Geo-Llama: Leveraging LLMs for Human Mobility Trajectory Generation with Constraints

Siyu Li*, Toan Tran*, Haowen Lin†, John Krumm†, Cyrus Shahabi†,
Lingyi Zhao‡, Khurram Shafique‡, Li Xiong*

*Dept. of Computer Science, Emory University, Atlanta, GA, USA
†Dept. of Computer Science, University of Southern California, Los Angeles, CA, USA
‡Novateur Research Solutions, Ashburn, VA, USA

{siyu.li, viet.toan.tran, lxiong}@emory.edu, {haowenli, jckrumm, shahabi}@usc.edu, {lzhao, kshafique}@novateur.ai

Abstract—Generating realistic human mobility data is essential for various application domains, including transportation, urban planning, and epidemic control, as real data is often inaccessible to researchers due to high costs and privacy concerns. Existing deep generative models learn from real trajectories to generate synthetic ones. Despite the progress, most of them suffer from training stability issues and scale poorly with increasing data size. More importantly, they often lack control mechanisms to guide the generated trajectories under constraints such as enforcing specific visits. To address these limitations, we formally define the controlled trajectory generation problem for effectively handling multiple spatiotemporal constraints. We introduce Geo-Llama, a novel LLM finetuning framework that can enforce multiple explicit visit constraints while maintaining contextual coherence of the generated trajectories. In this approach, pre-trained LLMs are fine-tuned on trajectory data with a visit-wise permutation strategy where each visit corresponds to a specific time and location. This strategy enables the model to capture spatiotemporal patterns regardless of visit orders while maintaining flexible and in-context constraint integration through prompts during generation. Extensive experiments on real-world and synthetic datasets validate the effectiveness of Geo-Llama, demonstrating its versatility and robustness in handling a broad range of constraints to generate more realistic trajectories compared to existing methods.

Index Terms—Human Mobility Trajectory Generation, Spatiotemporal Constraints, LLM

I. INTRODUCTION

Human mobility data, represented as trajectories or sequences of visits, is essential for advancing research and applications in transportation, urban planning, social dynamics, and epidemiology [1]–[3]. However, the collection of large-scale trajectory data is often limited by high costs and privacy concerns, which makes it inaccessible to researchers [4]. This gives rise to an important research area of generating synthetic but realistic trajectories.

One approach for synthetic trajectory generation involves developing micro-simulators, which require careful parameter calibration on sensor data, traffic patterns, and crowd movement statistics [5]. The resulting parameters often lack accuracy and are challenging to configure manually, as such traditional simulation methods relying on heuristics fail to capture complex human mobility patterns [6]. Data-driven approaches such as generative adversarial networks (GANs) and variational point process can generate large-scale and realistic trajectories by learning from real-world data [7]–[10]. However, many existing methods struggle with training stability and do not scale well as data size increases, resulting in the loss of fine-grained spatiotemporal patterns.

Another significant limitation of existing techniques is their lack of control over the trajectory generation process. Incorporating control in trajectory generation to enforce specific visits is crucial as it enables various scenario analysis for downstream tasks. For example, to use synthetic mobility data for epidemiology studies, we may need to generate trajectories that pass through a school building during certain times of the day to simulate school being open and trajectories staying home to represent school closure. This enables analysis of potential outcomes of alternative interventions, which is essential for informed decision-making in fields such as public health. Moreover, by synthesizing trajectories that adhere to specific patterns based on prior knowledge that might be available, the generated data aligns more closely with both real-world behaviors and individual preferences.

Incorporating spatiotemporal constraints such as specific visits in trajectory generation is particularly challenging and remains largely unexplored. Recent studies on controllable trajectory generation have focused on integrating structural constraints, such as road network, or ensuring moving validity by avoiding overlaps with obstacles [11], [12]. These are confined to implicit constraints, typically at a global or route level, and therefore cannot be generalized to enforce explicit visit-level constraints.

A recent work leverages transformer-based models to complete trajectories by infilling missing visits among given visit constraints [13]. While it represents a significant step towards explicitly incorporating visit constraints into mobility data generation, it is only effective in handling a large number of given constraints (known visits) due to its imputation nature and does not generate realistic trajectories when there are only a few visit constraints. Another recent work attempted to propagate a visit constraint into time and location constraints for all other visits, allowing constrained time and location sampling for each visit [14]. The complexity of such propagation restricts this approach to handling only a single constraint, limiting its applicability in more complex scenarios. Alternatively, rather than incorporating control directly during
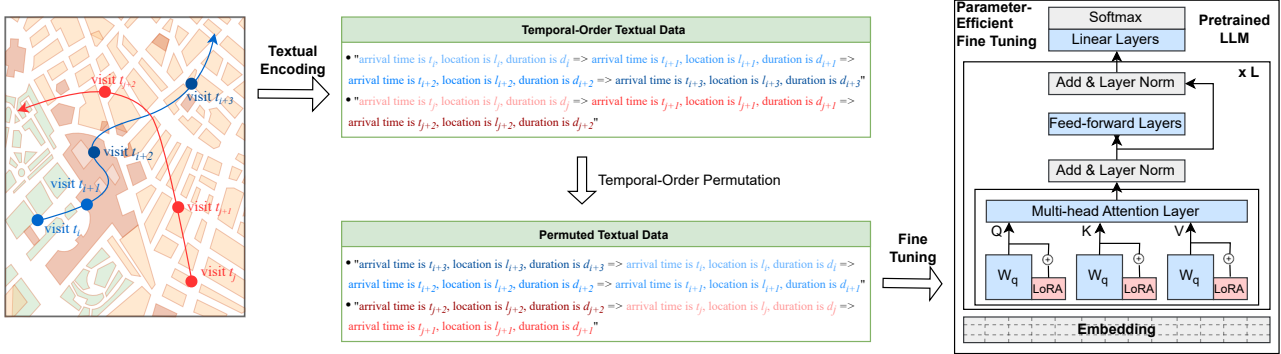
20

Fig. 1: Fine tuning mechanism of Geo-Llama. First, the trajectories are converted into text strings through Textual Encoding. Next, the Temporal-Order Permutation permutes the textual data. The permuted data is then used for fine tuning.
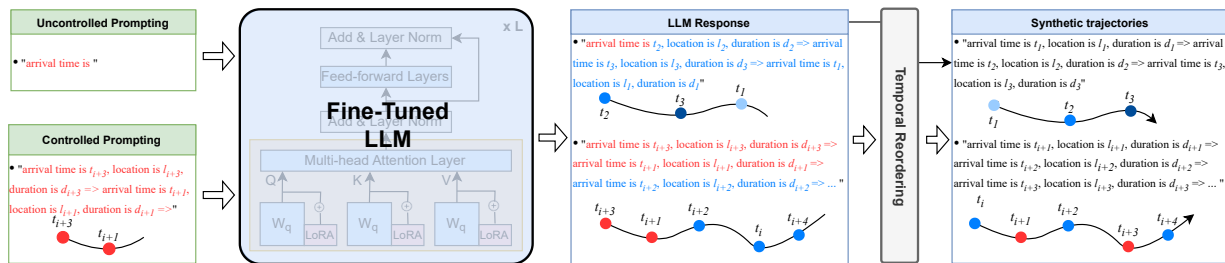

Fig. 2: The generation mechanism of Geo-Llama. It supports both uncontrolled and controlled generation. It generates responses based on the input prompts which are reordered based on the temporal order of visits' arrival time to achieve the final trajectories.

the generation stage, a potential approach [15] is to train the generator without explicit constraints and then apply post-processing to select samples with desired constraints from a large pool of uncontrolled outputs. However, it is highly unlikely for an unconstrained trajectory from a large combinatorial space to satisfy a very specific visit constraint.

Contributions. In this paper, we study the problem of controlled generation of human mobility trajectories with multiple visit constraints. Our trajectories are specified by a series of visits, where each visit is defined by a location and arrival time. Our constraints include one or more required visits with specific time and location ranges. The controllable generation should ensure that both the visit constraints and the statistical properties of the training data are maintained. This will help preserve context-coherent transitions between the specified visits and other visits.

To solve this problem, we propose Geo-Llama, a framework for generating human mobility data with spatiotemporal constraints leveraging large language models (LLMs). The main idea is to represent trajectories as sequences of tokens and fine-tune LLMs on next-token prediction tasks. This process captures the inherent spatiotemporal correlations and movement patterns (as shown in Figure 1). The fine-tuned model can then be used to generate context-coherent trajectories to support both uncontrolled generation and controlled generation that satisfies specified constraints (Figure 2).

Specifically, to enable the model to capture true movement

patterns based on time features (regardless of the order of the visits in the sequence), we propose a visit-wise permutation strategy that randomly permutes the visits in a sequence before using them for fine-tuning the LLMs. In other words, we can consider the problem as learning the spatiotemporal correlations from a set of visits with time and location instead of an ordered sequence of visits. This also enables us to generate a trajectory in which the constrained visit may occur anywhere in the sequence. The generated sequences can be simply reordered based on the time features into ordered visits. Lastly, a thoroughly designed integrity-check algorithm is employed during post-processing to ensure the consistency and coherence of the trajectories (e.g., by preventing overlapping time intervals between consecutive visits).

Our major contributions can be summarized as follows:

- We formulate the controlled trajectory generation problem with multiple visit constraints. The goal is to generate realistic trajectories that represent the patterns in the training data while enforcing the specific visit constraints.
- We introduce Geo-Llama, the first general LLM-based fine-tuning framework for synthetic trajectory generation. It supports both uncontrolled and controlled generation. It utilizes a visit-wise permutation strategy for encoding the trajectories and leverages LLMs to capture the inherently complex spatiotemporal patterns underlying human mobility trajectories while enforcing multiple constraints in a context-coherent manner.

- We conduct extensive experiments comparing Geo-Llama with state-of-the-art generation methods on real-world and synthetic datasets. We demonstrate that Geo-Llama not only enables controlled synthetic trajectory generation but also generates the most realistic trajectories in both controlled and uncontrolled settings. Moreover, Geo-Llama achieves higher data efficiency compared to existing approaches.

II. RELATED WORK

Human Mobility Trajectory Generation. Conventional simulation approaches generate human mobility trajectories using physical estimations based on parameters collected by sensors and surveys [16], [17]. These methods heavily depend on human heuristics and are error-prone. Recent works leveraged large language models (LLMs) for agent based modeling and simulation but simulating large-scale LLM agents remains highly challenging [18].

Deep learning and generative models directly learn and sample trajectories from data-driven latent representations. Early works used recurrent neural networks (RNNs) to predict (generate) the next visited location by modeling spatiotemporal transitions [19], [20]. These models primarily focus on short-term predictions and struggle to generate long, coherent trajectories that capture real-world patterns over time. Generative adversarial networks (GAN) [21] based approaches are becoming prevalent owing to their strong distribution modeling capability via a mini-max game between the generator and discriminator. For example, TrajGAN [9] converts trajectories into two-dimensional images and uses CNN-based GANs for generation. SeqGAN [22] proposes a reinforcement learning framework that adopts the discriminator outputs as rewards augmented by the Monte-Carlo rollout to provide fine-grained supervision on the generator's next location prediction. MoveSim [6] extends SeqGAN by leveraging self-attention networks as the generator backbone and incorporating prior knowledge of human mobility patterns such as Points of Interest (POIs) into the generation process. CSGAN [8] addresses the sampling diversity issues by clustering different motion modalities, enabling more granular learning of mobility patterns and enhancing the realism of generated trajectories. Despite the progress, these approaches still struggle with unstable training and poor scalability as data size increases. Their temporal encoding with fixed time intervals (e.g., every 15 minutes) results in computationally intensive and unnecessarily lengthy sequences, reducing representation efficiency. Such long sequences with repeated locations (corresponding to a stay point) also make it challenging to enforce specific visit-level constraints. To address such limitations, recent works adopt the variants of Deep Spatiotemporal Point Process (DeepSTPP) [23] for trajectory generation [10], [23]. Such approaches represent the trajectory as a sequence of distinct visit points instead of the fixed-interval representation, and therefore better capture the moving patterns by jointly modeling spatial and temporal factors. All of these methods do not consider constraints, and we will compare our approach with representative ones under uncontrolled setting.

Recently, LLMs have demonstrated powerful sequence generation capabilities, enabling the understanding of complex contexts and the utilization of rich prior knowledge. A recent work [24] applies prompt engineering on pre-trained LLMs for trajectory generation by crafting targeted prompts using statistical summaries of demographic information, event types, and event-temporal correlations, emulating travel diaries without requiring additional model training. Another work [25] presents a self-consistent activity pattern identification framework, where the LLM leverages its prior knowledge to evaluate all potential user profiles and POI background information paired with a specific trajectory, resulting in optimal prompt combinations. Retrieval-augmented generation (RAG) is then used to refine trajectory generation within the prompt-engineering paradigm. While these methods rely on rich contextual data like user attributes and POI details, their prompt-engineering nature indicates that the geographic coordinates or temporal dependencies are not explicitly modeled, limiting their applicability for datasets without augmented info. In contrast, our approach directly learns from timestamps and coordinates with LLM fine-tuning, enabling fine-grained modeling of spatiotemporal patterns without requiring extensive external information. Therefore, we do not consider these methods as baselines for comparison.

GPS Trajectory Generation. While human mobility trajectory generation focuses on discrete movement events (usually corresponding to a number of stay points per day) and their spatiotemporal relationships, GPS trajectory generation focuses on producing fine-grained sequences of geographic coordinates at regular, short intervals (e.g., every second). Early approaches [26], [27] relied on perturbing real trajectories with noise or combining trajectory segments from real data to generate synthetic trajectories, which often alter spatial-temporal characteristics and compromise data utility. GAN-based methods [28], [29] are also proposed for GPS trajectory generation and suffer similar issues as for human mobility generation. DiffTraj [30] introduces a spatial-temporal diffusion probabilistic model combined with a Trajectory-specific UNet to effectively capture spatiotemporal patterns from GPS trajectories, ensuring stable training while preserving the original geographic distribution and supporting downstream tasks. Given the distinct objectives and granularity, we do not include GPS trajectory generation methods in our comparisons.

Controlled Mobility Trajectory Generation. In controlled settings, some works start incorporating constraints tailored to domain-specific requirements. FDSVAE [12] learns and operates on latent representations to enforce spatiotemporal validity constraints. ControlTraj [11] is a diffusion-based trajectory generation framework that enforces road network topology constraints to adapt to new geographical contexts. Another line of work [31], [32] addresses privacy concerns in synthetic trajectory generation by enforcing differential privacy constraint, focusing on improving privacy and accuracy tradeoff. These

constraints are either implicit or at a global or route level instead of at a visit level.

To accommodate explicit visit-level constraints, Traj-GPT [13] treats spatiotemporal constraints as fixed templates and leverages encoder-decoder transformer-based models to learn the spatiotemporal statistics of trajectory to fill the gaps. Due to the infilling nature, such an approach is more effective in handling dense constraints (e.g., their experiments masked 20% visits for infilling while having 80% as constraints) but is not suitable when constraints are sparse or in uncontrolled setting when there is no constraint. Therefore, we will not compare it with our approach as a baseline. To incorporate explicit spatiotemporal constraints in a generative setting, Geo-CETRA [14] introduces constraint factorization into spatiotemporal point processes, allowing in-context adjustments to all visits conditioned on the constraint visit. However, the inflexibility of factorization restricts this approach to handling only a single constraint. We will use this as a baseline for comparison under one constraint but our approach is more flexible and can handle multiple constraints.

Controlled Sequence Generation in Other Domains. Beyond mobility data, there are controllable generation work focusing on image data, using constraints like labels or properties, but these methods are not suitable for sequence data with specific constraints [33]. One potentially related line of work is in the natural language processing (NLP) area which aims to generate text that adheres to specific linguistic and structural constraints for downstream applications [15]. Most of these approaches only handle semantic-level constraints which are too implicit to accommodate specific hard requirements such as placing a specific word in a precise position within the text. Some approaches [34] formulate hard constraint enforcement as an infilling problem, but rely on fixed positions, making them not readily applicable for mobility generation with visit constraints.

Next Point-of-Interest (POI) Recommendation. Related to the task of generation, next POI recommendation predicts a user's next point of interest (POI) based on historical mobility data and contextual information, often framed as a sequence recommendation task. A variety of methods have been proposed including probabilistic models, deep learning models, graph-based models, and LLMs [35]–[39], [39]–[42]. Unlike the next POI recommendation, human mobility trajectory generation focuses on producing entire trajectories, often from noise, without requiring historical inputs. Although the sequence-based POI recommendation models can be extended autoregressively to construct full trajectories, they often fail to capture global dependencies. Given these fundamental differences, we do not consider next POI recommendation models as baselines for our work.

III. Preliminaries

In this section, we define the terminology and our problem formulation.

Definition 1: A ***Trajectory Dataset*** is a set of trajectories denoted as $\mathbb{D} = \{\tau^{(i)} | i = 1, .., N\}$, where $\tau^{(i)}$ stands for the i-th trajectory and N is the number of trajectories.

Definition 2: A ***Trajectory*** is defined as a sequence of visits through space and time, denoted as $\tau = \{v_j \mid j = 1, \ldots, L\}$, where each visit $v_j = (t_j, l_j, d_j)$ consists of the arrival time $t_j \in \mathbb{R}$, the location $l_j \in \mathcal{L}$, and a feature vector $d_j \in \mathbb{R}^p$ representing attributes such as stay duration. Here, L is the total number of visits in trajectory τ.

In addition to the visit-based representation, a trajectory can also be expressed in a ***fixed-time-interval representation*** as $\tau' = \{l_k \mid k = 1, \ldots, T\}$, where T is the total number of time intervals, and $l_k \in \mathcal{L}$ is the location associated with the k-th time interval. For example, if we model daily trajectories, and use fixed time interval $\Delta t = 15$ minutes, it results in $T = \frac{24 \times 60}{\Delta t} = 96$. Consider a trajectory in the visit-based representation $\tau = \{..., (10\text{am}, l_1), (11\text{am}, l_2), ...\}$, the corresponding fixed-time-interval representation is $\tau' = \{..., l_1, l_1, l_1, l_1, l_2, l_2, l_2, l_2, ...\}$.

Fixed time interval representation creates redundant lengthy sequences and inflexibility in constraint enforcement. Visit-based trajectories offer flexible, compact representations, which we adopt for this work.

Definition 3: A ***Spatiotemporal Visit Constraint*** is a required visit denoted as $c = (l_c, t_{c1}, t_{c2})$, where l_c is a specific location, $[t_{c1}, t_{c2}]$ is the range for the arrival time. A trajectory τ satisfies a visit constraint c if: $\exists v_i$ s.t. $l_i = l_c$ & $t_{c1} \leq t_i \leq t_{c2}$. i.e. the trajectory must contain a visit to l_c that arrives between t_{c1} and t_{c2}.

Definition 4: ***Controlled Synthetic Trajectory Generation*** is to generate a synthetic set $\mathbb{D}'$ which preserves the spatiotemporal patterns of the original dataset $\mathbb{D}$ while satisfying a set of given visit constraints. The problem can be formalized as Equation 1, where ψ is a function that measures the similarity between the trajectory properties of $\mathbb{D}'$ and $\mathbb{D}$, such as distributional divergence (e.g., JSD) over temporal, spatial, or structural statistics. While in rare cases where the constraint set C happens to exactly match a subset of $\mathbb{D}$, the optimal solution $\mathbb{D}'$ may coincide with $\mathbb{D}$, the more typical scenario involves partial or contextually shifted constraints, requiring the model to generate trajectories that resemble but are not identical to the original dataset.

$$\text{maximize} \quad \psi(\mathbb{D}', \mathbb{D}) \quad \text{s.t.} \quad \mathbb{D}' \text{ satisfies } C \quad (1)$$

*Definition 5: **Time & Location Discretization***: The location domain is divided into equal-sized grid cells, each represented by a unique ID from the set $\mathcal{G} = \{g_1, \ldots, g_{|\mathcal{G}|}\}$. Time domain is discretized into fixed intervals, forming the set $\mathcal{I} = \{i_1, \ldots, i_{|\mathcal{I}|}\}$. After discretization, each location $l_j \in \mathcal{G}$ and time point $t_j \in \mathcal{I}$ represent distinct tokens in our trajectory dataset, reducing complexity and leveraging the model's strength in handling categorical data.

IV. Methodology

In this section, we describe our Geo-Llama framework which leverages pretrained LLMs for realistic synthetic tra-

jectory generation. There are two main stages: fine-tuning and generation.

A. Fine-Tuning

Figure 1 illustrates the fine-tuning mechanism of Geo-Llama. Each trajectory is first converted into textual data. The temporal order of the sequences is then permuted to enable the LLM to learn the spatiotemporal correlations rather than the order of the visits. A pre-trained LLM is fine-tuned using parameter-efficient techniques. This study focuses on decoder-only LLMs, which represent the state-of-the-art architecture for many generation tasks [43].

Textual Encoding. Textual Encoding converts trajectory $\tau^{(i)}$ into textual representation $\mathbb{S}^{(i)}$. To simplify, we consider d_j of visit (t_j, l_j, d_j) as a univariate feature, particularly the duration, which represents how long the person stayed at l_j. All time and locations adopt discretized representations where finer-grained ones increase computational complexity, while coarser ones may degrade performance. Each visit is represented by a simple text format – "`arrival time is $t_j$, location is $l_j$, duration is $d_j$`". The visits are separated by a special token and whitespaces, i.e., "` => `". Additionally, each text sequence representing a trajectory begins with "`<BOS>`" (Beginning Of Sentence token) and ends with "`<EOS>`" (End Of Sentence token). In this representation, the text sequence length depends only on the number of visits. Therefore, this method can effectively handle sporadic data. Meanwhile, previous methods heavily rely on fixed-interval location sequences, which can be challenging to manage high-resolution intervals or infrequent visits [20], [22].

Temporal-Order Permutation. A critical challenge of decoder-only LLMs is their heavy dependence on the order of sequences. To enable controllable generation and enable the LLM to learn the spatiotemporal correlations regardless of the order of the visits, we randomly permute the visits of $\mathbb{S}^{(i)}$. The permuted text sequence of $\mathbb{S}^{(i)}$ is denoted as $\mathbb{S}'^{(i)} = permute(\mathbb{S}^{(i)})$. This permutation only changes the order of visits without modifying any attributes or the order of tokens within each visit. Therefore, the semantic meaning remains the same. This way, the permuted data enhances the LLM's ability to learn the intrinsic properties and spatial-temporal relationships using the arrival times and durations rather than the visit order. For example, in the trajectory "`9am at work, 5pm at home`", the specific order of visits is not inherently crucial to their semantic meaning. By permuting, the model can focus on directly learning the relationships between location and time.

Parameter-Efficient Fine-Tuning. While we demonstrate Geo-Llama using Llama2 [44], an open-source decoder-only LLM, our framework can finetune any LLM using the permuted textual data. Geo-Llama employs the standard causal language modeling task for fine-tuning, known as next-token prediction. The loss function is the cross entropy across the model's vocabulary, presented in Equation 2, where p is the probability, N is the number of trajectories, θ stands for the trainable parameters, and x_j represents tokens in the permuted text sequence $\mathbb{S}'^{(i)}_N$.

$$\mathcal{L}(\theta) = -\sum_{i=1}^{N} \sum_{x_j \in \mathbb{S}'^{(i)}} \log p\left(x_j | x_{j-1}, x_{j-2}, ..., x_1, \theta\right) \quad (2)$$

For computing efficiency, we employ a Parameter-efficient fine-tuning (PEFT) technique, known as Low Rank Adaption (LoRA) [45]. Specifically, given a pretrained weight matrix $W_0 \in \mathbb{R}^{dim \times k}$, LoRA introduces two trainable matrices $A \in \mathbb{R}^{r \times k}$ and $B \in \mathbb{R}^{dim \times r}$. r is the rank, which is very small compared to dim and k. This method constrains the update of W_0 by its low-rank decomposition $W_0 + \Delta W = W_0 + BA$. A regular forward pass of input x, $h = W_0 x$, can be modified using LoRA as:

$$h = W_0 x + \Delta W x = W_0 x + BAx$$

B. Generation

Figure 2 illustrates the generation process of Geo-Llama, supporting both controlled and uncontrolled prompts. The fine-tuned LLM generates complete sequences based on the input prompts, which are then temporally reordered to construct the final trajectories. A trajectory integrity check is performed to ensure coherence in the generated trajectories, preventing issues such as overlapping time intervals between consecutive visits.

Uncontrolled Prompting. As all the text sequences for fine tuning start with "`arrival time is`", uncontrolled prompting uses this exact phrase as prompts. The entire trajectory is then generated by the LLM.

Controlled Prompting. For controlled generation, initial prompts are created based on the given visit constraints, which means all visit constraints are strictly enforced. As mentioned in Def. 3, a constraint c includes a time period from t_{c1} to t_{c2}. To simplify, we randomly sample a specific timestamp within the constrained period. We then convert the set of constraints into prompt using the same textual encoding that converts a trajectory into textual representation. Next, we feed the prompts into the fine-tuned LLM. The remainder of the trajectory is then generated by the LLM. Our temporal-order permutation strategy enables the LLM to generate visits with arrival time both before or after the time of the constraints. It is worth noting that without permutation, the LLM would be restricted to generating visits that occur only after the constrained period, which would be limited in real-world trajectory generation.

LLM Generation. Let $x^{(k)} = (x_1, x_2, ..., x_k)$ be the current prompt, consisting of tokens after the tokenization. The LLM generation is an iterative process as follows. First, the probability of next tokens $p^{(k+1)}$ are computed by:

$$p^{(k+1)} = \text{softmax}\left(\frac{f(x^{(k)}; \theta)}{T}\right),$$

where T is a temperature parameter, controlling the randomness of the sampling process. Higher T increases diversity by exploring less likely tokens, which may lead to more varied but less coherent trajectories, while lower T reduces randomness, favoring more likely tokens and resulting in more consistent but potentially repetitive trajectories. We sample based on the given probability $p^{(k+1)}$ to obtain the next predicted token

x_{k+1}. The process then repeats with the updated prompt $x^{(k+1)} = (x^{(k)}, x_{k+1})$. This iterative procedure continues until the predicted token is <EOS>. Notably, due to the inherent randomness in sampling from $p^{(k+1)}$, the generated trajectories may vary even when the same prompts are used.

Integrity Check. We first perform prompt format validation and reordering, followed by an integrity check to ensure the validity of the generated trajectories. Specifically, we detect and resolve overlaps in adjacent visit times, and correct out-of-bound location grid IDs by replacing them with nearby valid locations.

V. EXPERIMENTS

In this section, we evaluate the performance of Geo-Llama on both real-world and synthetic datasets. We compare Geo-Llama with state-of-the-art mobility generation methods across uncontrolled, controlled, and data-efficiency study settings. Notably, our framework is uniquely designed to satisfy visit constraints, whereas most baseline methods, except Geo-CETRA, are not inherently suited for controlled generation. Since Geo-CETRA can only handle a single visit constraint during controlled generation, we include a stand-alone comparison with this baseline under the single-visit constraint setting. To ensure comparability in the controlled setting, we employ a straightforward approach by forcibly inserting (FI) the constrained visits into the trajectories generated by the baseline methods. Our code can be accessed at: `https://github.com/Emory-AIMS/Geo-Llama`.

TABLE I: Training Dataset Statistics for Geolife and MobilitySyn

Dataset	Total Daily Trajectories Used	Average Stay Points per Trajectory
Geolife	7,000	4.2
MobilitySyn	7,000	6.7

A. Datasets & Preprocessing

We conduct extensive experiments on the benchmark Geo-Life dataset and a synthetic MobilitySyn dataset.

- **GeoLife**. This GPS trajectory dataset [46] was collected by MSRA with 182 users over five years (from April 2007 to August 2012). It contains 17,621 trajectories, where each trajectory is a sequence of GPS records including the timestamp, latitude, and longitude. Here we only focus on data in the greater Beijing area.
- **MobilitySyn**. A realistic simulation of 5,000 agents performing daily movements across a metropolitan area over the course of a week. The simulation generates second-by-second GPS records for each agent, describing their recurring daily visits to various locations. The generated trajectories reflect plausible daily movement patterns.

Staypoints Calculation. Staypoints are defined as locations where an individual stays within a specified radius for a duration exceeding a predefined threshold. From the raw trajectory data, we extract staypoints from GPS records using a 1-kilometer radius for Geolife and a 0.1-kilometer radius for MobilitySyn, with a minimum duration of 15 minutes.

The smaller radius for MobilitySyn accounts for its dense but geographically compact urban area, ensuring that small movements between nearby staypoints are not overlooked. In contrast, Geolife spans larger metropolitan regions, where a 1-kilometer radius better captures significant location transitions. The 15-minute threshold is used across both datasets to reflect similar visit patterns. Next, we construct single-day trajectories, discarding those with fewer than three stay points. Table I shows the number of daily trajectories we used and average number of staypoints per trajectory in both datasets.

Trajectory Representations. Our baselines include approaches with both 1) fixed interval representation which are sequences of 96 visited locations per day, with fixed-length 15-minute intervals, and 2) sequence of visits representations. Geo-CETRA and Geo-Llama use sequences of visits, where the sequence length varies based on the number of visits, For Geo-Llama, each visit is represented by discretized location and time. The grid size is set to be a 1-kilometer box for Geolife and a 0.1-kilometer box for MobilitySyn, and the time interval is set to be 15-minutes. These setups align with the parameters used for staypoint calculation. Geo-CETRA uses continuous time and location, we perform the same discretization on its output for comparison.

Visit Constraint Generation. For each stay point sequence, vist constraints are generated by randomly sampling a subset of visits within the trajectory. For each sampled visit, a corresponding location and time window are created, narrowly centered around the visit. This approach ensures that the visit constraints align with the actual data distribution and reflect realistic human mobility patterns.

B. Baselines

We evaluate the performance of our model against the following seven state-of-the-art baselines:

- **GRU** [47] and **LSTM** [48]: Recurrent neural networks that are efficient for sequential data generation. These models are able to predict the next location based on historically visited locations.
- **Transformer** [49]: A powerful deep learning model used in various natural language processing (NLP) and computer vision tasks that leverages self-attention mechanisms. A multi-layer Transformer decoder is utilized for location prediction (generation).
- **Variational Autoencoders (VAE)** [50]: A generative model that learns to encode input data into a probabilistic latent space and reconstruct it. It converts the trajectories into 2D matrices, where each cell represents the location at a specific time step.
- **SeqGAN** [22]: A sequence GAN that introduces a discriminator as a reward signal to guide the gradient policy update of the generator, which performs the next location prediction task based on the past states.
- **MoveSim** [6]: An extension from SeqGAN. It introduces the attention module as the generator and incorporates domain knowledge such as POI information in the model. For a fair comparison, we remove POI embedding in the

25

Dataset	Model	Trajectory-level (↓)					Global-level (↓)	
		Distance	G-radius	Duration	DailyLoc	I-rank	G-rank	Transition
GeoLife	GRU	0.0074	0.0630	0.0582	0.0839	0.0182	0.0139	0.0444
	LSTM	0.0118	0.0594	0.1186	0.1197	**0.0091**	0.0139	0.0501
	Transformer	0.0099	0.0460	0.1234	0.1413	0.0175	0.0173	0.0618
	VAE	0.4350	0.6785	0.2660	0.6900	0.0414	0.1930	0.0647
	MoveSim	0.0039	0.0447	0.0693	0.1875	0.0216	0.0207	0.0539
	SeqGAN	0.0076	0.1324	0.0541	0.3849	0.0273	0.0208	0.0599
	Geo-CETRA	0.0100	0.2479	0.1743	0.1168	0.0220	**0.0000**	**0.0041**
	Geo-Llama	**0.0010**	**0.0120**	**0.0014**	**0.0009**	0.0161	0.0173	0.0267
MobilitySyn	GRU	0.0093	0.1703	0.0617	0.1150	**0.0000**	0.0173	0.0085
	LSTM	0.0138	0.1716	0.1084	0.1842	**0.0000**	0.0182	0.0071
	Transformer	0.0337	0.2642	0.1608	0.3590	**0.0000**	0.0236	**0.0060**
	VAE	0.5347	0.6819	0.2817	0.6931	0.0249	0.2505	0.0127
	MoveSim	0.0089	**0.0474**	0.5191	0.0948	0.0473	0.0207	0.0275
	SeqGAN	0.0063	0.1584	0.4637	0.0820	0.0034	0.0277	0.0109
	Geo-CETRA	0.0146	0.3620	0.2545	0.1706	0.0321	**0.0000**	0.0062
	Geo-Llama	**0.0010**	0.0689	**0.0231**	**0.0181**	**0.0000**	0.0139	0.0099

TABLE II: Results of uncontrolled trajectory generation.

MoveSim implementation, as we do not have access to POI information.

- **Geo-CETRA** [14]: A spatiotemporal point process-based framework for trajectory generation that incorporates constraint factorization and beam decoding to produce realistic trajectories under visit constraints. Due to the fact that it is limited to handling a single constraint and works on continuous geo-coordinates, we compare it with Geo-Llama under the single visit constraint setting by aligning generated continuous coordinate with discrete grid IDs.

C. Evaluation Metrics

As constraints are strictly enforced and satisfied in our controllable generation framework, we focus our evaluation on the realism and contextual coherence of the generated trajectories rather than constraint satisfaction. Following previous works [6], [8], we employ a variety of metrics to assess the quality of the generated trajectories by comparing key mobility pattern distributions between the generated and real trajectories.

Distance is the distribution of cumulative travel distance per user each day. **G-radius** (radius of gyration) represents the distribution of spatial range of user's daily movement. **Duration** is the distribution of the duration per visited location. **DailyLoc** is the distribution of the number of visited locations per day for each user. **G-rank** is the global distribution of number of visits per location for top-100 visited locations. **I-rank** is an individual version of the G-rank.

For the above properties, the Jensen-Shannon divergence, $\text{JSD}(\mathbb{D}, \mathbb{D}') = h\left(\frac{\mathbb{D}+\mathbb{D}'}{2}\right) - \frac{h(\mathbb{D})+h(\mathbb{D}')}{2}$, is applied to measure the distance or discrepancy between the distributions of generated $\mathbb{D}'$ and real trajectory $\mathbb{D}$, where h represents the Shannon information. The lower the JSD, the more realistic the generated trajectories.

In addition, we also have metrics for comparing transition distribution between the generated and real trajectories. **Transition** is the probability distribution of a trajectory transitioning from location l_1 to location l_2 over the set of all discretized locations $\mathcal{G}$. **Top-K Transition** are transitions involving top-K locations included in the constraints, which we introduce to better evaluate whether the methods incorporate the constraints with realistic transitions.

We use the Frobenius norm of their difference: $\|P_{\mathbb{D}} - P_{\mathbb{D}'}\|_F = \sqrt{\sum_{l_1=1}^{|\mathcal{G}|} \sum_{l_2=1}^{|\mathcal{G}|} |P_{\mathbb{D}}(l_1, l_2) - P_{\mathbb{D}'}(l_1, l_2)|^2}$ as the discrepancy metric. The lower the better. We note that G-rank and Transition correspond to global level while others correspond to trajectory-level patterns.

D. Implementation Details

- **GRU, LSTM, and Transformer** are trained for 200 epochs using the Adam optimizer with a learning rate of 0.001. Models share an embedding size of 256, with GRU and LSTM using 6 layers of 512 hidden units, while Transformer adopts a decoder-only architecture with 4 layers and 4 attention heads.

- **VAE** uses three fully connected layers in both the encoder and decoder, each with 128 units. The latent dimension is 64, and training was conducted over 200 epochs with a batch size of 256.

- **SeqGAN** includes an LSTM-based generator trained with 16-dimensional embeddings and 16 hidden units. Discriminator employs diverse filter sizes and counts, with 40 epochs of pre-training and 20 epochs of adversarial training.

- **MoveSim** builds upon SeqGAN with similar parameter settings but adds an attention-based generator with a location embedding size of 256, a time embedding size of 16, and a hidden dimension of 64.

- **Geo-CERTA** employs a conditional Gaussian Mixture Model with 8 spatial and temporal mixture components and a beam search strategy with a beam size 10 and top k=3. Optimization is performed with the Adam optimizer, a learning rate of 0.01, a scheduler with a decay factor of 0.99, and z-score normalization for input data.

- **Geo-Llama** fine-tunes the Llama-2-7b-chat-hf model using LoRA with a batch size of 48, a learning rate of 0.00001, LoRA alpha32, LoRA dropout 0.02, LoRA r 16, and 20 epochs. Sampling uses a temperature of 1.2 by default. We note that our framework is general and can use any pretrained LLMs.

Dataset	Model	Trajectory-level (↓)					Global-level (↓)		
		Distance	G-radius	Duration	DailyLoc	I-rank	G-rank	Transition	Top-K Transition
GeoLife *(w 1-3 Visit Constraints)*	GRU w/ FI	0.0078	0.0997	0.0479	0.1618	0.0157	0.0173	0.0385	0.0477
	LSTM w/ FI	0.0145	0.1063	0.1151	0.2183	0.0115	0.0242	0.0438	0.0521
	Transformer w/ FI	0.0080	0.0458	0.0962	0.0980	0.0232	0.0242	0.0469	0.0555
	VAE w/ FI	0.3286	0.6452	0.2719	0.6849	0.0221	0.1324	0.0595	0.0781
	MoveSim w/ FI	0.0030	0.0173	0.0317	0.1646	0.0216	0.0139	0.0349	0.0456
	SeqGAN w/ FI	0.0051	0.0613	0.0140	0.3059	0.0364	0.0208	0.0333	0.0438
	Geo-Llama	**0.0007**	**0.0098**	**0.0014**	**0.0016**	0.0061	**0.0000**	**0.0209**	**0.0384**
MobilitySyn *(w 2-5 Visit Constraints)*	GRU w/ FI	0.0089	0.2551	0.0381	0.1784	**0.0000**	0.0251	0.0069	0.0146
	LSTM w/ FI	0.0132	0.2639	0.0806	0.2254	**0.0000**	0.0277	0.0063	**0.0140**
	Transformer w/ FI	0.0243	0.2894	0.1225	0.3132	**0.0000**	0.0286	**0.0057**	0.0152
	VAE w/ FI	0.3044	0.6555	0.2624	0.6920	0.0248	0.0991	0.0124	0.0150
	MoveSim w/ FI	0.0066	0.0815	0.0995	0.1000	0.0360	0.0243	0.0166	0.0464
	SeqGAN w/ FI	0.0051	0.1695	0.0487	**0.0816**	0.0034	**0.0242**	0.0076	0.0174
	Geo-Llama	**0.0021**	**0.0704**	**0.0217**	0.1784	**0.0000**	0.0278	0.0073	0.0154

TABLE III: Results of controlled trajectory generation with multiple visit constraints. All baseline approaches forcibly insert (FI) specified visits from the constraints.

Dataset	Model	Trajectory-level (↓)					Global-level (↓)		
		Distance	G-radius	Duration	DailyLoc	I-rank	G-rank	Transition	Top-K Transition
GeoLife	Geo-CETRA w/ Single Constraint	0.0093	0.2772	0.1789	0.1091	**0.0140**	**0.0005**	**0.0038**	**0.0324**
	Geo-Llama w/ Single Constraint	**0.0011**	**0.0132**	**0.0011**	**0.0028**	0.0174	0.0208	0.0295	0.0581
MobilitySyn	Geo-CETRA w/ Single Constraint	0.0139	0.4054	0.2616	0.1601	0.0208	**0.0000**	**0.0059**	0.1036
	Geo-Llama w/ Single Constraint	**0.0024**	**0.0902**	**0.0246**	0.2038	**0.0000**	0.0278	0.0083	**0.0188**

TABLE IV: Results of controlled trajectory generation under single visit constraint. Geo-CETRA is designed to handle only one constraint.

E. Results

Uncontrolled Generation. Table II compares Geo-Llama with all baselines under the uncontrolled setting. It shows that Geo-Llama consistently outperforms all baselines across the majority of location-based metrics (Distance, G-radius, G-rank, and I-rank), time-based metrics (Duration and DailyLoc), as well as transition metrics. This highlights the superior capability of the finetuned LLM of Geo-Llama in capturing the spatiotemporal patterns compared to other models. The significant performance advantage on time-based metrics can be also attributed to Geo-Llama's representation of the trajectory as a sequence of visit with time and location as separate features (in contrast to the fixed interval representation), resulting in shorter sequences, averaging 5 visits per trajectory. This advantage can be also observed in Geo-CETRA which also uses sequence of visit representation. In contrast, most other baseline methods use 96-time-step fixed intervals, which may obscure temporal patterns due to the repeated locations for a single staypoint, although some still perform adequately on location frequency metrics.

Among the baselines, VAE struggled to converge and effectively model mobility patterns potentially because sequences are transformed into an ill-suited 2D image format. Despite relying on the sequence model as the generator, SeqGAN and MoveSim encountered the inherent training instability of GAN-based methods, which yields poor sampling diversity and degraded performance on both time and location metrics. On the other hand, GRU, LSTM, and Transformer not only effectively capture location-based patterns, but also outperform VAE, MoveSim, and SeqGAN on time-based metrics, underscoring their effectiveness in handling temporal information.

In uncontrolled settings, we conclude that Geo-Llama consistently outperforms the baselines in generating realistic trajectories across a range of metrics and datasets.

Controlled Generation. We compare Geo-Llama with all baselines under the controlled setting. We separate the comparison under multiple and single visit constraints. For multiple visit constraints, we use 1-3 visit constraints per trajectory for GeoLife dataset and 2-5 visit constraints for MobilitySyn dataset respectively. We will also evaluate the impact of number of visit constraints in later experiments. All baselines forcibly insert (FI) the specified visits after generation to satisfy the constraints. For single constraint, we compare with Geo-CETRA since it only supports a single visit constraint.

1) Comparison of Geo-Llama under Multiple Visit Constraints: Table III demonstrates that the trajectories generated by Geo-Llama under multiple visit constraints per trajectory remain contextually coherent, and reflect true underlying spatiotemporal patterns. In contrast, the trajectories generated by baseline models, display unrealistic time, and transition patterns due to the forcible insert of constrained visits. Geo-Llama's ability to model spatiotemporal dynamics independently of sequence order, relying solely on the temporal attributes of each event, allows it to integrate external visit constraints in a context-aware manner, enabling the generation of realistic trajectories. In addition, while forcibly inserting specified visits from constraints can boost location-based metrics (e.g., G-rank and I-rank) of baseline methods by introducing commonly occurring real-world locations, these approaches fail to preserve correct temporal order and smooth transitions between visits because the forcible insertion disrupts the realistic spatiotemporal relationships, leading to unrealistic and contextually incoherent trajectories.

Overall, by leveraging our temporal-order-permutation strategy, Geo-Llama not only consistently enforces the specified constraints but also generates more realistic trajectories than baselines.

2) Comparison of Geo-Llama under Single Visit Constraint with Geo-CETRA: Table IV indicates that Geo-Llama demon-
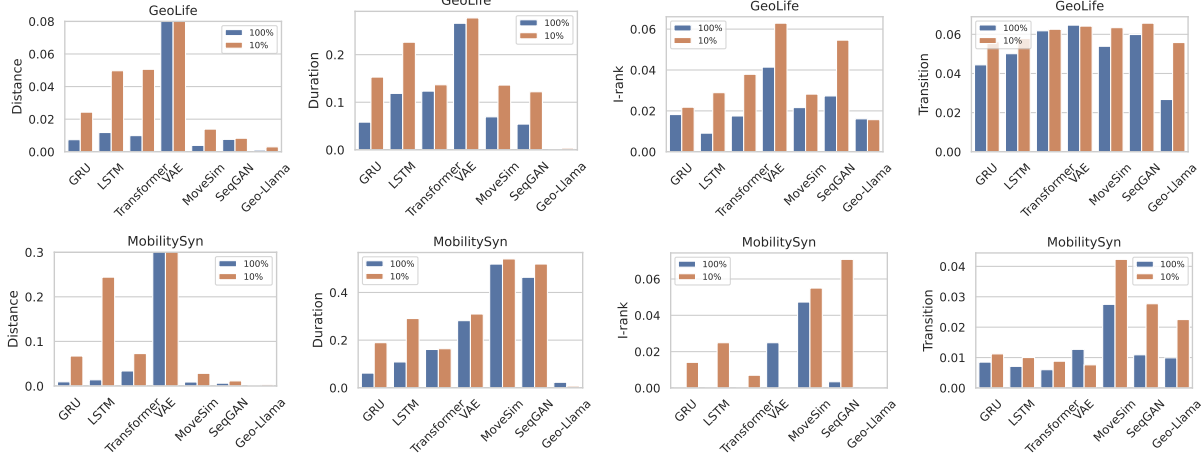
Fig. 3: Data-efficient learning study. We crop some extremely high values of VAE for better visualization.

strates superior performance in both spatial (Distance, G-radius) and temporal (Duration, DailyLoc) metrics compared to Geo-CETRA, indicating more consistent modeling of spatiotemporal patterns. While Geo-CETRA excels in transition-based metrics, particularly Top-K Transition, due to its constraint decomposition algorithm's ability to capture realistic transitions between constrained and subsequent events, this comes at the cost of increased spatial and temporal distortion. The higher G-radius and lower DailyLoc for Geo-CETRA highlight its inability to maintain precise spatial locations and daily visit patterns. Furthermore, Geo-CETRA's constraint decomposition algorithm is inherently inflexible, supporting only a single visit constraint, which limits its applicability for more complex trajectory generation tasks. *Overall, Geo-Llama outperforms Geo-CETRA in producing more context-coherent trajectory under a single visit constraint and is also more flexible in taking more complicated visit constraints.*

Data-efficient Learning. We train Geo-Llama and all baselines on 10% of both datasets (∼700 trajectories) to evaluate their data-efficient learning capabilities. Figure 3 presents the model performance in different metrics using 100% and 10% of the data. When trained on 10% of the dataset, Geo-Llama has only a slight performance drop across most metrics compared to training on the full dataset. In contrast, baseline models have a significant performance drop as the training data size decreases. For instance, the distance JSD of the baseline methods increases by 2 to 10 times, while Geo-Llama's remains stable. This data-efficient feature of Geo-Llama may result from leveraging pretrained LLMs, whereas baseline models require training from scratch. The results demonstrate that *Geo-Llama is data-efficient whereas the baseline methods show significant performance degradation with limited training data.*

VI. PARAMETER STUDIES

Impact of Temporal-Order Permutation. To validate our hypothesis that adopting temporal-order permutation

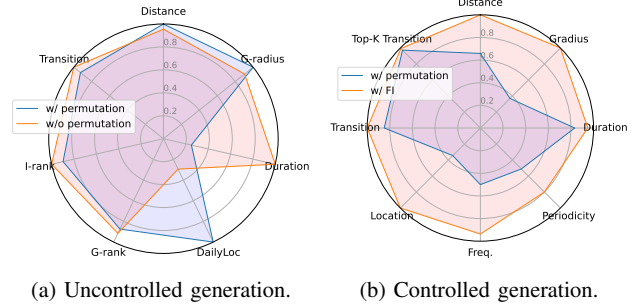(a) Uncontrolled generation.   (b) Controlled generation.

Fig. 4: Impact of temporal-order permutation. The reported performance metrics represent average values across both datasets. For each dataset, the values are scaled to a range of 0 to 1 by dividing them by their respective maximum values.

enhances the capture of true temporal features among visits regardless of sequence orders, we train two identical models on the same dataset and with same hyperparameter setting: one incorporating permutation and the other not. Figure 4a shows the results for uncontrolled generation. Permutation has a negligible effect on most metrics, including G-radius, Distance, G-rank, G-rank, and Transition. However, notable differences are observed in Duration and DailyLoc. With permutation, Geo-Llama performs better on Duration, likely because it allows the model to capture the relationship between arrival time and duration without relying on sequence order. In contrast, the model without permutation outperforms on DailyLoc, as preserving temporal order simplifies the recognition of periodic patterns.

For controlled generation without temporal-order permutation, even if we can still use the constraints as prompts, Geo-Llama would not generate realistic trajectories because all the generated visits will be after the arrival time of the last constrained visit. To construct a baseline, we first conduct uncontrolled generation using Geo-Llama without permutation and then forcibly insert (FI) the constrained visits into the

trajectories. Figure 4b illustrates each method's performance, demonstrating that Geo-Llama with permutation significantly outperforms the forcible-insertion approach across all metrics. Notably, employing permutation yields around 80% improvement in both G-rank and G-radius. Overall, for uncontrolled generation, *Geo-Llama yields competitive performance with or without permutation. However, for controlled generation, the proposed permutation strategy enables LLMs to generate more realistic trajectories compared to forcible insertion.*

Impact of Location Popularity in Constraints. Intuitively, constraints involving popular (frequently visited) locations are easier to satisfy. To investigate this relationship, we categorize constraints into three types based on location visit counts in the training data: frequent, moderate, and infrequent. We then rank locations by their total visits and select the top 40 (most visited), middle 40, and bottom 40 (least visited) to construct the corresponding constraint sets. Figure 5a shows that the synthetic trajectories are more realistic when their constraints contain more frequently visited locations, as expected. Notably, an 80% performance disparity exists between frequent and infrequent constraints for Top-K Transition, G-radius, and DailyLoc, potentially because more frequent locations appear more in the training data, enabling the LLMs to learn patterns of those locations more effectively. Overall, *Geo-Llama tends to perform better with constraints involving more frequently visited locations*.
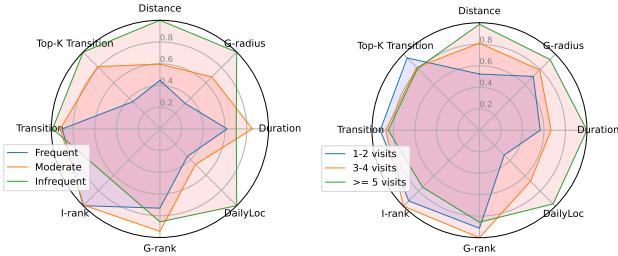
(a) Varying location frequency.　　(b) Varying numbers of visits.

Fig. 5: Impact of constraint visits. The reported performance metrics represent average values across both datasets, with each metric scaled by its maximum value within each dataset.

Impact of Number of Visits in Constraints. Figure 5b illustrates the performance of Geo-Llama with varying number of visits in the constraints. Trajectories containing 1-2 constrained visits exhibit significantly lower discrepancy in Distance, G-radius, Duration and DailyLoc — about 40% less than those with 3-4 visits and 20% less than those with at least 5 constrained visits. Intuitively, fewer constrained visits allow LLMs greater flexibility during generation. Therefore, *fewer constrained visits generally yield more realistic generated trajectories*.

Impact of Temperature. Temperature is an important hyperparameter in LLM sequence generation, controlling randomness: higher values increase diversity but risk incoherence, while lower values favor consistency but may lead to repetition. Figures 6 depicts the performance of Geo-Llama with temperature increasing from 0.7 to 1.6 for both uncontrolled

and controlled generation. Most metrics initially decrease to reach their lowest points around a temperature of 1.2 and then recover as the temperature continues to rise. This result indicates that *temperature significantly affects performance, the optimal temperature is approximately the same at 1.2 for uncontrolled and controlled generation with a good tradeoff between diversity and consistency.*
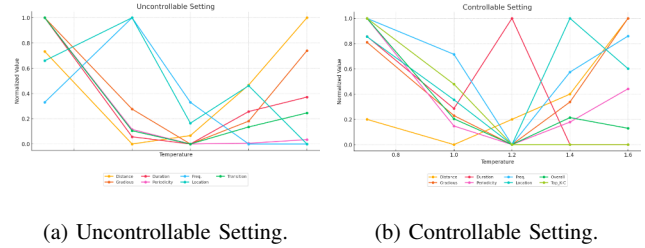

(a) Uncontrollable Setting.　　　(b) Controllable Setting.

Fig. 6: Impact of temperature on generation. The normalized values are averaged across two datasets and scaled to the [0,1] range.

VII. CONCLUSION

We have presented Geo-Llama, a novel framework that employs LLM fine-tuning to address the challenges of realistic trajectory generation with spatiotemporal constraints such as enforcing specific visits. By utilizing a visit-wise permutation strategy for trajectory fine-tuning, our method ensures the generation of high-quality, contextually coherent synthetic trajectories that strictly adhere to the constraints. Our approach not only outperforms existing approaches in capturing complicated mobility patterns and generating realistic trajectories in a more data-efficient way, but also overcomes the limitations of current controllable sequence generation techniques, which can only handle semantic-level and implicit constraints or a single explicit constraint.

In future work, we plan to extend Geo-Llama by incorporating auxiliary information, such as point of interest (POI) and modality features, and by utilizing continuous time and location representations. These enhancements aim to improve the modeling of more realistic and complex trajectories. In addition, we plan to explore domain transfer techniques to enable adaptation to unknown regions with limited data, investigate the impact of model size by comparing different LLMs, and develop prompt compression methods such as soft prompt tuning to reduce input length and computational cost.

REFERENCES

[1] T. Hu, S. Wang, B. She, M. Zhang, X. Huang, Y. Cui, J. Khuri, Y. Hu, X. Fu, X. Wang *et al.*, "Human mobility data in the covid-19 pandemic: characteristics, applications, and challenges," *International Journal of Digital Earth*, vol. 14, no. 9, pp. 1126–1147, 2021.

[2] R. A. Becker, R. Caceres, K. Hanson, J. M. Loh, S. Urbanek, A. Varshavsky, and C. Volinsky, "A tale of one city: Using cellular network data for urban planning," *IEEE Pervasive Computing*, vol. 10, no. 4, pp. 18–26, 2011.

[3] Y. Chen, C. Hu, and J. Wang, "Human-centered trajectory tracking control for autonomous vehicles with driver cut-in behavior prediction," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 9, pp. 8461–8471, 2019.

[4] M. e. a. Mokbel, "Mobility data science: Perspectives and challenges," *ACM Trans. Spatial Algorithms Syst.*, vol. 10, no. 2, jul 2024. [Online]. Available: https://doi.org/10.1145/3652158

[5] D.-T. Le, G. Cernicchiaro, C. Zegras, and J. Ferreira, "Constructing a synthetic population of establishments for the simmobility microsimulation platform," *Transportation Research Procedia*, vol. 19, pp. 81–93, 2016, transforming Urban Mobility. mobil.TUM 2016. International Scientific Conference on Mobility and Transport. Conference Proceedings. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352146516308560

[6] J. Feng, Z. Yang, F. Xu, H. Yu, M. Wang, and Y. Li, "Learning to simulate human mobility," in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 2020, pp. 3426–3433.

[7] H. Lin, S. Shaham, Y.-Y. Chiang, and C. Shahabi, "Generating realistic and representative trajectories with mobility behavior clustering," in *Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems*, 2023, pp. 1–4.

[8] M. Zhang, H. Lin, S. Takagi, Y. Cao, C. Shahabi, and L. Xiong, "Csgan: Modality-aware trajectory generation via clustering-based sequence gan," in *2023 24th IEEE International Conference on Mobile Data Management (MDM)*. IEEE, 2023, pp. 148–157.

[9] K. Ouyang, R. Shokri, D. S. Rosenblum, and W. Yang, "A nonparametric generative model for human trajectories." in *IJCAI*, vol. 18, 2018, pp. 3812–3817.

[10] Q. Long, H. Wang, T. Li, L. Huang, K. Wang, Q. Wu, G. Li, Y. Liang, L. Yu, and Y. Li, "Practical synthetic human trajectories generation based on variational point processes," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 4561–4571.

[11] Y. Zhu, J. J. Yu, X. Zhao, Q. Liu, Y. Ye, W. Chen, Z. Zhang, X. Wei, and Y. Liang, "Controltraj: Controllable trajectory generation with topology-constrained diffusion model," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 4676–4687.

[12] L. Zhang, L. Zhao, and D. Pfoser, "Factorized deep generative models for trajectory generation with spatiotemporal-validity constraints," *arXiv preprint arXiv:2009.09333*, 2020.

[13] S.-L. Hsu, E. Tung, J. Krumm, C. Shahabi, and K. Shafique, "Trajgpt: Controlled synthetic trajectory generation using a multitask transformer-based spatiotemporal model," in *Proceedings of the 32nd ACM International Conference on Advances in Geographic Information Systems*, 2024, pp. 362–371.

[14] H. Lin, J. Krumm, C. Shahabi, and L. Xiong, "Controllable visit trajectory generation with spatiotemporal constraints," in *Proceedings of the 2024 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2024.

[15] H. Zhang, H. Song, S. Li, M. Zhou, and D. Song, "A survey of controllable text generation using transformer-based pre-trained language models," *ACM Computing Surveys*, vol. 56, no. 3, pp. 1–37, 2023.

[16] N. Pelekis, C. Ntrigkogias, P. Tampakis, S. Sideridis, and Y. Theodoridis, "Hermoupolis: a trajectory generator for simulating generalized mobility patterns," in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part III 13*. Springer, 2013, pp. 659–662.

[17] S. Jiang, Y. Yang, S. Gupta, D. Veneziano, S. Athavale, and M. C. González, "The timegeo modeling framework for urban mobility without travel surveys," *Proceedings of the National Academy of Sciences*, vol. 113, no. 37, pp. E5370–E5378, 2016.

[18] C. Gao, X. Lan, N. Li, Y. Yuan, J. Ding, Z. Zhou, F. Xu, and Y. Li, "Large language models empowered agent-based modeling and simulation: A survey and perspectives," *Humanities and Social Sciences Communications*, vol. 11, no. 1, pp. 1–24, 2024.

[19] Q. Liu, S. Wu, L. Wang, and T. Tan, "Predicting the next location: A recurrent model with spatial and temporal contexts," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, no. 1, 2016.

[20] J. Feng, Y. Li, C. Zhang, F. Sun, F. Meng, A. Guo, and D. Jin, "Deepmove: Predicting human mobility with attentional recurrent networks," in *Proceedings of the 2018 world wide web conference*, 2018, pp. 1459–1468.

[21] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[22] L. Yu, W. Zhang, J. Wang, and Y. Yu, "Seqgan: Sequence generative adversarial nets with policy gradient," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 31, no. 1, 2017.

[23] Z. Zhou, X. Yang, R. Rossi, H. Zhao, and R. Yu, "Neural point process for learning spatiotemporal event dynamics," in *Learning for Dynamics and Control Conference*. PMLR, 2022, pp. 777–789.

[24] P. Bhandari, A. Anastasopoulos, and D. Pfoser, "Urban mobility assessment using llms," in *Proceedings of the 32nd ACM International Conference on Advances in Geographic Information Systems*, 2024, pp. 67–79.

[25] J. Wang, R. Jiang, C. Yang, Z. Wu, M. Onizuka, R. Shibasaki, N. Koshizuka, and C. Xiao, "Large language models as urban residents: An llm agent framework for personal mobility generation," *arXiv preprint arXiv:2402.14744*, 2024.

[26] M. P. Armstrong, G. Rushton, and D. L. Zimmerman, "Geographically masking health data to preserve confidentiality," *Statistics in medicine*, vol. 18, no. 5, pp. 497–525, 1999.

[27] P. A. Zandbergen, "Ensuring confidentiality of geocoded health data: Assessing geographic masking strategies for individual-level data," *Advances in medicine*. 2014, no. 1, p. 567049, 2014.

[28] X. Wang, X. Liu, Z. Lu, and H. Yang, "Large scale gps trajectory generation using map based on two stage gan," *Journal of Data Science*, vol. 19, no. 1, pp. 126–141, 2021.

[29] C. Cao and M. Li, "Generating mobility trajectories with retained data utility," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 2610–2620.

[30] Y. Zhu, Y. Ye, S. Zhang, X. Zhao, and J. Yu, "Difftraj: Generating gps trajectory with diffusion probabilistic model," *Advances in Neural Information Processing Systems*, vol. 36, pp. 65 168–65 188, 2023.

[31] S. Takagi, L. Xiong, F. Kato, Y. Cao, and M. Yoshikawa, "Hrnet: Differentially private hierarchical and multi-resolution network for human mobility data synthesization," *arXiv preprint arXiv:2405.08043*, 2024.

[32] H. Wang, Z. Zhang, T. Wang, S. He, M. Backes, J. Chen, and Y. Zhang, "{PrivTrace}: Differentially private trajectory synthesis by adaptive markov models," in *32nd USENIX Security Symposium (USENIX Security 23)*, 2023, pp. 1649–1666.

[33] S. Wang, Y. Du, X. Guo, B. Pan, Z. Qin, and L. Zhao, "Controllable data generation by deep learning: A review," *arXiv preprint arXiv:2207.09542*, 2022.

[34] C. Donahue, M. Lee, and P. Liang, "Enabling language models to fill in the blanks," *arXiv preprint arXiv:2005.05339*, 2020.

[35] C. Cheng, H. Yang, M. R. Lyu, and I. King, "Where you like to go next: Successive point-of-interest recommendation," in *Twenty-Third international joint conference on Artificial Intelligence*, 2013.

[36] D. Kong and F. Wu, "Hst-lstm: A hierarchical spatial-temporal long-short term memory network for location prediction." in *Ijcai*, vol. 18, no. 7, 2018, pp. 2341–2347.

[37] K. Sun, T. Qian, T. Chen, Y. Liang, Q. V. H. Nguyen, and H. Yin, "Where to go next: Modeling long-and short-term user preferences for point-of-interest recommendation," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 01, 2020, pp. 214–221.

[38] Y. Luo, Q. Liu, and Z. Liu, "Stan: Spatio-temporal attention network for next location recommendation," in *Proceedings of the web conference 2021*, 2021, pp. 2177–2185.

[39] L. Zhang, Z. Sun, Z. Wu, J. Zhang, Y. S. Ong, and X. Qu, "Next point-of-interest recommendation with inferring multi-step future preferences." in *IJCAI*, 2022, pp. 3751–3757.

[40] N. Lim, B. Hooi, S.-K. Ng, X. Wang, Y. L. Goh, R. Weng, and J. Varadarajan, "Stp-udgat: Spatial-temporal-preference user dimensional graph attention network for next poi recommendation," in *Proceedings of the 29th ACM International conference on information & knowledge management*, 2020, pp. 845–854.

[41] S. Yang, J. Liu, and K. Zhao, "Getnext: trajectory flow map enhanced transformer for next poi recommendation," in *Proceedings of the 45th International ACM SIGIR Conference on research and development in information retrieval*, 2022, pp. 1144–1153.

[42] P. Li, M. de Rijke, H. Xue, S. Ao, Y. Song, and F. D. Salim, "Large language models for next point-of-interest recommendation," in *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2024, pp. 1463–1472.

[43] H. Naveed, A. U. Khan, S. Qiu, M. Saqib, S. Anwar, M. Usman, N. Akhtar, N. Barnes, and A. Mian, "A comprehensive overview of large language models," 2024. [Online]. Available: https://arxiv.org/abs/2307.06435

[44] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[45] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[46] Y. Zheng, X. Xie, W.-Y. Ma *et al.*, "Geolife: A collaborative social networking service among user, location and trajectory." *IEEE Data Eng. Bull.*, vol. 33, no. 2, pp. 32–39, 2010.

[47] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[48] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[49] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[50] D. Huang, X. Song, Z. Fan, R. Jiang, R. Shibasaki, Y. Zhang, H. Wang, and Y. Kato, "A variational autoencoder based generative model of urban human mobility," in *2019 IEEE conference on multimedia information processing and retrieval (MIPR)*. IEEE, 2019, pp. 425–430.