# RHYTHM: Reasoning with Hierarchical Temporal Tokenization for Human Mobility


**Haoyu He*♮ Haozheng Luo*† Yan Chen† Ryan Qi Wang ♮**

♮Northeastern University, Boston, MA 02115 USA; †Northwestern University, Evanston, IL 60208 USA


## Introduction

**Background**:

- **Periodicity gap**: Traditional models miss long-term patterns (93% of trajectories are predictable) due to vanishing gradients or static time treatment
- **Reasoning gap**: Mobility-specific models fail to leverage LLMs' proven capabilities in complex spatio-temporal reasoning

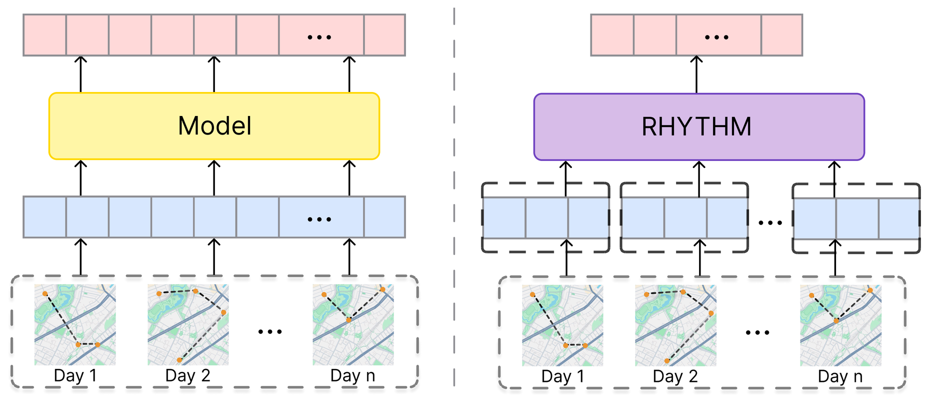



Figure 1: **Motivation for RHYTHM.** Instead of processing entire trajectories as a continuous sequence, RHYTHM segments trajectories into tokens to better capture periodic patterns.

**Contribution.** We propose **RHYTHM** (Reasoning with Hierarchical Temporal Tokenization for Human Mobility)

- Temporal tokenization captures multi-scale periodicity via hierarchical attention
- Prompt-guided approach enhances semantic pattern understanding
- 87.6% parameters frozen $\rightarrow$ 24.6% computational savings

## Problem Definition

Given:

- Historical trajectory: $\mathcal{X} = \{x_1, x_2, \ldots, x_T\}$ where $x_i = (t_i, l_i)$
  - $t_i$: timestamp
  - $l_i \in \mathcal{L}$: location from finite set $\mathcal{L}$
- Future timestamps $\mathcal{T} = \{t_{T+1}, t_{T+2}, \ldots, t_{T+H}\}$
  - $H$: prediction horizon

**Objective**: Predict future locations $\mathcal{Y} = \{l_{T+1}, l_{T+2}, \ldots, l_{T+H}\}$

**Goal**: Learn mapping $f : (\mathcal{X}, \mathcal{T}) \mapsto \mathcal{Y}$

## Model

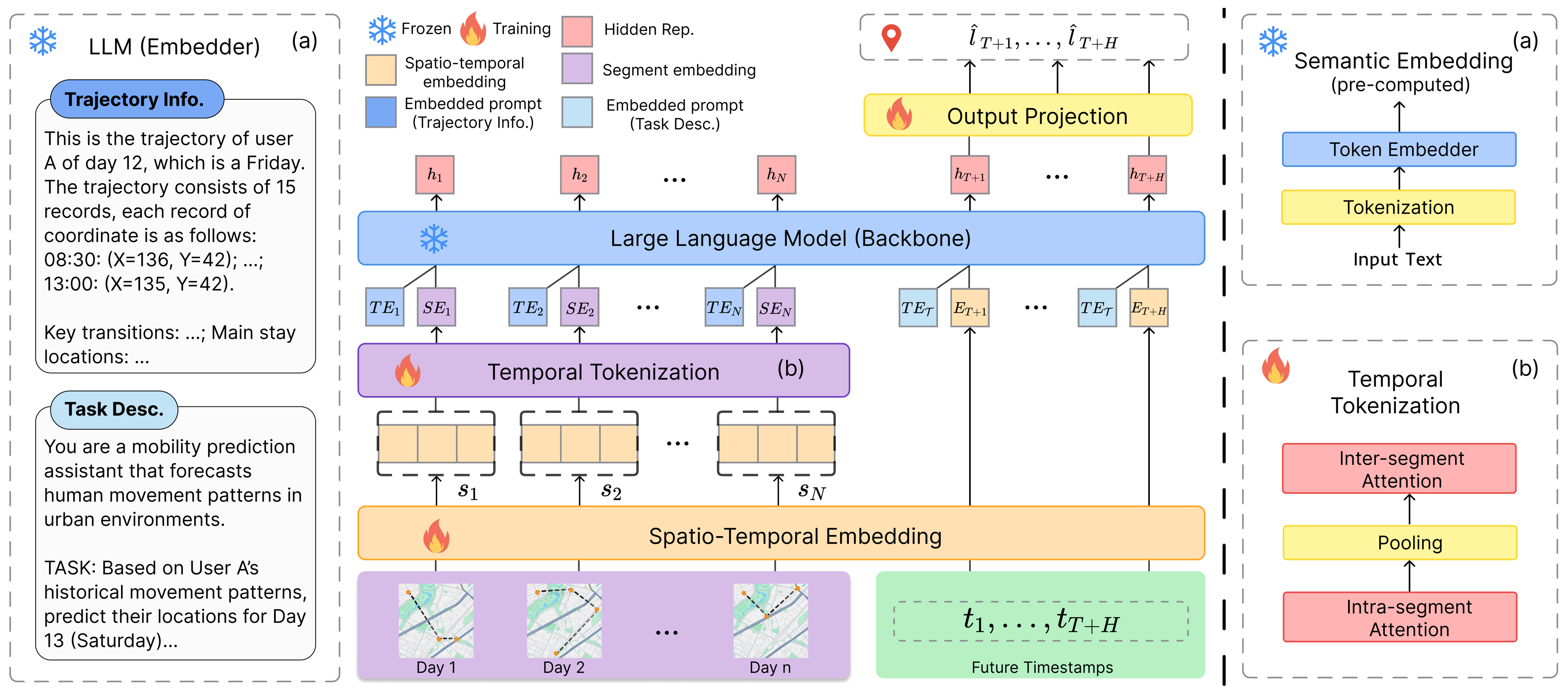



Figure 2: **The proposed architecture of RHYTHM.** Our framework processes historical trajectories through spatio-temporal embedding and temporal tokenization (**b**), capturing local and global dependencies via hierarchical attention. Segment representations are enriched with semantic embeddings from trajectory information, while future timestamps incorporate task description context (**a**). This combined sequence passes through a frozen LLM backbone with output projection to generate location predictions.

**Structure**:

- **Temporal Tokenization**: Capture multi-scale patterns by reducing $T \rightarrow N$ segments

$$\mathbf{s}_i = \{E_{(i-1)L+1}, \ldots, E_{iL}\} \quad \text{(partition)}$$

$$\widetilde{\mathbf{E}}^{(i)} = \mathrm{Attention}(\mathbf{s}_i) \quad \text{(local patterns)}$$

$$\mathbf{SE}_i = \mathrm{Pool}(\widetilde{\mathbf{E}}^{(i)}) \quad \text{(compress)}$$

$$\widetilde{\mathbf{SE}}_{1:N} = \mathrm{Attention}(\mathbf{SE}_{1:N}) \quad \text{(global patterns)}$$

- **Semantic Embedding**: Enrich tokens with trajectory context using frozen LLM

$$\mathbf{TE}_i = \mathrm{SelectLast}(\mathrm{LLM}(\mathrm{Prompt}(s_i)))$$

$$\mathbf{CE}_i = \widetilde{\mathbf{SE}}_i + \mathbf{TE}_i$$

- **LLM Reasoning**: Leverage pretrained backbone for location prediction

$$h_i = \mathrm{LLM}(\mathbf{CE}_i)$$

$$P(l_{T+j} | \mathcal{X}, \mathcal{T}) = \mathrm{softmax}(W_o h_{T+j} + b_o)$$

**Efficiency Gains**:

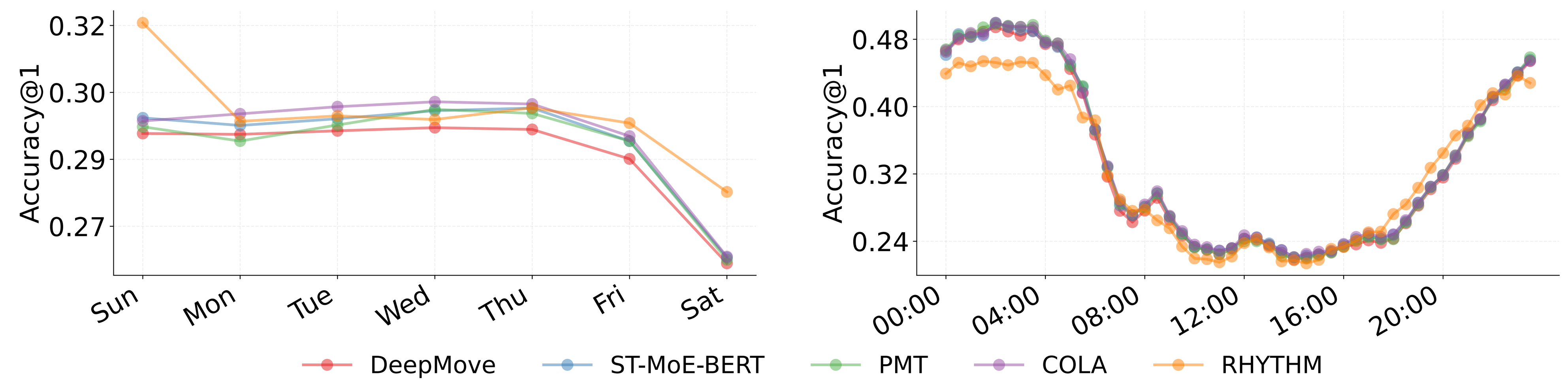
- Offline semantic embedding computation—no runtime LLM inference
- Attention complexity: $\mathcal{O}((T+H)^2) \rightarrow \mathcal{O}((N+H)^2)$ where $N \ll T$
- Frozen LLM backbone: only 12.37% trainable parameters

## Experiments

Table 1: **Performance of RHYTHM and baselines on the Kumamoto, Sapporo, and Hiroshima datasets.** The evaluation metrics include Accuracy@k for different values of k, with variance $\leq 2\%$. The best results are highlighted in **bold**, and the second-best results are underlined. RHYTHM demonstrates superior performance compared to baselines across most configurations.

| | Kumamoto | | | Sapporo | | | Hiroshima | | |
|---|---|---|---|---|---|---|---|---|---|
| **Model** | Acc@1 | Acc@3 | Acc@5 | Acc@1 | Acc@3 | Acc@5 | Acc@1 | Acc@3 | Acc@5 |
| LSTM | 0.2652 | 0.4799 | 0.5472 | 0.2310 | 0.3940 | 0.4526 | 0.2129 | 0.3775 | 0.4415 |
| DeepMove | 0.2779 | 0.4986 | 0.5683 | 0.2825 | 0.4672 | 0.5264 | 0.2804 | 0.4810 | 0.5477 |
| PatchTST | 0.2751 | 0.5018 | 0.5716 | 0.2703 | 0.4582 | 0.5168 | 0.2752 | 0.4839 | 0.5522 |
| iTransformer | 0.2609 | 0.4724 | 0.5412 | 0.2696 | 0.4500 | 0.5070 | 0.2804 | 0.4857 | 0.5523 |
| TimeLLM | 0.2712 | 0.4848 | 0.5535 | 0.2792 | 0.4746 | 0.5352 | 0.2698 | 0.4753 | 0.5426 |
| CMHSA | 0.2862 | 0.5182 | 0.5887 | 0.2890 | **0.4901** | <u>0.5525</u> | 0.2874 | 0.5001 | 0.5684 |
| PMT | 0.2697 | 0.4475 | 0.5187 | 0.2878 | <u>0.4896</u> | 0.5522 | 0.2850 | 0.4982 | 0.5668 |
| COLA | 0.2864 | 0.5186 | 0.5896 | 0.2847 | 0.4865 | 0.5497 | 0.2874 | 0.5013 | 0.5708 |
| ST-MoE-BERT | 0.2862 | 0.5155 | 0.5871 | 0.2869 | 0.4856 | 0.5480 | 0.2839 | 0.4925 | 0.5601 |
| Mobility-LLM | 0.2666 | 0.4793 | 0.5448 | 0.2838 | 0.4703 | 0.5288 | 0.2826 | 0.4856 | 0.5525 |
| RHYTHM-LLaMA-1B | <u>0.2929</u> | <u>0.5200</u> | 0.5835 | 0.2931 | 0.4876 | 0.5502 | 0.2913 | 0.5027 | 0.5753 |
| RHYTHM-Gemma-2B | 0.2923 | 0.5191 | <u>0.5932</u> | **0.2943** | 0.4896 | **0.5545** | **0.2953** | **0.5074** | **0.5798** |
| RHYTHM-LLaMA-3B | **0.2941** | **0.5205** | **0.5947** | <u>0.2938</u> | 0.4875 | 0.5523 | <u>0.2929</u> | <u>0.5032</u> | <u>0.5756</u> |

**Overall Performance**: RHYTHM outperforms all baselines with **2.4%** higher prediction accuracy across most metrics.




**Temporal Analysis**: Achieves **5.0%** improvement on challenging scenarios (weekends & peak hours) where baselines struggle.

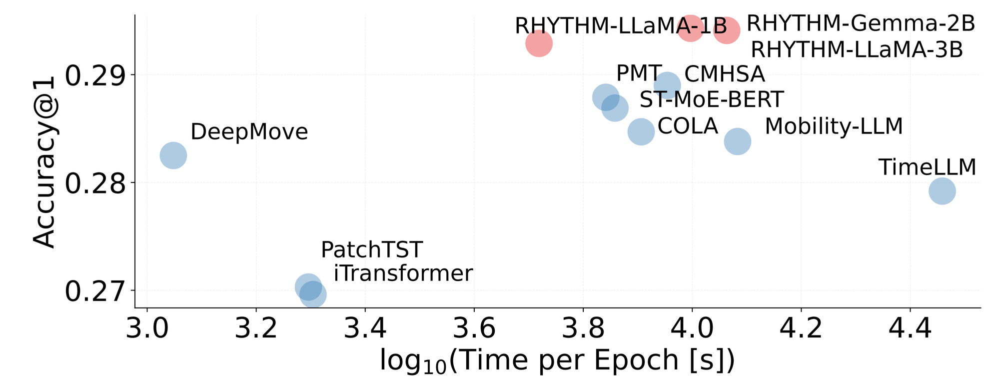



Figure 3: **Training Speed vs. performance of RHYTHM and baseline models on the Sapporo Dataset.**

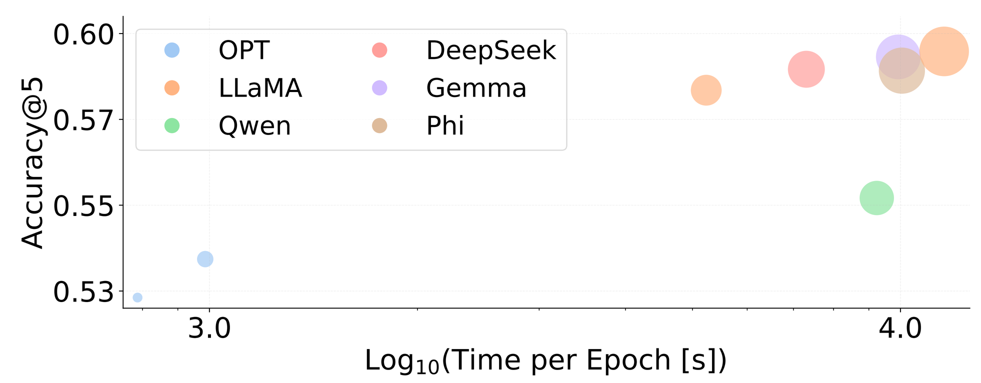



Figure 4: **Efficiency comparison of alternative LLMs, evaluated by the same configuration of Table 4.**

**Efficiency**: Reduces training time by **24.6%** compared to best baseline while maintaining superior performance.

**Scalability**: Performance scales predictably with model size, following established scaling laws.

## Contact


**Haoyu He**

he.haoyu1@northeastern.edu

PhD Candidate, Northeastern University