is more sensitive to the size of the intersection. It is defined as

$$\text{Dice}(i,j) = \frac{2\left|S(r^{\text{bus}}_{2015,i}) \cap S(r^{\text{bus}}_{2025,j})\right|}{\left|S(r^{\text{bus}}_{2015,i})\right| + \left|S(r^{\text{bus}}_{2025,j})\right|}. \tag{15}$$

Edit Distance on Real sequences (EDR) quantifies the minimum number of edit operations, including insertions, deletions, or substitutions, required to transform one bus station sequence into another. Smaller values indicate greater structural similarity.

Dynamic Time Warping (DTW) aligns sequences of unequal length by minimizing cumulative matching cost. In our experiment, the matching cost is defined based on whether the name of bus stops are the same. Lower DTW values indicate closer alignment in bus station sequences.

E.1.3 Detailed Analysis and Discussion

Table 1 quantitatively reports the spatial similarity between Shanghai's bus station sequences in 2015 and 2025. All similarity metrics consistently yield high scores, indicating that the spatial structure of bus routes remains largely stable over a ten-year time span.

For order-sensitive metrics, LCS reaches 0.8109 at the citywide level, implying that more than 80% of bus stops can be aligned while preserving relative order for most bus routes. This suggests that although some bus routes may undergo stop adjustments or branch modifications, their overall spatial structure remains stable. Consistently, EDR and DTW values are 0.2522 and 0.1355, respectively, indicating relatively low edit and alignment costs when transforming 2015 bus sequences into their

2025 counterparts. From a set-level perspective, Jaccard and Dice similarities achieve 0.6975 and 0.7937, respectively, demonstrating substantial overlap in bus stop coverage between the two set of bus station sequences. The higher Dice score, which emphasizes intersection size, further indicates that the majority of bus stops are retained over time.

When such analysis is restricted to area within the Outer Ring Road, which is the central area of shanghai, all metrics show further improvement. LCS increases to 0.8546, indicating greater order stability in the urban area. Jaccard and Dice similarities rise to 0.7459 and 0.8383, respectively, while EDR and DTW decrease to 0.2011 and 0.1062. These results suggest that structural adjustments to bus routes are even more limited in the urban area, and spatial stability is more guaranteed there.

E.2 Detailed Experimental Setup for Spatiotemporal Correlations Between Bus Timetables and Mobility Trajectories

To analyze the spatiotemporal correlations between bus timetables and general mobility trajectories, we partition the study area of each city into regular grids with $1km \times 1km$ resolution. Subsequently, all bus station locations and mobility trajectory points are mapped to their corresponding grid cells.

E.2.1 Spatial Distribution Consistency Metrics

Let c_i^{bus} and c_i^{mob} be the counts of bus stops from bus timetables and mobility trajectory points that fall into grid cell i, respectively. We first define the normalized spatial distributions as

$$P_i^{\text{bus}} = \frac{c_i^{\text{bus}}}{\sum_j c_j^{\text{bus}}}, \qquad P_i^{\text{mob}} = \frac{c_i^{\text{mob}}}{\sum_j c_j^{\text{mob}}}. \tag{16}$$

To systematically evaluate the spatial alignment between bus timetables and mobility trajectories, we adopt multiple complementary metrics, including Jensen–Shannon Divergence (JSD_c), Cosine Similarity (Cos), and Spatial Pearson Correlation (Pear-S). Jensen–Shannon Divergence (JSD) is used to quantify the overall discrepancy between the two distributions:

$$\text{JSD}_c\big(P^{\text{bus}} \,\|\, P^{\text{mob}}\big) = \frac{1}{2}\text{KL}\big(P^{\text{bus}} \,\|\, M\big) + \frac{1}{2}\text{KL}\big(P^{\text{mob}} \,\|\, M\big)\,, \tag{17}$$

where

$$M = \frac{1}{2}\left(P^{\text{bus}} + P^{\text{mob}}\right). \tag{18}$$

Cosine Similarity (Cos) is used to evaluate the alignment of the two spatial distributions in terms of their direction:

$$\text{Cos}(p^{bus}, p^{mob}) = \frac{p^{bus} \cdot p^{mob}}{\|p^{bus}\| \, \|p^{mob}\|} \tag{19}$$

.

Spatial Pearson Correlation (Pear-S) is computed to measure the linear correlation between the two spatial distributions.

$$\text{Pear-S}(P^{bus}, P^{Mob}) = \frac{\sum_i \left(P_i^{\text{bus}} - \bar{P}^{\text{bus}}\right)\left(P_i^{\text{mob}} - \bar{P}^{\text{mob}}\right)}{\sqrt{\sum_i \left(P_i^{\text{bus}} - \bar{P}^{\text{bus}}\right)^2}\sqrt{\sum_i \left(P_i^{\text{mob}} - \bar{P}^{\text{mob}}\right)^2}} \tag{20}$$

E.2.2 Spatial Coverage Metrics

To further analyze the spatial coverage capacity of bus timetables, we construct urban travel corridors based on bus timetables. Specifically, for a bus timetable $\tau^{\text{bus}} = \{(s_1, \tilde{t}_1), (s_2, \tilde{t}_2), \ldots, (s_M, \tilde{t}_M)\}$, we represent its spatial trajectory as a polyline, denoted as

$$\ell^{\text{bus}} = \text{LineString}(s_1, s_2, \ldots, s_m). \tag{21}$$

We apply a buffer with fixed radius r to each bus polyline ℓ_k^{bus} to obtain a bus corridor region $\text{Buffer}\big(\ell_k^{\text{bus}}, r\big)$. In all experiments, we set the buffer radius to $r = 100m$, which reflects a typical pedestrian access distance to bus routes in urban environments Daniels and Mulley [2013], Tetali et al. [2025]. Aggregating all such buffered corridors yields the city-scale bus corridor region:

$$\mathcal{C}^{\text{bus}} = \bigcup_k \text{Buffer}\big(\ell_k^{\text{bus}}, r\big)\,. \tag{22}$$

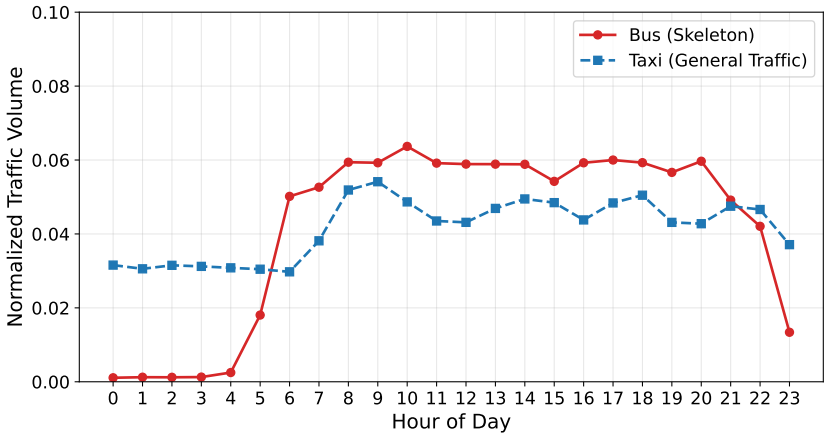


Figure 5: Hourly temporal distribution of bus timetables and mobility trajectories in Shanghai.

Point Coverage is used to quantify the proportion of mobility trajectory points that fall within bus corridors. Let $\{p_i\}$ denote all points from mobility trajectories. The point coverage is defined as

$$\mathrm{PCov} = \frac{1}{|\{p_j\}|} \sum_j \mathbf{1}\big(p_j \in \mathcal{C}^{\mathrm{bus}}\big)\,, \tag{23}$$

where $\mathbf{1}(\cdot)$ is the indicator function. Point Coverage reflects the extent to which mobility trajectories spatially overlap with bus transit structures at the point level. To further capture path-level structural overlap, we define Route Coverage. Given a mobility trajectory represented as a polyline ℓ^{mob}, its route coverage is defined as

$$\mathrm{RCov}_{\mathrm{traj}}(\ell^{\mathrm{mob}}) = \frac{\mathrm{Length}\big(\ell^{\mathrm{mob}} \cap \mathcal{C}^{\mathrm{bus}}\big)}{\mathrm{Length}(\ell^{\mathrm{mob}})}. \tag{24}$$

Averaging over all mobility trajectories yields the overall route coverage:

$$\mathrm{RCov} = \mathbb{E}_{\ell^{\mathrm{mob}}}\left[\mathrm{RCov}_{\mathrm{traj}}(\ell^{\mathrm{mob}})\right]. \tag{25}$$

Point coverage reflects how frequently mobility trajectories are near bus corridors, while route coverage characterizes the extent to which mobility trajectories extend along bus station sequences. Together, these metrics provide complementary perspectives on the spatial representational capacity of bus transit trajectories for general urban mobility.

E.2.3 Temporal Correlation Metrics

Beyond spatial correlations, we further examine the temporal correlation between bus timetables and mobility trajectories. Each day is divided into 24 hourly intervals, and the frequencies of bus timetables and mobility trajectories are computed for each interval. The normalized temporal distributions are denoted as

$$\mathbf{T}^{\mathrm{bus}} = (T_1^{\mathrm{bus}}, \ldots, T_{24}^{\mathrm{bus}}), \qquad \mathbf{T}^{\mathrm{mob}} = (T_1^{\mathrm{mob}}, \ldots, T_{24}^{\mathrm{mob}}), \tag{26}$$

where

$$\sum_h T_h^{\mathrm{bus}} = \sum_h T_h^{\mathrm{mob}} = 1. \tag{27}$$

Temporal correlation is computed using the temporal Pearson correlation coefficient:

$$\text{Pear-T}(\mathbf{T}^{\mathrm{bus}}, \mathbf{T}^{\mathrm{mob}}) = \frac{\sum_h \left(T_h^{\mathrm{bus}} - \bar{\mathbf{T}}^{\mathrm{bus}}\right)\left(T_h^{\mathrm{mob}} - \bar{\mathbf{T}}^{\mathrm{mob}}\right)}{\sqrt{\sum_h \left(T_h^{\mathrm{bus}} - \bar{\mathbf{T}}^{\mathrm{bus}}\right)^2}\sqrt{\sum_h \left(T_h^{\mathrm{mob}} - \bar{\mathbf{T}}^{\mathrm{mob}}\right)^2}} \tag{28}$$

where $\bar{\mathbf{T}}^{\mathrm{bus}}$ and $\bar{\mathbf{T}}^{\mathrm{mob}}$ denote the mean values of the corresponding temporal distributions.

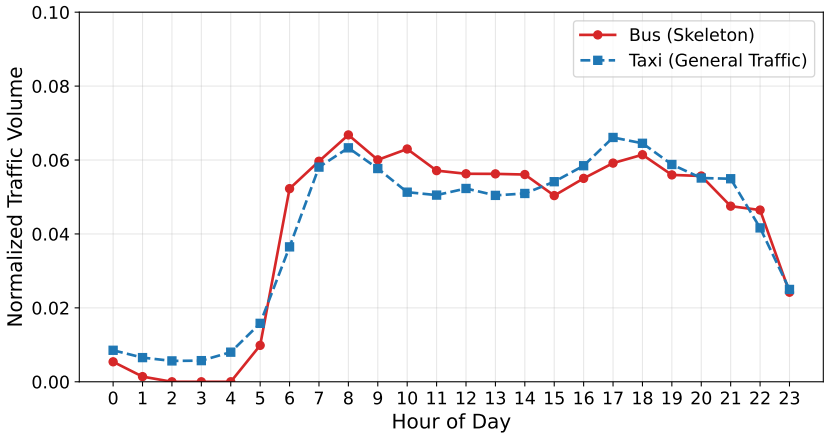


Figure 6: Hourly temporal distribution of bus timetables and mobility trajectories in Singapore.

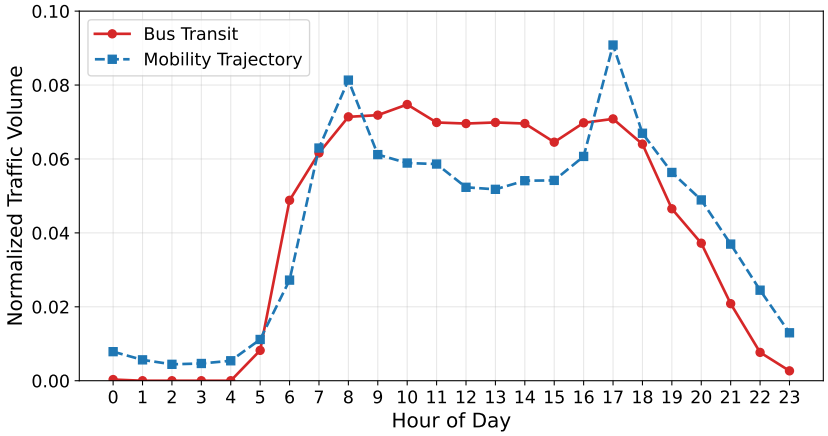


Figure 7: Hourly temporal distribution of bus timetables and mobility trajectories in Wuxi.

F Spatial Discretization

To ensure structural comparability across different cities, we employ a unified spiral indexing strategy for spatial discretization. Figure 8 provides visualizations of this strategy applied to Shanghai, Wuxi, and Singapore. In each representation, the study area is partitioned into regular grids with the same spatial resolution ($1km \times 1km$). The grid located at the geometric center of the study area, which is computed from the boundary polygon of urban center area, is designated as the anchor location and assigned the index 0 (corresponding to the token `<LOC_0>`). Subsequent indices are assigned to surrounding grids in an outward, clockwise spiral manner. Formally, grid labeled with index i corresponds to the spatial token `<LOC_i>`. Grids that fall within the predefined city boundary are highlighted in green, while those outside are shown in white. This consistent indexing scheme allows relative spatial positions (e.g., center vs. periphery) to be encoded similarly across different cities, facilitating unified trajectory representation.

G First-order Approximation for Projected Task Vector Difference in task-relevant Parameter Subspace

This section provides a detailed derivation of the key claims made in Sec. 4.4.3. Specifically, we show that for a trajectory dataset D, the projection of the task vector difference between base and

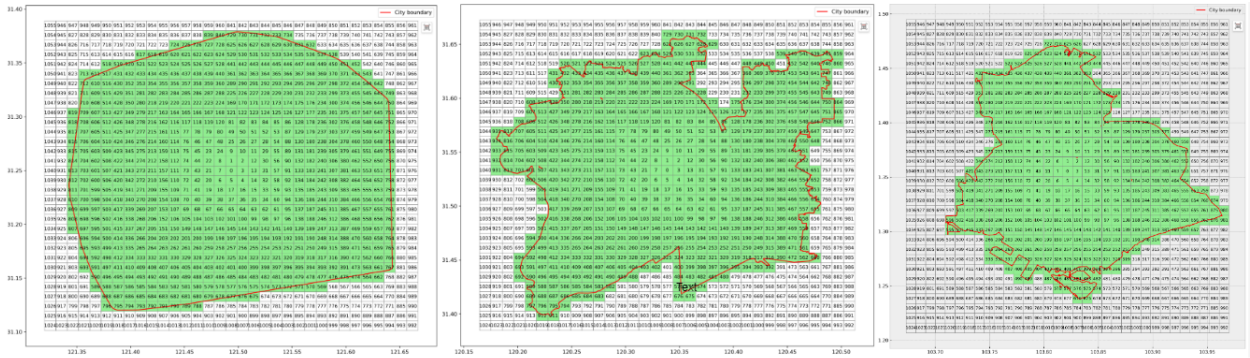


Figure 8: Visualization of spatial discretization and the spiral indexing strategy across different cities. Grid labeled with index i corresponds to the spatial token <LOC_i>.

instruction-tuned LLMs onto the task-relevant parameter subspace $\mathcal{T}_{\text{traj}}$ satisfies

$$\begin{aligned}\mathbf{P}_{\text{traj}}\big(\mathbf{V}_B(\mathcal{D}) - \mathbf{V}_I(\mathcal{D})\big) \approx \\ -\eta\,\mathbf{P}_{\text{traj}}\big(\nabla_\theta\mathcal{L}(\theta_0^{(B)};\mathcal{D}) - \nabla_\theta\mathcal{L}(\theta_0^{(I)};\mathcal{D})\big)\end{aligned} \tag{29}$$

We further demonstrate that, under a first-order approximation, this projected difference is expected to remain numerically small.

G.1 First-Order Approximation of Task Vectors

Let θ denote the model parameters, and let $\mathcal{L}(\theta;\mathcal{D})$ denote the trajectory generation training loss on trajectory dataset $\mathcal{D}$. Let θ_0 denote the initialization parameters for fine-tuning. A parameter-efficient fine-tuning procedure involving T gradient descent steps produces a parameter sequence $\{\theta_t\}_{t=0}^{T}$ via the update rule:

$$\theta_{t+1} = \theta_t - \eta_t\nabla_\theta\mathcal{L}(\theta_t;\mathcal{D}), \quad t = 0,\ldots,T-1, \tag{30}$$

where η_t is the learning rate at step t. The total parameter shift, defined as the task vector $\mathbf{V}(\theta_0,\mathcal{D})$, is therefore given by:

$$\mathbf{V}(\theta_0,\mathcal{D}) := \theta_T - \theta_0 = -\sum_{t=0}^{T-1}\eta_t\nabla_\theta\mathcal{L}(\theta_t;\mathcal{D}), \tag{31}$$

Under a small-step and parameter-efficient fine-tuning setting, parameter updates are small, which suppresses higher-order terms and makes a first-order approximation a reasonable abstraction. Thus, the optimization dynamics can be analyzed via a first-order Taylor expansion around the initialization θ_0. This yields the standard linearized task vector approximation:

$$\mathbf{V}(\theta_0,\mathcal{D}) \approx -\eta\nabla_\theta L(\theta_0;D). \tag{32}$$

where η represents the effective step size. Applying this approximation to the base and instruction-tuned LLMs' initialization parameters $\theta_0^{(B)}$ and $\theta_0^{(I)}$, respectively, we will get:

$$\mathbf{V}_B(\mathcal{D}) - \mathbf{V}_I(\mathcal{D}) \approx -\eta\big(\nabla_\theta\mathcal{L}(\theta_0^{(B)};\mathcal{D}) - \nabla_\theta\mathcal{L}(\theta_0^{(I)};\mathcal{D})\big). \tag{33}$$

Since the projection operator $\mathbf{P}_{\text{traj}}$ is linear, applying it to both sides gives Eq.28.

G.2 First-Order Weak Coupling of the Projected Gradient Shift

We introduce the projected gradient function

$$g_T(\theta;\mathcal{D}) := \mathbf{P}_{\text{traj}}\nabla_\theta\mathcal{L}(\theta;\mathcal{D}), \tag{34}$$

where $\mathbf{P}_{\text{traj}}$ denotes the orthogonal projection onto the task-relevant parameter subspace $\mathcal{T}_{\text{traj}}$. The projection operator $\mathbf{P}_{\text{traj}}$ is defined with respect to the task-relevant subspace $\mathcal{T}_{\text{traj}}$ induced at the reference initialization $\theta_0^{(B)}$ and is treated as a constant linear operator independent of θ. Under this convention, $\mathbf{P}_{\text{traj}}$ is a fixed linear operator, and it follows that

$$\nabla_\theta g_T(\theta; \mathcal{D}) = \mathbf{P}_{\text{traj}} \nabla_\theta^2 \mathcal{L}(\theta; \mathcal{D}) = \mathbf{P}_{\text{traj}} H(\theta; \mathcal{D}), \tag{35}$$

where $H(\theta; \mathcal{D})$ denotes the Hessian matrix of trajectory generation loss $\mathcal{L}(\theta; \mathcal{D})$ with respect to the parameters θ.

We perform a first-order Taylor expansion of $g_T(\theta; \mathcal{D})$ around the reference initialization $\theta_0^{(B)}$ and substitute $\theta_0^{(I)} = \theta_0^{(B)} + \delta_0$, obtaining

$$g_T(\theta_0^{(I)}; \mathcal{D}) = g_T(\theta_0^{(B)}; \mathcal{D}) + \mathbf{P}_{\text{traj}} H(\theta_0^{(B)}; \mathcal{D})\, \delta_0 + O(\|\delta_0\|^2). \tag{36}$$

Rearranging terms gives

$$g_T(\theta_0^{(B)}; \mathcal{D}) - g_T(\theta_0^{(I)}; \mathcal{D}) = -\mathbf{P}_{\text{traj}} H(\theta_0^{(B)}; \mathcal{D})\, \delta_0 + O(\|\delta_0\|^2). \tag{37}$$

Let $\mathbf{Q}_{\text{traj}} := \mathbf{I} - \mathbf{P}_{\text{traj}}$. Any vector, including δ_0, admits the decomposition

$$\delta_0 = \mathbf{P}_{\text{traj}} \delta_0 + \mathbf{Q}_{\text{traj}} \delta_0, \quad \mathbf{P}_{\text{traj}} \delta_0 \in \mathcal{T}_{\text{traj}},\ \mathbf{Q}_{\text{traj}} \delta_0 \in \mathcal{T}_{\text{traj}}^{\perp}. \tag{38}$$

Substituting this decomposition into $\mathbf{P}_{\text{traj}} H(\theta_0^{(B)}; \mathcal{D})\, \delta_0$ yields

$$\begin{aligned} \mathbf{P}_{\text{traj}} H(\theta_0^{(B)}; \mathcal{D}) \delta_0 = {} & \mathbf{P}_{\text{traj}} H(\theta_0^{(B)}; \mathcal{D}) \mathbf{P}_{\text{traj}} \delta_0 \\ & + \mathbf{P}_{\text{traj}} H(\theta_0^{(B)}; \mathcal{D}) \mathbf{Q}_{\text{traj}} \delta_0. \end{aligned} \tag{39}$$

The first term on the right-hand side is governed by the projection of the initialization difference δ_0 onto the task-relevant parameter subspace $\mathcal{T}_{\text{traj}}$. δ_0 is primarily associated with instruction-following and safety alignment rather than trajectory-related tokens, and it lies predominantly in the orthogonal complement $\mathcal{T}_{\text{traj}}^{\perp}$, rendering the projection $\|\mathbf{P}_{\text{traj}} \delta_0\|$ small even when $||\delta_0||$ itself might not. The second term on the right-hand side captures the interaction between $\mathcal{T}_{\text{traj}}$ and its orthogonal complement $\mathcal{T}_{\text{traj}}^{\perp}$. Under the setting of MobTA, given that the trajectory generation dynamics are structurally disentangled from specific instruction-following and safety alignment tasks, parameter perturbations in $\mathcal{T}_{\text{traj}}^{\perp}$ induce negligible gradient changes along trajectory-relevant directions, making the second term on the right-hand side small as well. Consequently, the first-order contribution of $\mathbf{P}_{\text{traj}} H(\theta_0^{(B)}; \mathcal{D})\, \delta_0$ is suppressed and the dominant contribution to $||g_T(\theta_0^{(B)}; \mathcal{D}) - g_T(\theta_0^{(I)}; \mathcal{D})||$ arises from higher-order terms of order $O(\|\delta_0\|^2)$, which are themselves minimal under a small-step and parameter-efficient fine-tuning setting Jacot et al. [2018], Hu et al. [2022], Malladi et al. [2023], making $||\mathbf{P}_{\text{traj}} H(\theta_0^{(B)}; \mathcal{D})\, \delta_0||$ numerically negligible. This leads to $||g_T(\theta_0^{(B)}; \mathcal{D}) - g_T(\theta_0^{(I)}; \mathcal{D})||$, and further $||\mathbf{P}_{\text{traj}} \big(\mathbf{V}_B(\mathcal{D}) - \mathbf{V}_I(\mathcal{D})\big)||$, numerically negligible.

G.3 Empirical Validation of the Projection Assumption

To empirically examine the projection assumption in Eq. 6, we compute the initialization parameter difference δ_0 between LLaMA-3.2-3B base and instruct models, and its projection onto the trajectory-relevant parameter subspace $\mathcal{T}_{\text{traj}}$. The results show that $\|\delta_0\| = 106.24$, while $\|\mathbf{P}_{\text{traj}} \delta_0\| = 5.31$, yielding a ratio of 0.049. This indicates that although instruction tuning introduces a non-negligible global parameter change, only a small fraction of this change lies in the trajectory-relevant subspace, empirically supporting the core assumption in Eq. 6. This validation primarily supports the stability analysis of MobTA across base and instruction-tuned LLMs.

H Data and Data Processing

H.1 City Selection and Source-City Sensitivity

We evaluate Shanghai (SH), Wuxi (WX), and Singapore (SG) because both bus timetables and independently collected mobility trajectories are available for these cities, allowing us to construct

Station	Departure #1	Departure #2	Departure #3	Departure #4
Jurong East Ave 1 - Jurong Polyclinic (28499)	5:47	5:54	5:58	6:00
Jurong East Ave 1 - Blk 316 (28469)	5:48	5:55	5:59	6:01
Jurong East Ave 1 - Opp Parc Oasis (28459)	5:50	5:57	6:01	6:03
Jurong Town Hall Rd - Blk 228 (28289)	5:51	5:58	6:02	6:04
Boon Lay Way - Bef Yuhua Cc (28089)	5:52	5:59	6:03	6:05
Boon Lay Way - Blk 210 (28079)	5:53	6:00	6:04	6:06
Boon Lay Way - Blk 203 (28069)	5:55	6:02	6:06	6:08
Boon Lay Way - Ng Teng Fong General Hosp (28059)	5:57	6:04	6:08	6:10
Boon Lay Way - Opp the Synergy (28049)	5:58	6:05	6:09	6:11
Boon Lay Way - Aft Boon Lay Sub Stn (28039)	6:00	6:07	6:11	6:13
Boon Lay Way - 8 @ Tradehub 21 (28029)	6:01	6:08	6:12	6:14
C'Wealth Ave West - Aft Toh Tuck Ave (28019)	6:03	6:10	6:14	6:16
C'Wealth Ave West - Blk 203 (20109)	6:05	6:12	6:16	6:18
C'Wealth Ave West - Blk 317 (17189)	6:07	6:14	6:18	6:20
Clementi Ave 3 - Clementi Int (17009)	6:08	6:15	6:19	6:21

Figure 9: Example of one bus transit timetable in Singapore. Bus transit timetables in Shanghai and Wuxi are in the same format.

the transfer tasks and evaluate generated trajectories against held-out real data. Their bus–mobility alignment differs across the spatial and temporal measures in Table 2, as do their distributional discrepancies in Table 3. Nevertheless, this evaluation is geographically limited and does not establish performance in cities with substantially different transport systems. The mobility datasets for SH and SG consist of taxi GPS trajectories, whereas the WX mobility dataset consists of private-vehicle GPS trajectories. None of these mobility datasets contains bus trajectories; thus, the measured bus–mobility alignment does not arise from including bus movements in the mobility data. For each transfer pair, target-city mobility trajectories are used for empirical analysis and final evaluation, not for training the three task-specific checkpoints or constructing the target-city generation model.

Table 4 evaluates all six directed source–target city pairs. Source-city choice affects the resulting JSD values, while MobTA outperforms COLA across the evaluated pairs. Stronger source-city bus–mobility alignment may be useful when choosing a source, but the evidence is mixed across metrics: for SH as the target, SG as the source yields lower distance and radius JSD than WX, whereas WX yields lower duration and location-count JSD. We therefore treat source-city alignment as a preliminary indicator rather than a validated selection criterion.

H.2 Bus Timetable Collection and Station Coordinate Extraction

We collect bus timetables from Moovit platform [5]. Specifically, for each city, we crawl all available bus route pages, parse the complete bus station sequences and scheduled arrival times for each route, and convert all extracted data into a unified "station × departure time" tabular format. This process yields standardized station-level bus timetable data. Example of one processed Singapore bus timetable is illustrated in Fig.9.

After obtaining bus timetables for each city, we further augment each bus stop with corresponding geographic coordinates. For Singapore, bus stop latitude and longitude information is obtained from the publicly released static dataset of the Land Transport Authority (LTA) [6]. We match the bus stop IDs extracted from Moovit with the official LTA records to assign geographic coordinates to each bus stop. For Shanghai and Wuxi, bus stop coordinates are queried via the Amap API. Since some bus stops are missing from public map services or exhibit obvious spatial deviations in queried coordinates, we conduct careful manual inspection and validation for bus stations in these two cities to ensure spatial accuracy. Moreover, as Amap adopts the GCJ-02 coordinate system, all coordinates are converted into the standard WGS-84 coordinate system to maintain consistency across data sources.

[5] https://moovitapp.com/

[6] https://datamall.lta.gov.sg/content/datamall/en/static-data.html

Table 6: Dataset statistics.

Dataset	Num of Records	Num of Trajs	Time Span
SH	18,462,492	1,007,133	2016-08-01 – 2016-08-03
WX	346,416	16,436	2020-07-18 – 2020-08-17
SG	3,600,936	187,988	2024-03-01 – 2024-03-31

H.3 Spatiotemporal Cropping and Filtering

For all three cities, our analysis focuses exclusively on their core urban areas. Both bus timetable data and mobility trajectory data are spatially cropped according to corresponding study boundaries, and only bus stations and trajectory records falling within these regions are retained. After spatial cropping, bus timetables and mobility trajectories with fewer than three records are discarded, as such short sequences lack meaningful spatiotemporal structure. Along the temporal dimension, we discretize each day into minute-level intervals, resulting in a total of $24 \times 60 = 1440$ time steps. The resulting time values are subsequently represented using existing numerical tokens, rather than newly introduced time-specific tokens. Since time-of-day is inherently ordered, this representation preserves relative temporal proximity and ordering information through the numerical values themselves. This unified temporal resolution ensures consistency across bus timetables and mobility trajectories in different cities.

H.4 Grid-Based Representation and Trajectory Length Control

To construct a unified spatial representation, all bus timetables and mobility trajectories are mapped to the regular $1km \times 1km$ grid cells in which they reside, thereby discretizing continuous geographic coordinates into grid-based location identifiers. Under this representation, both bus timetables and real mobility trajectories can be expressed as discrete sequences of time–location pairs. Considering the computational and modeling constraints of large language models when handling long sequences, we apply uniform trajectory length control. For both bus timetables and real mobility trajectories, if the trajectory length exceeds 50, it is split into two shorter sub-trajectories. This operation is applied only to excessively long samples to prevent individual trajectories from occupying an excessive number of sequence tokens. Summary statistics of mobility trajectory data for each city are reported in Table 6.

I Baseline Introduction and Implementation Details

Here we provide a brief introduction of baseline models we use, and implementation details of these baselines. To ensure a fair comparison, we select baselines under the same information constraints as our problem setting: during training, no real mobility trajectories from the target city are available; during generation, methods are conditioned only on a starting location and starting time sampled from target-city bus timetables. Under this criterion, we do not include trajectory forecasting or next-location prediction methods, such as NextLocMoE Liu et al. [2025],GTG Wang et al. [2025],TransferTraj Wei et al. [2025], and LLaMA-Mob-8B Tang et al. [2024]. Except for COLA, all baseline models are trained solely on target city bus timetables. For COLA, the target city model is trained using only target city bus timetables, while source-city data are varied across bus timetables, mobility trajectories, and bus timetables + mobility trajectories configurations. We report COLA's best result across these source-city data configurations. In addition, since no explicit road network information is considered, models that use road topology Zhu et al. [2024], Tao et al. [2024], Wei et al. [2024], Cao et al. [2025a], Wei et al. [2025] are excluded from comparison. For all methods, starting locations and times of generated trajectories are randomly sampled from the spatial and temporal distributions of target city bus timetables. All generated trajectories are evaluated on the target city mobility trajectory test sets.

- **TimeGeo** Jiang et al. [2016]: TimeGeo is a classical statistical trajectory generation method that models individual mobility by capturing periodic patterns and regularities in spatial stay and travel times.
- **MoveSim** Feng et al. [2020]: MoveSim simulates individual mobility behaviors by learning transition patterns and temporal distributions from real trajectory data.

- **TSG** Wang et al. [2021]: TSG is a two-stage trajectory generation framework. The first stage learns inter-region transition relationships, while the second stage refines trajectories at the path level using road network information. Since no road network data are used in our setting, we only adopt the first stage to generate trajectories.
- **TrajGDM** Chu et al. [2023]: TrajGDM is a diffusion-based trajectory generation method that learns the spatiotemporal distribution of real trajectories through a progressive denoising process.
- **DiffTraj** Zhu et al. [2023]: DiffTraj adopts a diffusion-based generative method for modeling human mobility trajectories.
- **TrajGEN** Cao and Li [2021]: TrajGEN focuses on learning the statistical characteristics of trajectories in terms of spatial distributions and temporal rhythms.
- **Traveller** Luo et al. [2025]: Traveller generates individual mobility trajectories via sequence modeling and preserves global structural consistency of trajectories.
- **Geollama**$^{+}$ Li et al. [2025c]: Geo-Llama leverages the strong semantic understanding and reasoning LLMs by reformulating trajectory generation as a sequence prediction problem. Unlike the original Geo-Llama design, we treat the ID of each location as a newly introduced spatial token and explicitly train its corresponding embedding and LM head. In addition, the up, down, and gate linear layers in the FFN module are jointly fine-tuned using LoRA. To distinguish it from the original Geo-Llama, we denote this variant with a superscript "+".
- **COLA** Wang et al. [2024b]: COLA is a transferable trajectory generation framework that explicitly models cross-city differences. In our setting, the target city is trained using only its bus timetables, while the source city are trained with multiple data combinations, including bus timetables, mobility trajectories, and bus timetables + mobility trajectories. We report the best-performing configuration, with detailed results for different combinations of WX→SH shown in Table 7.

Table 7: Performance comparison of COLA under different settings.

Method (Setting)	Dist.	Rad.	Dur.	Loc.
COLA (WX-Bus & WX-Mob , SH-Bus)	0.1911	0.1690	0.0316	0.1689
COLA (WX-Bus, SH-Bus)	0.1831	0.1564	0.0344	0.1544
COLA (WX-Mob, SH-Bus)	0.1843	0.1652	0.0335	0.1668

J Evaluation Metrics

We evaluate model performance by comparing generated trajectories with real mobility trajectories in terms of discrepancies across several statistical distributions. Following prior works, we compute the following statistical properties from both generated and real trajectories:

- Distance (Dist.) measures the cumulative geographic distance between consecutive visited locations within a single trajectory.
- Radius (Rad.) describes the spatial extent of a trajectory, reflecting the dispersion of an individual's activity area.
- Duration (Dur.) represents the distribution of consecutive dwell times at the same location.
- TrajLoc (Loc.) denotes the number of distinct locations visited within a single trajectory and serves as a measure of spatial diversity.

For each of the above statistical attributes, we adopt the Jensen–Shannon Divergence (JSD) to quantify the discrepancy between the generated distribution D' and the real distribution D. JSD is defined as

$$\mathrm{JSD}(D, D') = \frac{1}{2}\mathrm{KL}\left(D \,\middle\|\, \frac{D+D'}{2}\right) + \frac{1}{2}\mathrm{KL}\left(D' \,\middle\|\, \frac{D+D'}{2}\right), \tag{40}$$

where $\mathrm{KL}(\cdot)$ denotes the Kullback–Leibler divergence. As all these attributes are continuous-valued, we first estimate their probability distributions using histogram-based density estimation and then compute the corresponding JSD values. Smaller JSD values indicate that the generated trajectories more closely match the real trajectories, and therefore exhibit higher realism.

Table 8: Trajectory generation performance under bus-degradation settings. R50 randomly retains 50% of target-city bus routes, while S50 randomly retains 50% of stops within each bus route. Smaller JSD values indicate better performance.

Target	Method	R50				S50			
		Dist.	Rad.	Dur.	Loc.	Dist.	Rad.	Dur.	Loc.
SG	Traveller	0.286	0.304	0.049	0.241	0.253	0.272	0.046	0.223
SG	TrajGDM	0.291	0.331	0.047	0.227	0.257	0.301	0.042	0.198
SG	COLA (SH→SG)	0.281	0.239	0.051	0.191	0.238	0.207	0.046	0.151
SG	COLA (WX→SG)	0.322	0.274	0.080	0.252	0.276	0.239	0.074	0.211
SG	MobTA (SH→SG)	**0.146**	**0.119**	**0.043**	**0.131**	**0.108**	**0.064**	**0.040**	**0.102**
SG	MobTA (WX→SG)	0.183	0.137	0.046	0.165	0.133	0.084	0.043	0.138
WX	Traveller	0.336	0.218	0.114	0.176	0.299	0.192	0.107	0.153
WX	TrajGDM	0.324	0.229	0.094	0.121	0.283	0.203	0.087	0.103
WX	COLA (SH→WX)	0.228	0.182	0.060	0.159	0.191	0.151	0.055	0.126
WX	COLA (SG→WX)	0.258	0.214	0.067	0.165	0.221	0.179	0.061	0.132
WX	MobTA (SH→WX)	0.151	0.098	0.059	0.084	0.116	0.071	**0.049**	0.071
WX	MobTA (SG→WX)	**0.136**	**0.091**	**0.055**	**0.076**	**0.103**	**0.066**	0.053	**0.064**
SH	Traveller	0.255	0.212	0.070	0.129	0.232	0.191	0.065	0.111
SH	TrajGDM	0.266	0.228	0.059	0.136	0.241	0.205	0.054	0.118
SH	COLA (WX→SH)	0.233	0.199	0.041	0.221	0.198	0.171	0.037	0.182
SH	COLA (SG→SH)	0.206	0.189	0.049	0.192	0.176	0.163	0.045	0.159
SH	MobTA (WX→SH)	0.134	0.092	**0.027**	**0.072**	0.109	0.073	**0.023**	**0.058**
SH	MobTA (SG→SH)	**0.111**	**0.064**	0.037	0.081	**0.086**	**0.045**	0.033	0.063

K Bus Degradation Experiments

To evaluate MobTA's robustness when bus information becomes weaker, we conduct bus-degradation experiments over all source-target city pairs. We consider two degradation settings on the target-city bus timetables. In R50, we randomly retain 50% of bus routes, simulating reduced bus coverage. In S50, we randomly retain 50% of bus stops within each bus route, simulating increased structural sparsity of route information. These settings are not intended to fully represent all private-vehicle-dominant cities; instead, they provide controlled proxies for weakened bus-supported scenarios.

Table 8 reports the results. As bus information becomes weaker, all methods generally degrade, confirming that bus timetables provide important structural information for this problem. Nevertheless, MobTA consistently achieves the best performance across target cities, source cities, and degradation types. These results indicate that MobTA remains more robust than existing baselines under degraded bus information.

K.1 Sensitivity to Weak Bus Coverage

The R50 and S50 settings above distinguish route removal from stop sparsification. To examine progressively weaker bus signals, we focus on the R setting, which removes entire target-city routes and thereby reduces the extent of bus corridors. Removing stops along retained routes, as in S50, can leave much of the route-level corridor intact. Table 9 reports performance across seven route-retention levels for SH→SG and SG→SH, together with point coverage (PCov) and route coverage (RCov) as defined above.

MobTA's JSD values generally rise as fewer routes are retained, although they remain below COLA's at every reported level. For SH→SG, reducing retention from R100 to R25 lowers RCov from 63.82% to 43.97% and raises MobTA's distance JSD from 0.086 to 0.185. At R25, PCov is still 60.43%, illustrating that point-level coverage alone does not capture how much of a trajectory's path overlaps the bus corridors. In these controlled experiments, RCov around or below 50% is associated with severe degradation; it is an observed risk indicator, not a universal failure threshold.

Table 9: Sensitivity to target-city bus-route retention. PCov and RCov are percentages; lower JSD values are better. R100 retains all routes and R1 retains 1%.

City pair	Setting	PCov	RCov	COLA				MobTA			
		(%)	(%)	Dist.	Rad.	Dur.	Loc.	Dist.	Rad.	Dur.	Loc.
SH→SG	R100	80.08	63.82	0.217	0.184	0.044	0.119	0.086	0.047	0.039	0.096
	R75	77.12	60.79	0.235	0.199	0.046	0.139	0.101	0.062	0.040	0.105
	R50	71.86	55.73	0.281	0.239	0.051	0.191	0.146	0.119	0.043	0.131
	R25	60.43	43.97	0.350	0.310	0.059	0.281	0.185	0.165	0.047	0.153
	R10	41.78	28.88	0.424	0.388	0.068	0.373	0.212	0.193	0.054	0.162
	R5	24.63	16.59	0.462	0.427	0.075	0.430	0.245	0.225	0.058	0.188
	R1	2.87	1.73	0.505	0.475	0.084	0.492	0.323	0.300	0.065	0.272
SG→SH	R100	86.86	86.87	0.161	0.150	0.042	0.136	0.075	0.037	0.031	0.055
	R75	85.21	85.37	0.174	0.161	0.044	0.151	0.084	0.044	0.032	0.062
	R50	81.71	81.90	0.206	0.189	0.049	0.192	0.111	0.064	0.037	0.081
	R25	71.24	71.96	0.266	0.241	0.055	0.262	0.148	0.102	0.042	0.112
	R10	49.52	49.56	0.332	0.302	0.063	0.342	0.182	0.142	0.048	0.148
	R5	32.44	32.47	0.378	0.347	0.070	0.401	0.215	0.180	0.054	0.180
	R1	9.20	9.12	0.438	0.408	0.079	0.472	0.285	0.250	0.063	0.235

L Trajectory Generation Behavior under Different Spatial Data Coverage Conditions

We analyze MobTA's trajectory generation behavior under different spatial data coverage conditions. Taking the WX → SH transfer setting as a case study example, we categorize Shanghai locations based on whether their corresponding shared spatial tokens (e.g., <LOC_i>) are observed in WuXi mobility trajectories, Wuxi bus timetables, and Shanghai bus timetables. Each location is represented by a three-bit binary code $abc \in \{0,1\}^3$, where a, b and c indicate whether the location is covered in these three source of data. Fig. 10 illustrates the spatial distribution of these spatial data coverage categories in Shanghai.

To characterize trajectory generation behavior under these spatial data coverage conditions, we consider three complementary metrics.

We first examine the distribution of stay durations to assess how well MobTA captures local activity intensity in different spatial data coverage categories. Specifically, each trajectory is segmented into consecutive stay events, and each event is assigned to the corresponding spatial data coverage category based on its location. For each category, we compute the duration distributions for both real and generated trajectories and measure their discrepancy using the Jensen–Shannon Divergence (JSD), denoted as $\mathbf{JSD}_{dur}$.

Next, we quantify spatial reachability at the trajectory level by computing the trajectory coverage ratio. For each generated or real trajectory, we check whether it visits at least one location grid belonging to a given spatial data coverage category. Let $n^{\text{traj}}_{'abc'}$ denote the number of trajectories satisfying this condition in spatial data coverage category $'abc'$, and let N be the total number of trajectories. The trajectory coverage ratio for spatial data coverage category $'abc'$ is defined as

$$\rho^{\text{traj}}('abc') = \frac{n^{\text{traj}}_{'abc'}}{N}. \tag{41}$$

Finally, we analyze visit frequency by measuring the proportion of total visits falling into each spatial data coverage category. Let $n^{\text{traj}}_{'abc'}$ denote the total number of trajectory visits occurring in spatial data coverage category $'abc'$, and let N_v be the total number of visits across all trajectories. The visit frequency is defined as

$$\rho^{\text{visit}}('abc') = \frac{n^{\text{visit}}_{'abc'}}{N_v}. \tag{42}$$

Together, these metrics provide a fine-grained characterization of MobTA's trajectory generation behavior under different spatial data coverage conditions. Table 10 shows MobTA's performance under different spatial data coverage conditions. Overall, MobTA exhibits stable trajectory generation behavior, with particularly strong performance when locations are covered by all three data sources.

Table 10: MobTA's trajectory generation behavior under different spatial data coverage conditions.

WX Mob	WX Bus	SH Bus	#Loc	JSD_{dur}	$\rho^{\text{real}}_{\text{traj}}$	$\rho^{\text{gen}}_{\text{traj}}$	$\rho^{\text{real}}_{\text{visit}}$	$\rho^{\text{gen}}_{\text{visit}}$
×	×	×	70	0.0176	0.0521	0.0347	0.0084	0.0019
×	×	✓	140	0.0378	0.2649	0.3361	0.1112	0.1323
×	✓	×	0	–	–	–	–	–
×	✓	✓	2	0.2399	0.0568	0.0197	0.0081	0.0015
✓	×	×	15	0.0368	0.0053	0.0245	0.0007	0.0013
✓	×	✓	135	0.0254	0.4767	0.6332	0.1457	0.2456
✓	✓	×	18	0.0104	0.0309	0.0228	0.0038	0.0012
✓	✓	✓	350	0.0188	0.8820	0.8756	0.7220	0.6148

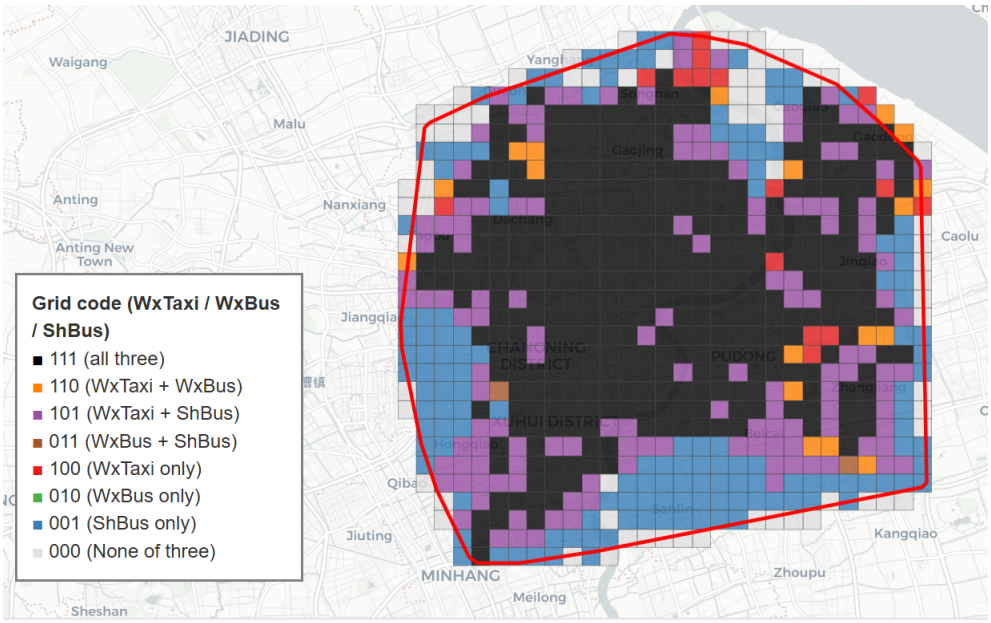


Figure 10: Different categories of Shanghai grids.

M Ablation Study

M.1 Ablation Study for Sub Task Vectors

Table 11 presents ablation results of task vectors applied to different parameter subspaces, where E, L, and H indicate whether task vectors are introduced in the embedding layer, the Transformer backbone, and the LM head, respectively. The results show that the Transformer backbone (L) constitutes the most critical component of the task vector. Introducing task vectors only in this subspace already leads to substantial improvements. However, relying solely on L is insufficient to achieve optimal performance. Further incorporating task vectors in E and H consistently yields additional performance gains. These results show that though the dominant contribution of task vector arises from the Transformer backbone, excluding task vectors in either the embedding layer or the LM head prevents MobTA from achieving optimal generation performance.

M.2 Necessity of Bus-timetable–guided Mobility Backbone

This ablation study investigates whether bus-timetable-based trajectory generation is essential for providing a city-level mobility backbone. We construct a diagnostic setting in which GeoLLaMA^{+} is trained solely on source-city mobility trajectories (Shanghai or Wuxi) and then directly applied to generate target city (Singapore) mobility trajectories. As shown in Table 12, models trained only on source-city mobility trajectory data can merely capture certain coarse-grained mobility characteristics. Moreover, SH (mob) $\rightarrow$SG consistently outperforms WX(mob)$\rightarrow$SG across all metrics, indicating

Table 11: Ablation study on different task vector components.

E	L	H	Dist.	Rad.	Dur.	Loc.
×	×	×	0.2058	0.1493	0.0342	0.1175
×	×	✓	0.2149	0.1557	0.0312	0.1167
×	✓	×	0.1217	0.0979	0.0486	0.0721
×	✓	✓	0.1256	0.0971	0.0464	0.0736
✓	×	×	0.2032	0.1400	0.0369	0.1051
✓	×	✓	0.2173	0.1509	0.0324	0.1098
✓	✓	×	0.1199	0.0929	0.0433	0.0779
✓	✓	✓	**0.0969**	**0.0639**	**0.0206**	**0.0487**

Table 12: Transfer results when only source city real mobility trajectories are used for training

Setting	Dist.	Rad.	Dur.	Loc.
SH (mob) →SG	0.2241	0.2017	0.0479	0.1466
WX(mob) →SG	0.2749	0.2361	0.0717	0.1998

that mobility patterns learned from a larger and more structurally complex city exhibit stronger cross-city generalization. However, compared with methods that incorporate bus timetables (e.g., COLA and MobTA), this setting exhibits clear degradation in spatial related metrics (Dist., Rad., and Loc.). These results highlight the critical role of bus timetables in cross-city trajectory generation. As a planning-driven and highly stable component of urban transportation systems, bus timetables encode the dominant, city-level mobility backbone. In the absence of such data, models trained solely on source-city mobility data struggle to adapt to the spatial layout of the target city, leading to degraded generation quality.

N Hyperparameter Sensitivity Analysis

Fig.11,12,13 present the sensitivity analysis of the weighting coefficients μ_E, μ_L, μ_H in WX $\rightarrow$ SH, which control the contributions of task vectors in the embedding layer, the Transformer backbone, and the output layer, respectively. Overall, MobTA exhibits stable performance across a range of parameter values, indicating that it is not sensitive to hyperparameter choices and demonstrates robustness. Among the three coefficients, μ_L has the most pronounced impact on generation performance. When μ_L is set too small or too large, the performance degrades noticeably, suggesting that the Transformer backbone plays a central role. In contrast, μ_E and μ_H mainly serve as auxiliary adjustment factors, and variations within reasonable ranges have relatively minor effects on overall generation quality.

To further examine whether the default hyperparameters remain stable across different city pairs, we conduct sensitivity analyses across all six source-target city pairs. Table 13 summarizes the stable hyperparameter regions across different city pairs. For all city pairs, the stable range of μ_L covers the default value of 0.5, while μ_E and μ_H remain stable within $[0.01, 0.06]$. These results indicate that MobTA's scaling coefficients are not tuned for a specific city pair; instead, there exist overlapping stable regions across heterogeneous urban environments. All MobTA results reported in the main experiments use the same fixed default configuration, namely $\mu_E = 0.01$, $\mu_L = 0.5$, and $\mu_H = 0.02$.

Based on this sensitivity analysis, we provide a practical guideline for hyperparameter selection in zero-shot scenarios. We observe that MobTA exhibits performance stability across a broad range of backbone coefficients ($\mu_L \in [0.2, 0.8]$). This insensitivity to precise hyperparameter tuning is critical for zero-shot scenarios: it implies that a fixed default configuration (e.g., $\mu_L = 0.5$, $\mu_E = 0.01, \mu_H = 0.02$) can be reliably applied to unseen target cities without the need for validation data. This discrepancy in magnitude of μ_L and μ_E, μ_H is because the LLM backbone is updated via LoRA, while the embedding and output head undergo full-parameter fine-tuning. Consequently, the task vectors derived from full-parameter updates possess larger magnitudes and thus necessitate smaller scaling factors to prevent disrupting the pre-trained feature space.

Table 13: Stable hyperparameter regions across source-target city pairs.

Source-target city pair	Stable range of μ_L	$\mu_E \in [0.01, 0.06]$	$\mu_H \in [0.01, 0.06]$
WX→SH	0.2–0.8	Stable	Stable
SH→WX	0.3–0.9	Stable	Stable
WX→SG	0.2–0.6	Stable	Stable
SG→WX	0.3–0.8	Stable	Stable
SH→SG	0.1–0.6	Stable	Stable
SG→SH	0.2–0.8	Stable	Stable

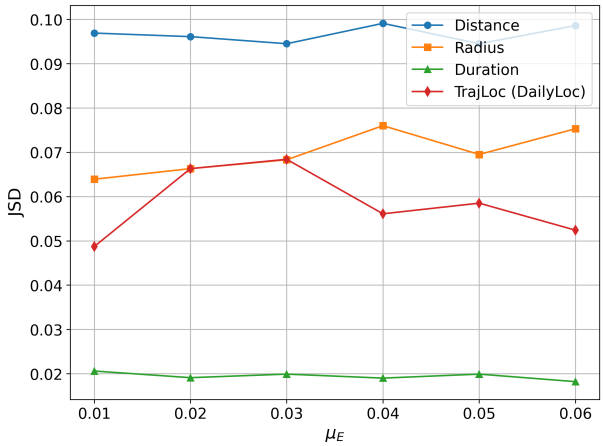


Figure 11: Hyperparameter sensitivity for μ_E.

O Efficiency Comparison for Transferable Models

We compare MobTA with COLA, the only transferable baseline, in terms of computational cost. MobTA is built upon publicly available pretrained LLMs, which can be treated as an external prerequisite rather than a computational cost specifically attributable to MobTA. Under the WX→SH setting, MobTA only needs to learn three task vectors, corresponding to WX-Bus, WX-Mob, and SH-Bus. Each task-vector fine-tuning run converges within 3 epochs and can be trained on a single NVIDIA RTX A5000 GPU with 24GB memory. COLA requires 1908 minutes under its default training setup. In contrast, the three MobTA task-vector fine-tuning runs take 226, 121, and 620 minutes for WX-Bus, WX-Mob, and SH-Bus, respectively. Therefore, MobTA requires 967 minutes when the three task vectors are trained sequentially, and the wall-clock time can be reduced to 620

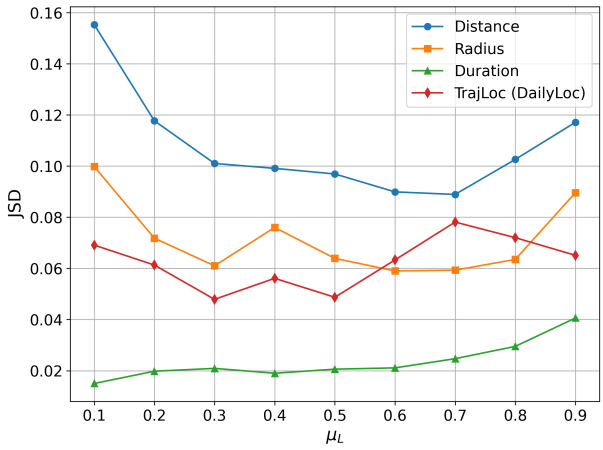


Figure 12: Hyperparameter sensitivity for μ_L.

Table 14: Hyperparameter Configuration.

LoRA Rank	32
LoRA Alpha	64
LoRA Target Module	Q,K,V,Up,Down,Gate
Optimizer	AdamW (8-bit)
Learning Rate	5e-5
Batch Size	4
Num Epochs	3
Max Sequence Length	2048
μ_E	0.01
μ_L	0.5
μ_H	0.02

minutes when they are trained in parallel on three GPUs. After these task vectors are obtained, constructing the target-city mobility generation model only involves linear task-vector composition and does not require additional gradient-based optimization. Compared with COLA, MobTA therefore achieves better generation quality with lower computational cost.

P Implementation Considerations and Technical Nuances

To enhance transparency and facilitate reproducibility, we additionally report implementation considerations and technical nuances.

P.1 Experimental Configurations and Reproducibility Details

We provide the implementation details and experimental configurations here. We utilize the Unsloth library to accelerate the fine-tuning of LLM models. To reduce memory footprint while maintaining optimization stability, we employ the 8-bit AdamW optimizer. Hyperparameter configurations are shown in Table 14. For all city pairs, we use the same default scaling coefficients ($\mu_E = 0.01$, $\mu_L = 0.5$, and $\mu_H = 0.02$). The stability of this default configuration across different city pairs is examined in Appendix N.

P.2 Practical Implementation for Querying Bus Stop Locations via Amap API

When querying geographic coordinates of urban bus stops using the Amap (Gaode Maps) API, we observe that whether the query keyword explicitly includes the term "bus stop" has a decisive impact on localization accuracy. If only the stop name is used as the query keyword, the API frequently returns locations associated with nearby roads, intersections, or surrounding landmarks, with spatial deviations that can exceed several kilometers in certain areas.

In contrast, explicitly appending the keyword "bus stop" to the query consistently yields coordinates that align well with the actual physical locations of bus stops. Fig. 14 illustrates this phenomenon by comparing the returned coordinates for the Shanghai bus stop "Dongfang Road–Pusan Road" with and without the "bus stop" keyword, revealing substantial spatial discrepancies. This issue is particularly pronounced in dense urban areas. If left unaddressed, it would introduce systematic errors into subsequent spatial discretization and eventually, the performance of trajectory generation.

P.3 Reproducibility Risks of Unsloth's New Token "Interpolation" Initialization

During implementation, we observed that Unsloth's function add_new_tokens(..., method="interpolation") internally first calls HuggingFace's tokenizer.add_tokens(new_tokens) and model.resize_token_embeddings(len(tokenizer)), thereby expanding the vocabulary and initializing the corresponding embedding and LM head parameters. Only after this step does it enter the so-called "interpolation" initialization logic.

Conceptually, interpolation initialization is intended to first decompose the string form of a new token into a sequence of existing token IDs using the original tokenizer, compute the mean of their embedding vectors to obtain $\bar{e}_{base}$, and then linearly interpolate between this value and the global

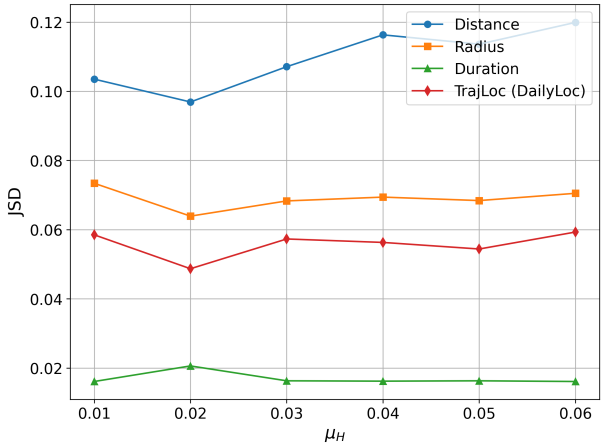


Figure 13: Hyperparameter sensitivity for μ_H.

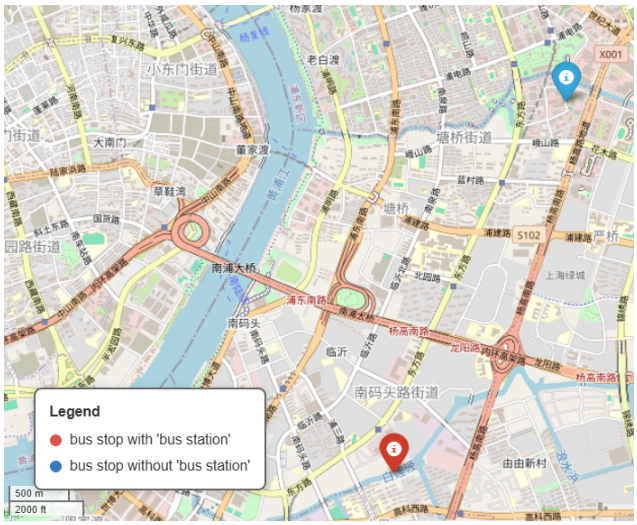


Figure 14: Failure case of Amap API querying.

embedding mean $\bar{e}$:

$$e_{new} = (1 - \alpha)\bar{e} + \alpha\bar{e}_{base}. \tag{43}$$

However, because resize_token_embeddings has already been executed prior to interpolation, the tokenizer at this stage already recognizes the new token. As a result, a string that should have been decomposed into a sequence of old tokens is instead matched directly to the new token itself. Consequently, the embedding vector retrieved during the interpolation step corresponds to the newly and randomly initialized token embedding rather than a semantic average of existing tokens. In practice, the resulting initialization is therefore closer to random initialization with linear mixing than to the intended semantic interpolation initialization.

This implementation order also introduces additional reproducibility risks. HuggingFace's resize_token_embeddings randomly initializes embeddings (and corresponding LM head parameters) for newly added tokens. If the random seed is not explicitly fixed, even two runs with identical data and configurations may yield different initial embeddings for new tokens. These differences can propagate to the embedding and LM head parameters after training and further affect downstream task vectors, ultimately undermining the stability of task vector arithmetic.

Consequently, to avoid such randomness and ensure reproducibility, in MobTA, we initialize the embeddings and LM head parameters of newly introduced spatial tokens using the deterministic global mean of the pre-trained LLM's existing token embeddings and output head weights, following the implementation provided by Unsloth's `mean` initialization option.

P.4 Automatic Untying of Embedding and LM Head After Adding New Tokens in Unsloth

In most mainstream LLM implementations, the input embedding matrix and the output LM head are typically weight-tied, meaning that they share the same parameter matrix to ensure consistency between input and output spaces. However, when extending the vocabulary using add_new_tokens within the Unsloth framework and subsequently fine-tuning the model with PEFT-based methods, we observe that Unsloth automatically unties the embedding and LM head parameters.

As a result, for newly introduced tokens, the input embedding vectors and the corresponding output-layer weights are treated as two independent trainable parameter subspaces. These parameters must be updated separately during training and handled independently during model saving and loading. They can no longer be assumed to represent a single shared parameter matrix or be copied interchangeably. This behavior is particularly important to account for when analyzing parameter updates and constructing task vectors involving newly added tokens.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the paper's main contributions and scope, including the bus-conditioned zero-shot trajectory generation setting, the MobTA task-arithmetic method, and the stability analysis across base and instruction-tuned LLMs.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer:[Yes]

   Justification: The conclusion discusses the scope and limitations of MobTA, including limited applicability to private-vehicle-dominant cities, possible solutions for substantially larger metropolitan areas, the absence of explicit bus-stop attributes or topology, and the lack of explicit physical feasibility constraints.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The main text provides the analysis and proof sketch, while Appendix G gives the detailed derivation and an empirical validation of the projection assumption.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer:[Yes]

Justification: The paper describes the experimental setting, datasets, preprocessing, spatial discretization, baselines, evaluation metrics, default scaling coefficients, LLM backbone, optimizer, LoRA configuration, and other implementation details. It also provides a public code repository to support reproducibility.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The paper provides a public code repository and describes the dataset collection, preprocessing, model training, baseline implementation, evaluation metrics, and hyperparameter settings.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies the datasets, experimental setting, LLM backbone, default scaling coefficients, dataset processing, baselines, hyperparameters choice and evaluation metrics.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

Justification: The paper does not report statistical significance. This is because the dominant source of uncertainty is cross-city and bus-mobility heterogeneity rather than random train/test resampling. To address this source of variability, the paper reports robustness evidence across all source-target city pairs, spatial coverage categories, weak-bus settings, hyperparameter ranges, and LLM families.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper states that experiments are conducted on four NVIDIA RTX A5000 GPUs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer:[Yes]

   Justification: The work is presented as an anonymous submission, uses synthetic trajectory generation to reduce reliance on real target-city user-level mobility data, discusses privacy and fairness considerations in the broader impact statement.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The broader impact statement discusses positive applications such as urban planning, transportation management, and epidemic control, as well as privacy benefits from avoiding target-city real user-level mobility data. It also discusses potential negative impacts from inherited LLM biases and cautions against unassessed use.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not claim to release a high-risk pretrained language model, image generator, or unsafe scraped dataset.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper properly credits the original sources of existing assets, including prior methods, baseline models, LLM backbones, and public data sources. These assets are used under their corresponding open-source licenses, API terms, or other permitted-use conditions, and the paper provides citations, footnotes, or URLs where appropriate.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The paper provides a public code repository for the proposed method. Detailed annotations are in this code.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing experiments or direct research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing experiments or direct research with human subjects, so IRB approval or equivalent human-subject review is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: LLMs are a core component of the proposed approach. The paper describes how LLMs are adapted for trajectory generation through task vectors.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.