Learning Generalized and Flexible Trajectory Models from Omni-Semantic Supervision

Yuanshao Zhu*
Southern University of Science and Technology, City University of Hong Kong, The Hong Kong University of Science and Technology (Guangzhou)
Shenzhen, China
yuanshao@ieee.org

James Jianqiao Yu†
Harbin Institute of Technology, Shenzhen
Shenzhen, China
jqyu@ieee.org

Xiangyu Zhao†
City University of Hong Kong
Hong Kong, China
xianzhao@cityu.edu.hk

Xiao Han
Zhejiang University of Technology
Hangzhou, China
hahahenha@gmail.com

Qidong Liu
Xi'an Jiaotong University, City University of Hong Kong
Xi'an, China
liuqidong@stu.xjtu.edu.cn

Xuetao Wei
Southern University of Science and Technology
Shenzhen, China
weixt@sustech.edu.cn

Yuxuan Liang†
The Hong Kong University of Science and Technology (Guangzhou)
Guangzhou, China
yuxliang@outlook.com

Abstract

The widespread adoption of mobile devices and data collection technologies has led to an exponential increase in trajectory data, presenting significant challenges in spatio-temporal data mining, particularly for efficient and accurate trajectory retrieval. However, existing methods for trajectory retrieval face notable limitations, including inefficiencies in large-scale data, lack of support for condition-based queries, and reliance on trajectory similarity measures. To address the above challenges, we propose **OmniTraj**, a generalized and flexible omni-semantic trajectory retrieval framework that integrates four complementary modalities or semantics—raw trajectories, topology, road segments, and regions—into a unified system. Unlike traditional approaches that are limited to computing and processing trajectories as a single modality, OmniTraj designs dedicated encoders for each modality, which are embedded and fused into a shared representation space. This design enables OmniTraj to support accurate and flexible queries based on any individual modality or combination thereof, overcoming the rigidity of traditional similarity-based methods. Extensive experiments on two real-world datasets demonstrate the effectiveness of OmniTraj in handling large-scale data, providing flexible, multi-modality queries, and supporting downstream tasks and applications.

CCS Concepts

• **Information systems** → **Location based services**; **Global positioning systems**; *Novelty in information retrieval.*

Keywords

GPS Trajectory Analysis; Multi-Modality Retrieval; Spatio-Temporal Data Mining

ACM Reference Format:
Yuanshao Zhu, James Jianqiao Yu†, Xiangyu Zhao†, Xiao Han, Qidong Liu, Xuetao Wei, and Yuxuan Liang. 2025. Learning Generalized and Flexible Trajectory Models from Omni-Semantic Supervision . In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '25), August 3–7, 2025, Toronto, ON, Canada.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3711896.3737019

1 Introduction

The widespread availability of GPS-enabled devices and urban sensing infrastructure has resulted in the generation of vast amounts of spatio-temporal trajectory data, capturing the movement patterns of vehicles, pedestrians, and public urban services [8, 11, 32, 44]. As a fundamental building block for smart city applications, trajectory retrieval enables semantically rich insight into mobility behaviors, ranging from traffic congestion prediction and emergency route planning to location-based recommendations [5, 16, 28]. For instance, urban planners might analyze trajectories passing through specific road segments to optimize traffic light scheduling, while

*Work done during the internship at HKUST(GZ)
†Corresponding authors

KDD '25, August 3–7, 2025, Toronto, ON, Canada

ACM ISBN 979-8-4007-1454-2/2025/08
https://doi.org/10.1145/3711896.3737019

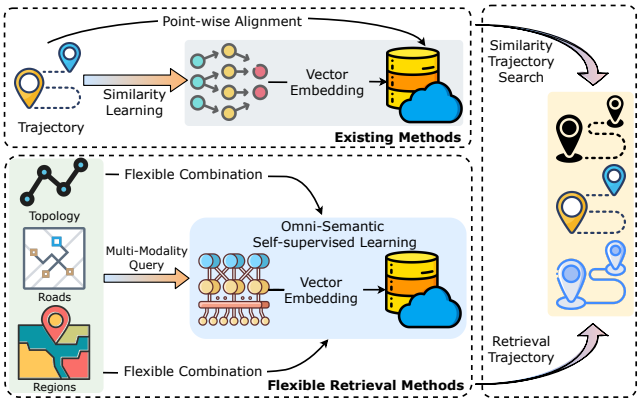


Figure 1: Comparison of trajectory retrieval methods. Our method can retrieve trajectories according to their different modalities and provides flexible retrieval solutions.

logistics companies could query delivery routes that intersect designated regions to evaluate operational efficiency [12, 13, 18, 29].

Existing approaches to trajectory retrieval focusing on trajectory similarity measures that can be broadly categorized into heuristic methods and deep learning-based methods, face critical limitations in handling *complex queries*. Heuristic methods, such as Dynamic Time Warping (DTW) [25] and Hausdorff Distance [1], focus on computing the similarity between full trajectories based on point-wise alignment. While effective for whole-trajectory matching, they suffer from prohibitive computational costs (e.g., $O(n^2)$) time complexity for DTW), especially when applied to large-scale datasets and are limited to supporting only full-trajectory similarity queries. Deep learning-based methods—including t2vec [20] and TrajCL [4]—address efficiency issues by encoding trajectories into compact embeddings ($O(n)$ time complexity when retrieving). However, these methods rely on holistic similarity computation and still do not address the support of feature-specific conditions like filtering trajectories that traverse designated road segments. Recent multimodal trajectory studies [21, 22] have attempted to integrate ancillary data (e.g., road networks or POIs), but have primarily targeted specific classification or prediction tasks rather than cross-modal retrieval. For example, models inspired by contrastive learning [35, 42] align trajectories with other descriptions, but lack a mechanism for querying trajectories based on spatial semantic combinations. Overall, existing methods suffer from inefficiency in handling large-scale data, tunnel vision on trajectory similarity tasks, and unavailability to support conditional (or modality-based) queries. These restrictions hinder their applicability to real-world scenarios, where queries are inherently multi-faceted and demand both precision and efficiency at city scales.

To address these issues, we propose to integrate diverse trajectory modalities, such as trajectory, topology, road segments, and regions, into a unified framework, enabling flexible and modality-supported queries. As shown in Figure 1, existing methods are limited to matching trajectories as a single modality for trajectory point-wise alignment or using deep learning models for similarity learning. Meanwhile, a multi-modality trajectory query needs to construct full semantic self-supervised models for multiple modalities to realize flexible retrieval based on user-specified conditions.

Table 1: Comparison of OmniTraj with representative trajectory retrieval/similarity methods.

Methods	DTW	t2vec	TrajCL	OmniTraj
Retrieval Complexity	$O(n^2)$	$O(n)$	$O(n)$	$O(n)$
Scalability	✗	✓	✓	✓
Conditional Query	✗	✗	✗	✓
Multi-Modality Support	✗	✗	✗	✓
Application Support	✗	✗	✗	✓

However, two fundamental challenges emerge in developing such a multi-modality retrieval model: (1) **Semantic Disentanglement**. How to represent trajectories in a modality-specific embedding space (e.g., road segments vs. regions) while maintaining their interrelationships? (2) **Flexibility and Efficiency**. How can multiple modalities during retrieval dynamically combine without sacrificing query efficiency while ensuring flexibility in supporting a wide range of complex queries? For example, DTW-based methods may retrieve trajectories matching a road segment but ignoring regional constraints, while deep learning models fail to isolate modality-specific features, yielding irrelevant results.

To bridge existing research gaps and address the challenges, we propose **OmniTraj**, a generalized and flexible omni-semantic trajectory retrieval framework designed to integrate diverse trajectory aspects (e.g., topology, road segments, and regions) into a unified system. Specifically, the primary innovation of the OmniTraj is its ability to handle both spatial and semantic complementary constraints and provide a richer understanding of trajectory data. Firstly, OmniTraj encodes the unique features of each trajectory modality individually, ensuring that the precise description of each modality is preserved during the retrieval process and supports a wide range of applications. The framework then employs a modular design while preserving their interrelationships without sacrificing scalability or query efficiency. By decoupling the encoding process of each modality and dynamically combining them during retrieval, OmniTraj enables flexible and efficient queries. In addition, the encoders for each modality can be used individually to support a wide range of downstream tasks. As highlighted in Table 1, OmniTraj outperforms traditional methods by flexible and scalable design, and supporting condition-based queries while maintaining high efficiency on large-scale datasets. As a result, OmniTraj addresses the limitations of traditional trajectory similarity measures and provides a powerful solution that combines accuracy, efficiency, and versatility in trajectory retrieval. By enabling the handling of complex queries based on a combination of trajectory, spatial, and semantic information, OmniTraj sets a new perspective for the trajectory retrieval task.

In summary, the contributions of our research are as follows:

- We formalize the novel task of multimodal trajectory retrieval, enabling condition-based queries that span diverse trajectory semantics. This approach overcomes the limitations of current trajectory similarity measures and provides a fresh insight into how trajectory retrieval tasks can be approached.
- We present OmniTraj, the first multimodal trajectory retrieval framework designed to process multiple trajectory modalities. This framework offers a generalized solution to trajectory retrieval by incorporating modality-specific encoders, enabling

flexible, scalable, and efficient querying across a wide range of trajectory components.
- Through extensive experiments, we demonstrate the effectiveness of OmniTraj, showcasing its ability to accurately retrieve trajectories under diverse query conditions. In addition, the framework is scalable, supports a variety of downstream tasks, and demonstrates strong generalization capabilities.

2 Preliminary

We formally define the multimodal trajectory retrieval task by introducing four key trajectory semantics, each providing distinct perspectives: trajectory, topology, road segments, and regions.

Definition 1 (Trajectory). A trajectory $\boldsymbol{T}$ is a temporally ordered sequence of spatio-temporal points, defined as: $\boldsymbol{T} = \{p_1, p_2, \ldots, p_n\}$, where $p_i = (x_i, y_i)$ represents the spatial coordinates (e.g., latitude and longitude) of the object (e.g., vehicle or pedestrian). In real-world application, a trajectory can be abstracted into multiple granular representations—including *topology*, *road segments*, and *regions*—to capture diverse semantic perspectives.

Definition 2 (Topology). The topology $\boldsymbol{T}^{(\text{top})}$ of a trajectory $\boldsymbol{T}$ refers to the subsequence of critical points along the trajectory. Formally, $\boldsymbol{T}^{(\text{top})} = \{tp_1, tp_2, \ldots, tp_k\}$, $(k \leq n)$, where each $tp_i \in \{p_1, p_2, \ldots, p_n\}$ indicates a significant change in movement (e.g., abrupt turns, marked velocity variations, or crossings at predefined landmarks). This semantic emphasizes the spatial geometry of the trajectory while omitting temporal details.

Definition 3 (Road Segments). For a trajectory that has been map-matched to a road network, its road segment representation is given by: $\boldsymbol{T}^{(\text{road})} = \{r_1, r_2, \ldots, r_j\}$, where each r_i is a unique identifier corresponding to a specific road segment (i.e., a continuous stretch of road between two intersections). This semantic captures the sequence of roads traversed by the trajectory.

Definition 4 (Regions). A region is a spatially bounded area with homogeneous properties, such as administrative boundaries, clusters of points of interest (POIs), or urban districts. The region semantic of a trajectory is defined as: $\boldsymbol{T}^{(\text{reg})} = \{\mathcal{R}_1, \mathcal{R}_2, \ldots\}$, where the condition $\boldsymbol{T} \cap \mathcal{R}_i \neq \emptyset$ indicates that at least one point of $\boldsymbol{T}$ lies within the region $\mathcal{R}_i$.

Problem Statement (Trajectory Retrieval). Let $\mathcal{D} = \{\boldsymbol{T}_1, \ldots, \boldsymbol{T}_N\}$ be a database of trajectories. Trajectory retrieval aims to efficiently query and retrieve relevant trajectories from $\mathcal{D}$ based on one or more feature modalities (e.g., topology, road segments, regions, or their combination). Formally, let T_q denote a query trajectory or query condition, which can be expressed using multiple modalities or combinations thereof. The retrieval function is defined as:

$$\mathcal{F}(\boldsymbol{T}_q) = \arg\max_{\boldsymbol{T}_i \in \mathcal{D}} \text{similarity}(\boldsymbol{T}_q, \boldsymbol{T}_i), \tag{1}$$

where similarity$(\cdot, \cdot)$ measures the degree of alignment or resemblance between the query $\boldsymbol{T}_q$ and a candidate trajectory $\boldsymbol{T}_i$ from the database. This similarity metric is designed to be highly versatile, potentially leveraging the information encoded in multiple modalities to provide a comprehensive comparison. *In practice, a robust trajectory retrieval framework should achieve high retrieval accuracy, offer flexibility for diverse query modalities, and efficiently manage large-scale databases with acceptable times.*

3 Methodology

3.1 Overview of the OmniTraj Framework

To overcome the limitations of existing trajectory similarity computation methods and supporting multi-modality retrieval tasks, we propose OmniTraj, a novel omni-semantic trajectory retrieval framework that integrates four complementary modalities—raw trajectories, topology, road segments, and regions—into a unified system for flexible and efficient querying. As depicted in Figure 2, OmniTraj is built upon four dedicated encoders, each meticulously designed to capture a distinct semantic aspect of trajectory data: The *Trajectory Encoder* processes the raw spatio-temporal sequence of each trajectory to generate latent representations that encapsulate movement dynamics and temporal evolution. The *Topology Encoder* identifies and embeds critical spatial landmarks (e.g., abrupt turns, speed changes) to characterize the geometric structure. The *Road Encoder* maps trajectories onto road network segments and learns road-aware embeddings through neural network architectures, effectively capturing connectivity and routing information. The *Region Encoder* detects and encodes the spatial-semantic properties of regions traversed by trajectories, thereby enriching the overall representation with contextual insights. This modular design provides three principal advantages:

- **Flexible Query Support**: OmniTraj empowers users to formulate queries using any combination of modalities. For instance, one may retrieve trajectories that pass through a specific road segment (e.g., Road 1) and intersect a particular region (e.g., Region B), enabling precise adaptation to diverse real-world scenarios.
- **Rich MultiModal Representations**: Each encoder generates modality-specific embeddings that preserve the unique semantic characteristics of its respective modality. A contrastive learning strategy is then employed to align these embeddings within a shared latent space, ensuring cross-modal compatibility.
- **Scalability and Efficiency**: The decoupled encoder architecture allows parallel processing of modalities, making the framework highly scalable and capable of efficiently handling large-scale trajectory databases. Additionally, when a modality-based query is required, it can be directly and smoothly integrated into existing frameworks without re-engineering.

3.2 Modality-Specific Encoders

In this section, we detail the construction of each encoder and describe how our design implements an efficient and flexible multimodal trajectory retrieval mechanism. More details about the structure of these encoders can be found in **Appendix A**.

3.2.1 ***Trajectory Encoder.*** Learning robust representations from raw spatio-temporal trajectories presents two key challenges: (1) capturing both local motion patterns and long-range spatial dependencies, and (2) accommodating variable sequence lengths while minimizing computational complexity for efficiency. To tackle these challenges, we adopt a transformer-based architecture inspired by the Vision Transformer [14] to model the dynamic patterns.

Given a trajectory $\boldsymbol{T}$, we first normalize its coordinates to a local reference frame and up/down-sample to a fixed length L using cubic spline interpolation, ensuring consistency across different scales or coordinate systems. Then the resampled trajectory $\hat{\boldsymbol{T}}$ is

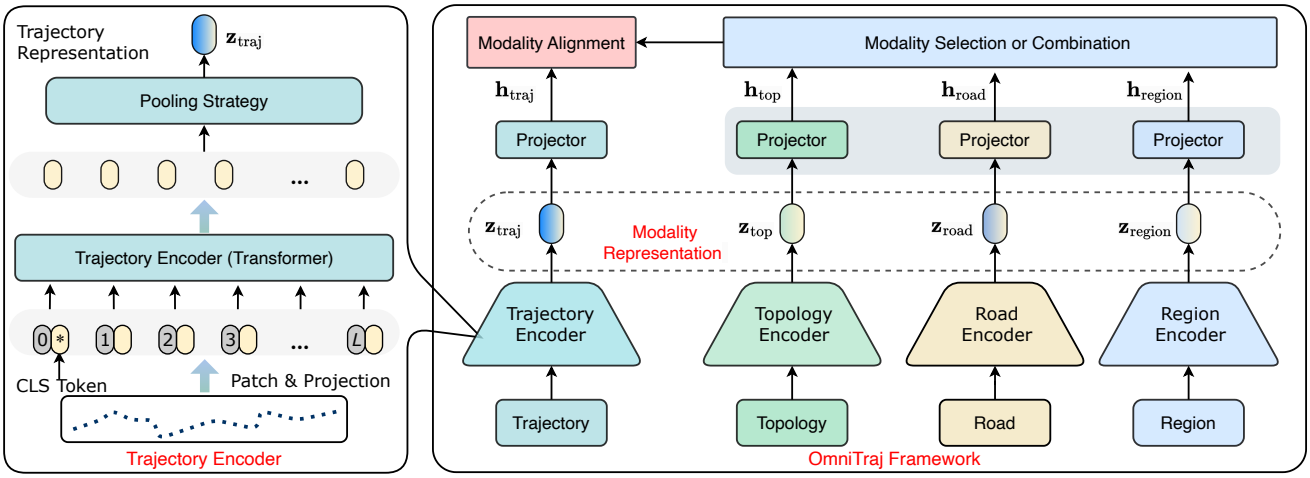


Figure 2: Overview of the proposed OmniTraj framework, which contains four individual modal encoders. Each encoder captures a diverse semantic representation of a trajectory and projects it into a shared space.

divided into no-overlapping patches, where the number of patches is $N_p = \lfloor L/P \rfloor$ and P is the patch size. Each patch, denoted as $\boldsymbol{z}_p$, is computed by applying a learnable projection matrix $\boldsymbol{W}_p$ that maps the flattened patch coordinates into a d-dimensional latent space:

$$\boldsymbol{z}_p = \boldsymbol{W}_p \cdot \hat{\boldsymbol{T}} \in \mathbb{R}^{N_p \times d}, \tag{2}$$

To maintain the sequential order of the trajectory, we introduce learnable positional embeddings $\boldsymbol{E}_{\text{pos}} \in \mathbb{R}^{(N_p+1)\times d}$ which are added to each patch embedding. Additionally, a special [CLS] token $\boldsymbol{z}_{\text{cls}}$ is prepended to the sequence to aggregate global trajectory features. The full sequence $\boldsymbol{z} = [\boldsymbol{z}_{\text{cls}}; \boldsymbol{z}_p] + \boldsymbol{E}_{\text{pos}}$ is then processed by N stacked transformer blocks. Each block computes multi-head self-attention over the spatial neighbors of each patch, allowing the model to learn both local motion patterns and long-range dependencies. Finally, the trajectory embedding is derived from the [CLS] token or by pooling all patch embeddings. The trajectory representation is thus computed as:

$$\boldsymbol{z}_{\text{traj}} = \textbf{TrajEncoder}(T) \in \mathbb{R}^d. \tag{3}$$

This encoder efficiently learns trajectory representations by capturing local patterns via patches, while also leveraging the self-attention mechanism to model long-range dependencies. By using patches and projection, we reduce the computational complexity from the traditional $O(L^2)$ to $O((L/P)^2)$, significantly improving the efficiency of trajectory representation computation. The resulting fixed-length trajectory embedding is well-suited for subsequent retrieval tasks, whether based on similarity or condition-based queries, offering both accuracy and computational efficiency solutions in large-scale settings.

3.2.2 ***Topology Encoder.*** The main challenge in building a topology encoder is to capture the overall geometric semantics of the trajectory with only a limited number of key points. This means that the encoder must efficiently process each point and derive a global spatial understanding from this sparse representation. To achieve this, given a topological sequence $\boldsymbol{T}^{(\text{top})}$ with k points, the topology encoder first embeds each point $tp_i \in \mathbb{R}^2$ into a high-dimensional vector $\{\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_k\}$ using a learned embedding layer. To preserve the sequential order and capture the inherent spatial structure of these points, we incorporate Rotary Positional Embedding (RoPE) [30]. RoPE injects relative positional information by applying a position-dependent rotation to each embedding, thereby allowing the model to naturally learn the spatial dependencies between points. Specifically, for each embedding $\boldsymbol{e}_i$, the rotation angle is $\theta_i = i/10000^{2k/d}$, and the embedding is split into two halves, $\boldsymbol{e}_i^{(1)}$ and $\boldsymbol{e}_i^{(2)}$. The RoPE mechanism then rotates these splits as follows:

$$\text{RoPE}(\boldsymbol{e}_i) = \begin{pmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{pmatrix} \begin{pmatrix} \boldsymbol{e}_i^{(1)} \\ \boldsymbol{e}_i^{(2)} \end{pmatrix}. \tag{4}$$

After positional encoding, the sequence of RoPE-enhanced embeddings is processed by N transformer layers, which leverage self-attention mechanisms to model the interactions among sparse topological points. A special [CLS] token is prepended to the sequence to aggregate global geometric features. The final topological embedding is then obtained as:

$$\boldsymbol{z}_{\text{top}} = \textbf{TopologyEncoder}(\boldsymbol{T}^{(\text{top})}) \in \mathbb{R}^d. \tag{5}$$

By leveraging RoPE, our topology encoder efficiently models sparse geometric structures and preserves critical spatial dependencies without the need for dense, point-wise embeddings.

3.2.3 ***Road Encoder.*** The road encoder is designed to model the semantic and structural relationships between road segments in a map-matched trajectory, enabling flexible query conditions that go beyond exact matching. Considering that our approach is not limited to precise road segment sequences, it should also support queries such as retrieving trajectories that pass through or cover a certain road segment sequence. To facilitate this, we incorporate a series of augmentation techniques during training, which help the model learn robust and invariant representations. These augmentations include:

- Reversing: Flip the order of road segments to simulate bidirectional traversal.
- Discarding/Truncating: Randomly removing or retaining only part of the segments to mimic incomplete queries.
- Shuffling: Perturbing the order of segments within local windows to encourage the learning of invariant features.
- Random Replacement: Substituting some portions of segments with others to enhance generalization.

Considering that each road segment has a unique identifier, we first project each segment r_i into a d-dimensional space using a learnable lookup table:

$$z_i = \boldsymbol{W}_{\text{road}}[r_i] \in \mathbb{R}^d, \tag{6}$$

where $\boldsymbol{W}_{\text{road}} \in \mathbb{R}^{r \times d}$ is the road embedding matrix and $|r|$ denotes the total number of unique road segments. Next, we feed it into the N transformer block enhanced with RoPE for processing to obtain the final road segment embedding:

$$z_{\text{road}} = \textbf{RoadEncoder}(\boldsymbol{T}^{(\text{road})}) \in \mathbb{R}^d. \tag{7}$$

By integrating these augmentation techniques and advanced encoding strategies, the road encoder not only captures detailed structural information from the road network but also learns flexible semantic representations. This robustness enables the system to support a variety of condition-based queries and enhances its generalization to real-world retrieval scenarios, where queries might be incomplete, unordered, or require a degree of invariance.

3.2.4 ***Region Encoder.*** Our motivation for constructing the region encoder is to model the spatial relationships and semantic features of the regions through which the trajectories pass, thereby providing robust and flexible representations of the paths of the trajectories in different spatial regions. Similar to the road encoder, the region encoder supports condition-based queries by learning invariant and generalizable representations from region identifiers. Specifically, each region identifier $\mathcal{R}_i$ is first projected into a high-dimensional embedding $\boldsymbol{e}_i \in \mathbb{R}^d$ via a learnable lookup table, yielding a sequence of region embeddings $\{\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_m\}$. Unlike the road encoder, which emphasizes relative positional information, the region encoder processes this sequence of embeddings using a series of transformer blocks without explicitly incorporating positional information. The self-attention mechanism in the transformer naturally captures both local and global relationships among the regions, enabling the model to learn complex spatial dependencies. To further enhance generalization and robustness, especially under conditions where region data may be incomplete or unordered, we apply augmentation techniques such as random shuffling and random removal of regions during training. These augmentations force the encoder to focus on the invariant semantic properties of the regions rather than their precise order. Finally, the aggregated region trajectory embedding, denoted as:

$$z_{\text{region}} = \textbf{RegionEncoder}(\boldsymbol{T}^{(\text{region})}) \in \mathbb{R}^d, \tag{8}$$

which can be obtained either by extracting the [CLS] token or by applying pooling over the sequence of region embeddings. This flexible aggregation strategy supports efficient querying based on region-specific conditions and ensures that the final representation effectively captures the semantic context of the trajectory.

3.3 Omni-Semantic Supervision

After obtaining the modality-specific embeddings $z_{\text{mod}} \in \mathbb{R}^d$ and $z_{\text{mod}} \in \{z_{\text{traj}}, z_{\text{top}}, z_{\text{road}}, z_{\text{region}}\}$ from the trajectory, topology, road, and region encoders respectively, the next critical step in OmniTraj is to align these representations to supports flexible and efficient querying. To achieve this, we designed a projection head employing a two-layer linear neural network for each modal representation. This transformation is expressed as: $\boldsymbol{h}_{\text{mod}} = \boldsymbol{W}_{\text{mod}} \cdot z_{\text{mod}}$, where $\boldsymbol{W}_{\text{mod}} \in \mathbb{R}^{h \times d}$ is learnable projection matrix. This transformation ensures that the resulting embeddings $\{\boldsymbol{h}_{\text{traj}}, \boldsymbol{h}_{\text{top}}, \boldsymbol{h}_{\text{road}}, \boldsymbol{h}_{\text{region}}\} \in \mathbb{R}^h$ all reside in the same latent space, making them directly comparable and combinable for downstream tasks. To enforce consistency and semantic alignment across these different modalities, we employ a contrastive learning approach using the InfoNCE loss [26]. Although we adopt the simplest variant of the InfoNCE loss, it plays a crucial role in our framework by encouraging embeddings from the same trajectory to be close, while pushing apart those from different trajectories or modalities. Specifically, for a given trajectory $\boldsymbol{T}_q$ and its corresponding modality-specific representations $\boldsymbol{h}_{\text{mod}}$, the objective is to maximize the cosine similarity between the query embedding and its positive sample (i.e., the same embedding of trajectory from another modality), while minimizing the similarity with negative samples drawn from other trajectories. The InfoNCE loss is defined as:

$$\mathcal{L}_{\text{cont}} = -\log \frac{\exp(\text{sim}(\boldsymbol{h}_q, \boldsymbol{h}_p)/\tau)}{\sum_i \exp(\text{sim}(\boldsymbol{h}_q, \boldsymbol{h}_i)/\tau)}, \tag{9}$$

where $\boldsymbol{h}_q$ is the query embedding, $\boldsymbol{h}_p$ is the positive sample embedding (i.e., the same semantic embedding from another modality), $\boldsymbol{h}_i$ represents the negative samples. The temperature parameter τ controls the concentration of the distribution and $\text{sim}(\cdot,\cdot)$ denotes the cosine similarity calculated by: $\text{sim}(\boldsymbol{a}, \boldsymbol{b}) = \frac{\boldsymbol{a}^\top \cdot \boldsymbol{b}}{\|\boldsymbol{a}\| \|\boldsymbol{b}\|}$. The contrastive learning framework is applied in both directions: from trajectory to modality and vice versa. This bidirectional alignment loss function is formalized as:

$$\mathcal{L} = \mathcal{L}_{\text{traj}\to\text{modality}} + \mathcal{L}_{\text{modality}\to\text{traj}}. \tag{10}$$

Beyond aligning single modalities, our design also supports the fusion of multiple modality representations. In practice, we can simply concatenate the required embeddings and pass them through a linear projector to produce a fused representation. This fused embedding inherits semantic cues from each individual modality, offering even richer representations for condition-based queries. The flexibility of this design allows the model to query using a single modality, any combination thereof, or a fully fused representation, thereby catering to diverse retrieval scenarios.

3.4 Discussion

Framework Design Analysis. Our design is driven by both theoretical insights and practical requirements. We adopt a modular structure combined with the vanilla InfoNCE loss [26, 27] to project modality-specific embeddings into a shared latent space. This modular design not only facilitates efficient alignment across diverse inputs but also enables seamless integration of new modalities without extensive re-engineering. While more sophisticated contrastive losses (e.g., those with hard negative mining) or cross-attention

mechanisms might offer further alignment improvements, we prioritize simplicity and scalability. The vanilla InfoNCE loss provides a proven, robust foundation that minimizes hyperparameter tuning and preserves computational efficiency—both critical factors for deployment in large-scale, real-world scenarios. In essence, our design achieves a generalized, flexible, and scalable framework that is well-suited to city-scale datasets.

Time Complexity Analysis. The computational cost of OmniTraj mainly stems from two components: processing modality-specific embeddings through transformer layers and computing similarity scores. The self-attention in the transformer layers incurs a time complexity of $O(N^2 \times d)$, where N denotes the number of tokens (e.g., patches, segments, or regions) and d is the embedding dimension. For similarity computation, the cost is $O(|\mathcal{D}| \times n)$, where $|\mathcal{D}|$ represents the number of candidate embeddings and n is the number of query samples. Importantly, our use of vector similarity-based retrieval allows precomputation and storage of all candidate embeddings, avoiding redundant calculations and ensuring scalability and efficiency. In stark contrast, non-learned methods like DTW or Hausdorff scale quadratically $O(|\mathcal{D}|^2 \times L^2)$ with the number of trajectories and sequence length L, rendering OmniTraj orders of magnitude faster for city-scale applications.

4 Experiments

In this section, we first describe the datasets, baseline methods, and evaluation metrics used in our experiments. We then comprehensively evaluate the OmniTraj framework on two real-world datasets to answer the following research questions:

- **RQ1**: Can OmniTraj effectively retrieve trajectories compared to state-of-the-art trajectory similarity baselines?
- **RQ2**: How does OmniTraj perform when retrieving trajectories under specific conditions?
- **RQ3**: How do different parameter settings and architectural choices affect OmniTraj's performance?
- **RQ4**: Does OmniTraj exhibit transferability, scalability, and semantic understanding in real-world scenarios?

4.1 Experimental Setups

4.1.1 ***Datasets***. We conducted experiments on two large-scale trajectory datasets collected from Chengdu and Xi'an, both widely recognized for their extensive coverage, high data quality, and frequent use in trajectory analysis studies [4, 35]. For each dataset, we reserve 20,000 trajectories for testing and 50,000 for validation, while approximately 1.1 million trajectories are used for training. Detailed statistics and descriptions are provided in **Appendix B.1**.

4.1.2 ***Implement Details***. The OmniTraj implementation involves four dedicated modality encoders and hyperparameters. The implementation code is provided online[1]. See **Appendix A** for details.

4.1.3 ***Baselines***. We evaluate OmniTraj on two distinct retrieval tasks—trajectory similarity retrieval and condition-based retrieval—by comparing it against two groups of baselines. For trajectory similarity retrieval, we consider two types of baselines. First, we adopt four classical heuristic algorithms like DTW [25], EDR [6], Hausdorff [1], and Fréchet [2], which offer well-established distance or similarity measures between trajectories. Second, we incorporate four state-of-the-art self-supervised learning approaches like E2DTC [15], t2vec [20], TrjSR [3], and TrajCL [4], which learn compact embeddings for trajectory representation. During our evaluation, these methods were applied using topology modality to calculate the similarity with the target trajectory. *For condition-based retrieval, since no existing work directly addresses this task*, we adopt a set of simple multimodal approaches (e.g., basic spatial or semantic alignment techniques) as baselines. Further details regarding these baselines can be found in **Appendix B.2**.

[1] https://github.com/Yasoz/OmniTraj

Table 2: Comparison of trajectory retrieval performance with similarity computation methods.

Dataset	Method	MR	MRR	HR@1	HR@10
Chengdu	DTW	23.051	0.522	0.402	0.771
	EDR	16.15	0.279	0.137	0.587
	Hausdorff	5.643	0.760	0.682	0.897
	Fréchet	12.659	0.466	0.336	0.728
	E2DTC	8.394	0.498	0.447	0.670
	t2vec	11.474	0.513	0.354	0.638
	TrjSR	6.472	0.568	0.536	0.795
	TrajCL	1.463	0.846	0.791	0.974
	OmniTraj (reg)	2.811	0.474	0.352	0.756
	OmniTraj (reg+top)	1.441	0.825	0.762	0.932
	OmniTraj (road)	1.473	0.839	0.760	0.967
	OmniTraj (road+top)	1.394	0.845	0.785	0.944
	OmniTraj (reg+road+top)	1.401	0.843	0.782	0.943
	OmniTraj	**1.280**	**0.909**	**0.857**	**0.989**
Xi'an	DTW	35.651	0.391	0.273	0.608
	EDR	11.539	0.339	0.191	0.669
	Hausdorff	5.364	0.774	0.696	0.927
	Fréchet	12.740	0.419	0.283	0.693
	E2DTC	9.592	0.453	0.406	0.658
	t2vec	10.143	0.417	0.364	0.664
	TrjSR	7.811	0.579	0.532	0.816
	TrajCL	1.492	0.832	0.805	0.968
	OmniTraj (reg)	2.058	0.695	0.659	0.927
	OmniTraj (reg+top)	1.450	0.860	0.783	0.983
	OmniTraj (road)	1.434	0.865	0.795	0.981
	OmniTraj (road+top)	1.319	0.897	0.840	0.987
	OmniTraj (reg+road+top)	1.326	0.896	0.838	0.986
	OmniTraj	**1.304**	**0.903**	**0.847**	**0.990**

4.1.4 ***Evaluation Metrics***. For trajectory similarity retrieval tasks, we use standard ranking metrics, including Mean Rank (**MR**), Mean Reciprocal Rank (**MRR**), and Hit Rate (**HR@k**) at various cutoff values, to assess the effectiveness of the retrieval. For the condition-based retrieval task, we employ the coverage metric (**CR**) to measure the accuracy of condition matching. Specifically, we report **CR@1** and **CR@5**, which represent the proportion of query conditions correctly matched in the top-1 and top-5 retrieved results, respectively. Additional details can refer to **Appendix B.3**.

4.2 Comparison to Trajectory Similarity Models

Table 2 summarizes the performance of various trajectory retrieval or similarity computation methods on the Chengdu and Xi'an datasets. Overall, the results demonstrate that the OmniTraj consistently outperforms both classical heuristic algorithms and state-of-the-art learning-based approaches in the trajectory similarity retrieval task, particularly when using the topology modality. This superior performance is primarily attributable to OmniTraj 's innovative architecture, which effectively encodes and aligns the

Table 3: Comparison of condition-based retrieval performance with modality alignment methods.

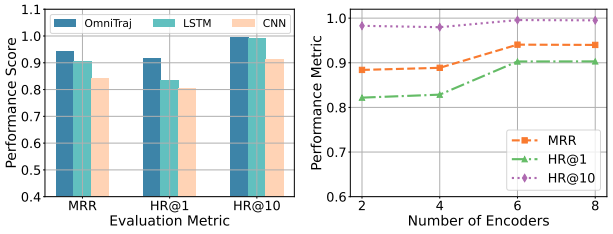
Dataset	Method	Road		Region	
		CR@1	CR@5	CR@1	CR@5
Chengdu	Embedding	0.103	0.074	0.225	0.185
	Linear	0.148	0.130	0.283	0.261
	CLIP (LSTM)	0.906	0.469	0.837	0.653
	CLIP (no-aug)	0.941	0.594	0.970	0.848
	CLIP	0.969	0.645	0.975	0.906
	OmniTraj (no-aug)	0.981	0.570	0.956	0.853
	OmniTraj	0.989	0.670	0.989	0.915
Xi'an	Embedding	0.087	0.054	0.210	0.187
	Linear	0.118	0.064	0.260	0.202
	CLIP (LSTM)	0.891	0.518	0.925	0.737
	CLIP (no-aug)	0.946	0.570	0.968	0.803
	CLIP	0.935	0.647	0.978	0.867
	OmniTraj (no-aug)	0.945	0.537	0.973	0.823
	OmniTraj	0.987	0.676	0.994	0.893

complementary semantics of trajectory and topology, thereby enabling a fine-grained understanding of spatial dynamics. Among the learning-based methods, TrajCL shows competitive results; However, OmniTraj achieves significantly higher HR@1 values, indicating a markedly improved ability to retrieve the most relevant trajectory as the top result. This above result suggests that while contrastive learning methods like TrajCL are effective for embedding trajectories, the multi-modality alignment strategy employed by OmniTraj leads to more precise and reliable retrieval outcomes. Moreover, the incorporation of topology modalities further enhances matching accuracy.

We further evaluate individual modalities by comparing OmniTraj (reg) and OmniTraj (road) to their combinations with topology (OmniTraj (reg+top), OmniTraj (road+top), and all of them OmniTraj (reg+road+top)). The results indicate that although both road segment and region semantics provide valuable contextual information, they offer a relatively coarse-grained description of trajectories, e.g., many trajectories may pass through the same road segments or regions, limiting their ability to capture detailed spatial movements. When we gradually refine the granularity of retrieval or fuse more accurate modalities, the retrieval accuracy can be further improved. Nevertheless, these modalities are still helpful for rapidly narrowing down the candidate set in a multi-stage retrieval process: coarse filtering using region or road information can be followed by fine-grained topology-based matching (we add this extra experiment in **Appendix C**). This two-stage approach can reduce computational overhead and also improve overall efficiency, especially when scaling to city-level datasets. Overall, these findings validate our claims that the OmniTraj is a flexible, scalable, and robust framework for trajectory retrieval, demonstrating high accuracy and generalizability across different urban environments.

4.3 Condition-based Retrieval Performance

Table 3 presents the performance of various condition-based retrieval methods on the Chengdu and Xi'an datasets, evaluated using the coverage metrics CR@1 and CR@5 for both the road and region modalities. The results clearly demonstrate that the OmniTraj not only excels in trajectory similarity retrieval but also significantly



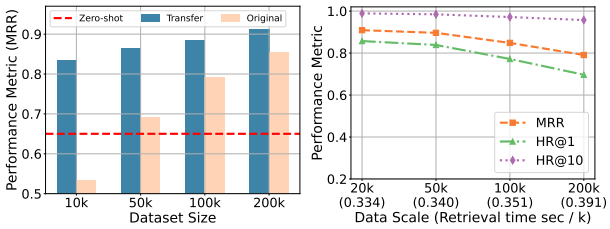
(a) Model Structure. (b) Number of Encoders.

Figure 3: Model structure and parameters and setting.

enhances the matching accuracy between query conditions and the retrieved trajectories. In particular, OmniTraj achieves the highest CR@1 scores across both datasets and modalities, outperforming baseline methods and indicating that a large proportion of query conditions are met in the top retrieval results. Notably, the robust performance of OmniTraj is maintained regardless of the encoder architecture used, whether LSTM or BERT in the CLIP approach, highlighting the flexibility and adaptability of our design. Moreover, the augmented version of OmniTraj consistently outperforms its unaugmented counterpart, especially in terms of CR@5. This improvement suggests that data augmentation substantially broadens the coverage of relevant trajectories, making the retrieval process more effective at identifying a wide range of candidates. Such an emphasis on coverage is critical for condition-based retrieval tasks, where the objective is to retrieve as many pertinent samples as possible rather than achieving pinpoint precision. Overall, these findings underscore that OmniTraj 's multimodal alignment strategy captures complex semantic relationships between queries and target trajectories and supports flexible, high-coverage retrieval across diverse conditions. We also provide an analysis of the coverage changes with the length of the sequences and the range of the retrieved in **Appendix C**.

4.4 Architecture Study and Parameters Setting

Furthermore, we examine the influence of different architectural choices and the number of encoder blocks on OmniTraj 's performance. Figure 3(a) compares the original OmniTraj architecture with alternative designs based on LSTM and CNN. The results clearly indicate that the original OmniTraj architecture delivers superior performance, significantly surpassing the LSTM and CNN variants. As we transition from OmniTraj to the other architectures, a gradual decline in retrieval accuracy is observed, underscoring the critical role of an effective multimodal encoding strategy. In particular, the combination of sequence modeling and attention mechanisms in OmniTraj enables a more robust alignment and better captures the underlying semantic relationships than traditional sequence-based models. Figure 3(b) further investigate the effect of varying the number of encoder blocks. As the number of blocks increases from 2 to 6, there is a steady improvement in retrieval performance, suggesting that additional layers enhance the model's capacity to capture and align complex multimodal representations. Notably, even with only two encoder blocks, the model achieves relatively good performance, demonstrating the inherent strength of our design. However, when the number of encoder blocks exceeds 6, the performance gains plateau, indicating that the model reaches a saturation point where further increases in depth yield



(a) Transfer learning. (b) Data scale impact.

Figure 4: Model transferability and scalability analysis.

diminishing returns. The above finding highlights the importance of balancing model complexity and computational efficiency, as increasing the number of encoder blocks beyond a certain threshold yields diminishing returns.

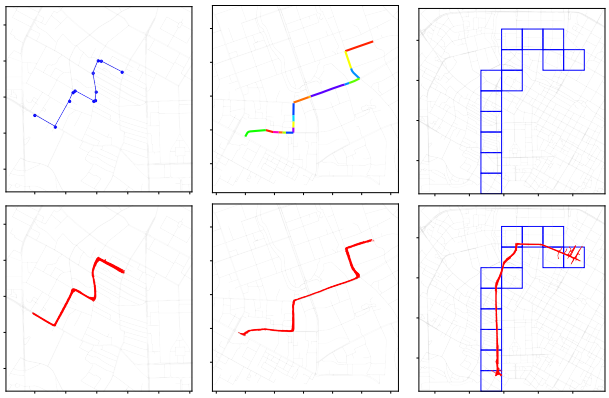
4.5 Model Applicability Analysis

In this section, we analyze the applicability of the OmniTraj by examining its transferability, scalability, and semantic understanding.

Transfer Learning. We first evaluate the zero-shot performance of the model and its ability to transfer between urban environments. Specifically, we trained a model on data from Chengdu city and then directly applied it to a Xi'an city, examining robust zero-shot capability. To further assess transferability, we fine-tune the pre-trained model on the target city using varying amounts of training samples (10k, 50k, 100k, and 200k trajectories). As shown in Figure 4(a), testing directly with pre-trained models in a new environment can also achieve comparable zero-shot performance to training with a 50k data size. This is due to the fact that our model encodes topological modalities without adding any additional city-related information, allowing it to have better generalization capabilities. The above statement is further verified by transfer learning. It clearly observes that even with limited data in the target domain, the performance of the model improves significantly ("transfer" for applying transfer learning with a pre-trained model and "origin" for direct training from scratch). As the amount of data increases, the gap between the two models also reduces as the amount of data increases, validating the applicability of our OmniTraj to the new urban environment.

Scalability. We check the scalability of the model by analyzing how the retrieval performance and processing time of the model change as the size of the dataset increases. We gradually increase the number of trajectories used for retrieval (20k, 50k, 100k, and 200k) and observe that although the overall retrieval performance decreases slightly with the increase in data size, the decrease is within acceptable limits. At the same time, the processing time per 1,000 query samples gradually increases from 0.334 seconds at 20k to 0.391 seconds at 200k. This smooth trend reflects a reasonable trade-off between performance and efficiency when the model is scaled to handle larger datasets and can be effectively applied to large-scale data retrieval.

Semantic Understanding. To assess OmniTraj's semantic understanding and its utility for downstream applications, we conducted a pioneering trajectory generation task. These encoders are then used to generate semantic embeddings that serve as conditional guidance inputs for the trajectory generation model [43]. As vividly illustrated in Figure 5, the generated trajectories consistently and



(a) Topology. (b) Road. (c) Region.

Figure 5: Use OmniTraj as a semantic representation to guide the model for trajectory generation.

accurately match the specified conditions. This clear visualization unequivocally confirms that OmniTraj effectively captures and utilizes the intricate semantic information embedded within different modalities. The capability underscores OmniTraj's profound understanding of the underlying spatial and semantic relationships within trajectories. onsequently, it demonstrates OmniTraj's robust potential to successfully support a wide range of downstream applications, including but not limited to targeted trajectory generation, intelligent route planning, and other spatially constrained tasks.

Overall, the above experiments validate that the OmniTraj is flexible and scalable and also exhibits strong transferability and semantic understanding, making it well-suited for real-world, large-scale trajectory retrieval and related applications.

5 Related work

Trajectory Retrieval and Querying. Trajectory retrieval seeks to identify trajectories from a database that meet specific query criteria, such as similarity to a given trajectory or adherence to particular feature-based conditions [31, 40]. This task is fundamental to spatio-temporal data analysis and underpins applications in intelligent transportation, urban analytics, and mobility recommendation [9, 10]. Most existing research emphasizes trajectory similarity measures, where the goal is to match a given query trajectory with candidate trajectories in terms of spatio-temporal closeness [7]. Traditionally, trajectory similarity is computed using distance measures (e.g., Dynamic Time Warping [25], Hausdorff [1], or Fréchet [2]) or accelerated via indexing structures (e.g., grid-based partitions [3, 4, 20] or road network mappings [19, 37]). However, these methods focus on matching entire trajectories and require processing all points, making them computationally expensive and ill-suited for flexible, feature-specific queries. More recent deep learning approaches have leveraged trainable embeddings—using RNN-based encoders or attention mechanisms—to capture complex semantic and motion dynamics [20, 36, 41], converting trajectories into compact embedding vectors that improve matching accuracy and efficiency. Nevertheless, these techniques largely address whole-trajectory similarity measures and offer limited support for partial or condition-based retrieval. Condition-based retrieval, in contrast,

aims to retrieve trajectories that satisfy specific constraints. While traditional indexing techniques can perform simple filtering, deep learning solutions have yet to fully integrate diverse trajectory features into a unified representation that supports precise condition-based queries [17, 33, 39]. This gap highlights the need for novel frameworks that reconcile the strengths of deep embeddings with the demands of multi-aspect trajectory retrieval.

Multimodal Learning. Multimodal learning, which involves integrating and processing information from multiple data sources or modalities, has emerged as a pivotal area in machine learning, significantly advancing fields such as Computer Vision (CV) and Natural Language Processing (NLP) [11, 34, 38]. Taking models such as CLIP as an example, a multimodal approach that combines image and textual descriptions enhances the robustness of visual representations and semantic understanding, while producing context-aware language models [27]. In addition, combining text data with other modalities, such as video and image inputs, can align the semantic features of different modalities to achieve cross-modal retrieval and querying [24]. Extending these advancements to trajectory data, multimodal learning facilitates the comprehensive analysis of trajectories by incorporating diverse aspects such as road segments, street views, and regional POI information [23, 35]. For example, the existing work incorporates multiple embeddings through a multi-task learning framework by combining trajectories with the rest of the corresponding modalities, which can effectively facilitate the trajectory-based representation learning task. However, existing multimodal learning approaches primarily focus on enhancing target representations through the fusion of multiple modalities, thereby improving tasks like classification, prediction, and anomaly detection [22, 42]. There is a noticeable gap in addressing multimodal retrieval, where the goal is to enable cross-modal or condition-based queries across different trajectory aspects. This limitation underscores the novelty of our work, which seeks to bridge this gap by developing a unified retrieval framework that leverages multimodal representations for flexible trajectory query.

6 Conclusion

In this work, we introduce a new task in multimodal trajectory retrieval that enables condition-based queries across different trajectory semantics, including topology, road segments, and regions. To achieve this target, we propose OmniTraj, a unified omni-semantic trajectory retrieval framework designed to offer efficiency, flexibility, and comprehensive support for complex queries. By leveraging semantic decoupling and a modular encoder design, OmniTraj ensures accurate and scalable retrieval while supporting applications that operate under various semantic constraints. This innovative approach addresses the limitations of traditional trajectory similarity measures, providing a fresh perspective on how trajectory retrieval tasks can be tackled. Future work will extend OmniTraj to dynamic environments and real-time trajectory streams, further enhancing the generalization and flexibility of the proposed model.

7 Acknowledgment

This work is mainly supported by the National Natural Science Foundation of China (No. 62402414). This work is also supported by the Guangdong Basic and Applied Basic Research Foundation (No. 2025A1515011994), Guangzhou Municipal Science and Technology Project (No. 2023A03J0011), the Guangzhou Industrial Information and Intelligent Key Laboratory Project (No. 2024A03J0628), and a grant from State Key Laboratory of Resources and Environmental Information System, and Guangdong Provincial Key Lab of Integrated Communication, Sensing and Computation for Ubiquitous Internet of Things (No. 2023B1212010007). This research was partially supported by Research Impact Fund (No.R1015-23), Collaborative Research Fund (No.C1043-24GF), Huawei (Huawei Innovation Research Program, Huawei Fellowship), Tencent (CCF-Tencent Open Fund, Tencent Rhino-Bird Focused Research Program), Alibaba (CCF-Alimama Tech Kangaroo Fund No. 2024002), Ant Group (CCF-Ant Research Fund), and Kuaishou.

References

[1] Helmut Alt. 2009. The computational geometry of comparing shapes. *Efficient Algorithms: Essays Dedicated to Kurt Mehlhorn on the Occasion of His 60th Birthday* (2009), 235–248.

[2] Helmut Alt and Michael Godau. 1995. Computing the Fréchet distance between two polygonal curves. *International Journal of Computational Geometry & Applications* 5, 01n02 (1995), 75–91.

[3] Hanlin Cao, Haina Tang, Yulei Wu, Fei Wang, and Yongjun Xu. 2021. On accurate computation of trajectory similarity via single image super-resolution. In *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–9.

[4] Yanchuan Chang, Jianzhong Qi, Yuxuan Liang, and Egemen Tanin. 2023. Contrastive trajectory similarity learning with dual-feature attention. In *2023 IEEE 39th International conference on data engineering (ICDE)*. IEEE, 2933–2945.

[5] Yanchuan Chang, Egemen Tanin, Gao Cong, Christian S Jensen, and Jianzhong Qi. 2024. Trajectory Similarity Measurement: An Efficiency Perspective. *Proceedings of the VLDB Endowment* 17, 9 (2024), 2293–2306.

[6] Lei Chen, M Tamer Özsu, and Vincent Oria. 2005. Robust and fast similarity search for moving object trajectories. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*. 491–502.

[7] Lisi Chen, Shuo Shang, Christian S Jensen, Bin Yao, and Panos Kalnis. 2020. Parallel semantic trajectory similarity join. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. IEEE, 997–1008.

[8] Wei Chen, Yuxuan Liang, Yuanshao Zhu, Yanchuan Chang, Kang Luo, Haomin Wen, Lei Li, Yanwei Yu, Qingsong Wen, Chao Chen, et al. 2024. Deep Learning for Trajectory Data Management and Mining: A Survey and Beyond. *arXiv preprint arXiv:2403.14151* (2024).

[9] Jian Dai, Bin Yang, Chenjuan Guo, and Zhiming Ding. 2015. Personalized route recommendation using big trajectory data. In *2015 IEEE 31st international conference on data engineering*. IEEE, 543–554.

[10] Rodrigo Augusto de Oliveira e Silva, Ge Cui, Seyyed Mohammadreza Rahimi, and Xin Wang. 2022. Personalized route recommendation through historical travel behavior analysis. *GeoInformatica* 26, 3 (2022), 505–540.

[11] Jinliang Deng, Xiusi Chen, Renhe Jiang, Xuan Song, and Ivor W Tsang. 2022. A multi-view multi-task learning framework for multi-variate time series forecasting. *IEEE Transactions on Knowledge and Data Engineering* 35, 8 (2022).

[12] Jinliang Deng, Feiyang Ye, Du Yin, Xuan Song, Ivor Tsang, and Hui Xiong. 2024. Parsimony or capability? decomposition delivers both in long-term time series forecasting. *Advances in Neural Information Processing Systems* 37 (2024).

[13] Ke Deng, Kexin Xie, Kevin Zheng, and Xiaofang Zhou. 2011. Trajectory indexing and retrieval. *Computing with spatial trajectories* (2011), 35–60.

[14] Alexey Dosovitskiy. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

[15] Ziquan Fang, Yuntao Du, Lu Chen, Yujia Hu, Yunjun Gao, and Gang Chen. 2021. E2dtc: An end to end deep trajectory clustering framework via self-training. In *2021 IEEE 37th International Conference on Data Engineering*. IEEE, 696–707.

[16] Ziquan Fang, Yuntao Du, Xinjun Zhu, Danlei Hu, Lu Chen, Yunjun Gao, and Christian S Jensen. 2022. Spatio-temporal trajectory similarity learning in road networks. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*. 347–356.

[17] Chongming Gao, Zhong Zhang, Chen Huang, Hongzhi Yin, Qinli Yang, and Junming Shao. 2020. Semantic trajectory representation and retrieval via hierarchical embedding. *Information Sciences* 538 (2020), 176–192.

[18] Chenjuan Guo, Bin Yang, Jilin Hu, Christian S Jensen, and Lu Chen. 2020. Context-aware, preference-based vehicle routing. *The VLDB Journal* 29 (2020), 1149–1170.

[19] Peng Han, Jin Wang, Di Yao, Shuo Shang, and Xiangliang Zhang. 2021. A graph-based approach for trajectory similarity computation in spatial networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 556–564.

[20] Xiucheng Li, Kaiqi Zhao, Gao Cong, Christian S Jensen, and Wei Wei. 2018. Deep representation learning for trajectory similarity computation. In *2018 IEEE 34th international conference on data engineering (ICDE)*. IEEE, 617–628.
[21] Yan Lin, Huaiyu Wan, Shengnan Guo, Jilin Hu, Christian S Jensen, and Youfang Lin. 2023. Pre-training general trajectory embeddings with maximum multi-view entropy coding. *IEEE Transactions on Knowledge and Data Engineering* 36, 12 (2023), 9037–9050.
[22] Yan Lin, Tonglong Wei, Zeyu Zhou, Haomin Wen, Jilin Hu, Shengnan Guo, Youfang Lin, and Huaiyu Wan. 2024. TrajFM: A Vehicle Trajectory Foundation Model for Region and Task Transferability. *arXiv:2408.15251* (2024).
[23] Yan Lin, Zeyu Zhou, Yicheng Liu, Haochen Lv, Haomin Wen, Tianyi Li, Yushuai Li, Christian S Jensen, Shengnan Guo, Youfang Lin, et al. 2024. UniTE: A Survey and Unified Pipeline for Pre-training ST Trajectory Embeddings. *arXiv e-prints* (2024), arXiv–2407.
[24] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. 2022. Clip4clip: An empirical study of clip for end to end video clip retrieval and captioning. *Neurocomputing* 508 (2022), 293–304.
[25] Meinard Müller. 2007. Dynamic time warping. *Information retrieval for music and motion* (2007), 69–84.
[26] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).
[27] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 8748–8763.
[28] Han Su, Shuncheng Liu, Bolong Zheng, Xiaofang Zhou, and Kai Zheng. 2020. A survey of trajectory distance measures and performance evaluation. *The VLDB Journal* 29 (2020), 3–32.
[29] Han Su, Kai Zheng, Haozhou Wang, Jiamin Huang, and Xiaofang Zhou. 2013. Calibrating trajectory data for similarity-based analysis. In *Proceedings of the 2013 ACM SIGMOD international conference on management of data*. 833–844.
[30] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing* 568 (2024), 127063.
[31] David Alexander Tedjopurnomo, Xiucheng Li, Zhifeng Bao, Gao Cong, Farhana Choudhury, and A Kai Qin. 2021. Similar trajectory search with spatio-temporal deep representation learning. *ACM Transactions on Intelligent Systems and Technology (TIST)* 12, 6 (2021), 1–26.
[32] Sheng Wang, Zhifeng Bao, J Shane Culpepper, and Gao Cong. 2021. A survey on trajectory data management, analytics, and learning. *ACM Computing Surveys (CSUR)* 54, 2 (2021), 1–36.
[33] Yong Wang, Kaiyu Li, Guoliang Li, and Nan Tang. 2022. Road-Aware Indexing for Trajectory Range Queries. *IEEE Transactions on Knowledge and Data Engineering* 35, 8 (2022), 8476–8489.
[34] Peng Xu, Xiatian Zhu, and David A Clifton. 2023. Multimodal learning with transformers: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 10 (2023), 12113–12132.
[35] Ronghui Xu, Hanyin Cheng, Chenjuan Guo, Hongfan Gao, Jilin Hu, Sean Bin Yang, and Bin Yang. 2024. MM-Path: Multi-modal, Multi-granularity Path Representation Learning–Extended Version. *arXiv preprint arXiv:2411.18428* (2024).
[36] Di Yao, Gao Cong, Chao Zhang, and Jingping Bi. 2019. Computing trajectory similarity in linear time: A generic seed-guided neural metric learning approach. In *2019 IEEE 35th international conference on data engineering (ICDE)*. IEEE, 1358–1369.
[37] Di Yao, Haonan Hu, Lun Du, Gao Cong, Shi Han, and Jingping Bi. 2022. TrajGAT: A graph-based long-term dependency modeling approach for trajectory similarity computation. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*. 2275–2285.
[38] Yuan Yuan, Zhaojian Li, and Bin Zhao. 2025. A Survey of Multimodal Learning: Methods, Applications, and Future. *Comput. Surveys* (2025).
[39] Bolong Zheng, Nicholas Jing Yuan, Kai Zheng, Xing Xie, Shazia Sadiq, and Xiaofang Zhou. 2015. Approximate keyword search in semantic trajectory database. In *2015 IEEE 31st International conference on data engineering*. IEEE, 975–986.
[40] Kai Zheng, Yan Zhao, Defu Lian, Bolong Zheng, Guanfeng Liu, and Xiaofang Zhou. 2019. Reference-based framework for spatio-temporal trajectory compression and query processing. *IEEE Transactions on Knowledge and Data Engineering* 32, 11 (2019), 2227–2240.
[41] Silin Zhou, Jing Li, Hao Wang, Shuo Shang, and Peng Han. 2023. GRLSTM: trajectory similarity computation with graph-based residual LSTM. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 4972–4980.
[42] Silin Zhou, Shuo Shang, Lisi Chen, Peng Han, and Christian S Jensen. 2024. Grid and Road Expressions Are Complementary for Trajectory Representation Learning. *arXiv preprint arXiv:2411.14768* (2024).
[43] Yuanshao Zhu, James Jianqiao Yu, Xiangyu Zhao, Qidong Liu, Yongchao Ye, Wei Chen, Zijian Zhang, Xuetao Wei, and Yuxuan Liang. 2024. Controltraj: Controllable trajectory generation with topology-constrained diffusion model. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4676–4687.
[44] Yuanshao Zhu, James Jianqiao Yu, Xiangyu Zhao, Xuetao Wei, and Yuxuan Liang. 2024. Unitraj: Learning a universal trajectory foundation model from billion-scale worldwide traces. *CoRR* (2024).

A Details of Framework

Structure of Encoders. As outlined in Section 3 and depicted in Figure 6, OmniTraj adopts a modular, decoupled design consisting of four independent encoders. Specifically, for the topology encoder, we first constructed a linear layer to project the topology points to a high-dimensional space and then added the [CLS] token and location information for processing by the RoPE-based Transformer block. Finally, the acquired high-dimensional vector embeddings are processed using the corresponding pooling strategy to obtain a semantic representation of the topology. For the road encoder, we replace the linear projection layer with an embedding layer to construct a unique embedding vector for each road identifier. The rest of the process is the same as for the topological encoder. For the region encoder, since we needed to cover as much of the sequence of the retrieval as possible, we removed the position encoding and used the native Transformer module. Finally, for the projection head, which transforms individual modality-specific representations into a shared latent space, we use two stacked linear layers. This simple but effective approach ensures that the multimodal embeddings can be aligned and compared in a common space, facilitating flexible multimodal querying.

Table 4: Parameters setting and statistics of OmniTraj.

	Traj. Enc	Top. Enc	Road. Enc	Reg. Enc
Input dim	2	2	1	1
Output dim	256	256	256	256
Proj. dim	512	512	512	512
Max. Seq. length	200	128	128	64
Blocks	6	6	6	6
Pooling	cls	cls	bos	cls
Atten. heads	8	8	8	8
Num. paras (M)	4.75	4.75	4.17	4.81

Implementation Details. We summarize the specific parameter settings for each module in OmniTraj, including dimensions, number of modules, and number of parameters. Table 4 highlights the key configuration details for each of the encoders in OmniTraj. All encoders share the same output dimension of 256, and the projection dimension is set to 512 across all modalities. The encoders are designed with six Transformer blocks each, ensuring a balance between model depth and computational efficiency. The attention mechanism is equipped with eight attention heads to capture diverse semantic features. The number of parameters in each encoder is around 4.7 million, with minor variations depending on the modality. These settings ensure that OmniTraj is capable of handling large-scale trajectory data efficiently while maintaining flexibility and accuracy in multimodal retrieval tasks.

B Experiment and Setup

The experiments were conducted in a controlled environment with the following specifications. All models were trained on a machine equipped with an NVIDIA A100 GPU and evaluated with an A6000

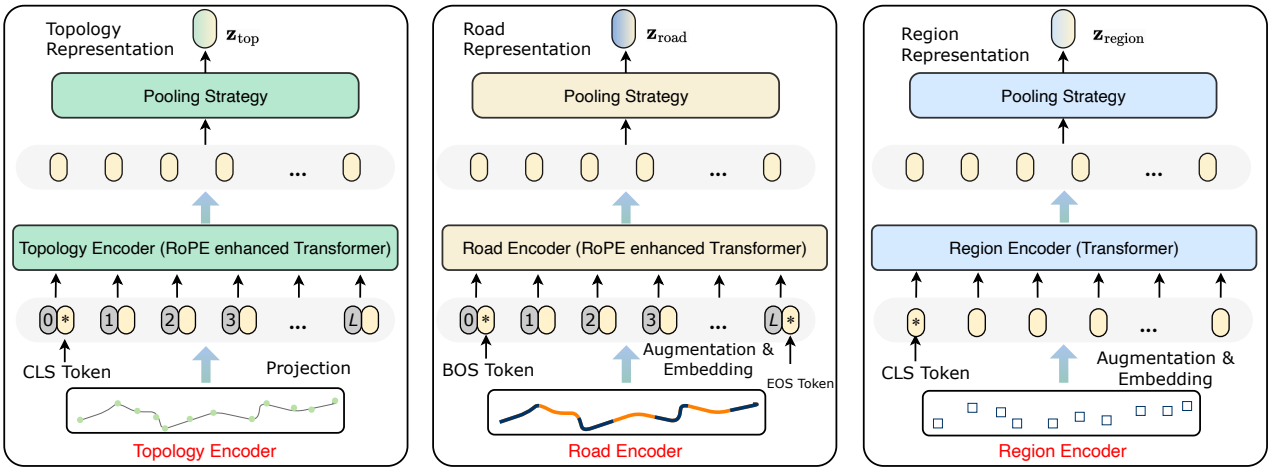


Figure 6: The main structure of each encoder in the OmniTraj framework, the trajectory encoder is shown in the main content.

GPU, which provided sufficient computational power for handling large-scale trajectory datasets. The code was implemented using Python 3.9 and PyTorch 1.8, leveraging the capabilities of CUDA for efficient GPU computation. We use the Adam optimizer InfoNCE loss with an initial learning rate of 2×10^{-4}.

B.1 Dataset

We evaluated the model performance of OmniTraj and baseline on trajectory datasets from two cities, Chengdu and Xi'an. Following the approach in [4, 5], we filtered out trajectories recorded outside urban areas or containing fewer than 20 points and transformed them to a fixed length of 200 using interpolation. For the Chengdu dataset, there are 7597 road segments, and for Xi'an, there are 6018 identifiers. We divided the two cities into $16 \times 16 = 256$ grids to represent the regions. Table 5 summarizes the statistics of the resulting preprocessed datasets, where "Avg. ", "Avg. Road", and "Avg. Region" indicates the average number of topological points, road segments, and regions per trajectory, respectively.

Table 5: Statistics of Two Trajectory Datasets.

Dataset	#Trajetory	Avg. Topology	Avg. Road	Avg. Region
Chengdu	1 224 317	13.33	21.08	9.52
Xian	1 175 949	12.02	22.12	9.27

B.2 Baselines

For the trajectory similarity retrieval task, we use a range of heuristics and deep learning-based algorithms and compare variants of OmniTraj and its modality combinations. Due to the high time complexity of heuristic methods, we restrict their evaluation to 1,000 query samples drawn from the 20,000 test trajectories, whereas the learning-based approaches are evaluated on the entire testing set.

- **DTW** [25]. A classical method for computing the similarity between trajectories by optimally aligning their sequences using dynamic time warping.
- **EDR** [6]: The Edit Distance on Real Sequence (EDR) metric measures trajectory similarity by considering spatial mismatches and is robust to noise and outliers.
- **Hausdorff** [1]: A metric that quantifies the maximum deviation between two trajectories, measuring the worst-case distance between point sets.
- **Fréchet** [2]: A distance measure that takes into account the continuity and order of points, often referred to as the "dog-walking" distance between curves.
- **E2DTC** [15]: A self-supervised method that learns trajectory embeddings by capturing spatio-temporal patterns through clustering learning techniques.
- **t2vec** [20]: A sequence-to-sequence deep learning-based approach that maps trajectories into a latent space using sequence modeling to capture the overall motion dynamics.
- **TrjSR** [3]: Convert trajectories to images and use generative models combined with image super-resolution to learn trajectory representations.
- **TrajCL**: [4]: A recent contrastive learning-based framework specifically designed for trajectory similarity computation, which leverages contrastive learning for model training.
- **OmniTraj**: The OmniTraj framework utilizes only topology modality for optimal trajectory retrieval. In addition, we tested a series of OmniTraj's variants, OmniTraj (modality), where modality represents different modes or their combinations. For example, OmniTraj (road) leverages road segments to generate trajectory embeddings and OmniTraj (reg+road+top) combines region, road, and topology modalities.

For condition-based retrieval, since no existing work directly addresses this task, we adopt a set of simple approaches as baselines. All of these baselines adopt InfoNCE for modality alignment.

- **Embedding**: It uses the embeddings generated from the lookup matrix, serving as a basic reference for comparison.
- **Linear**: A linear projection model where the trajectory embeddings are passed through a linear layer for further transformation.
- **CLIP**: CLIP encodes both the trajectory and condition (such as road or region) into shared space using vision and language principles (Bert), making it applicable for multimodal queries.
- **CLIP (LSTM)**: An variant of the CLIP model that uses an LSTM-based backbone to process sequential data instead of Bert.

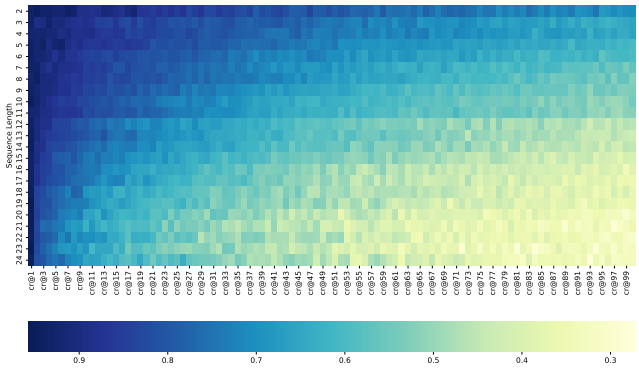


Figure 8: Coverage changes with the length of the retrieved road sequence and the breadth of the retrieval. Sequence length increases from top to bottom and retrieval breadth increases from left to right.

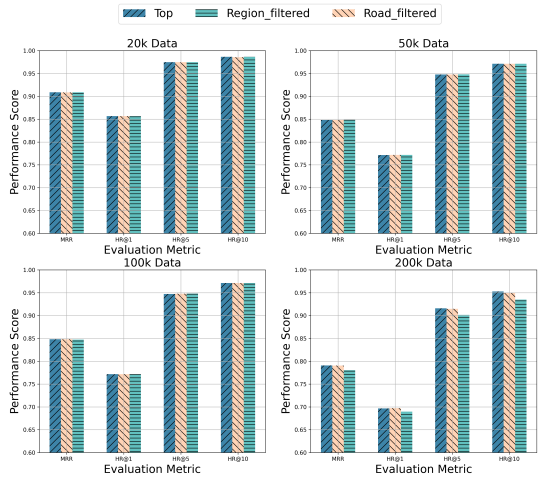


Figure 7: Comparison of retrieval of trajectories using a two-stage approach with coarse-grained modal filtering followed by precise retrieval using topology modalities in a subset.

- **CLIP (no-aug)**: This version of CLIP is evaluated without the use of data augmentation, providing a baseline to see the impact of augmentation techniques on retrieval performance.
- **OmniTraj**: The proposed method integrates modalities such as road segments, and regions for trajectory retrieval. This model is our primary approach for evaluating the effectiveness of multimodal retrieval in comparison to simpler models.
- **OmniTraj (no-aug)**: This baseline is similar to OmniTraj but does not utilize data augmentation during training. It serves as a control to assess how the augmentation impacts the model's performance on condition-based retrieval tasks.

B.3 Evaluation Metrics

We employ multiple ranking metrics to evaluate the retrieval performance: Mean Rank (MR) calculates the average position of the target sequence in the retrieved results. A lower MR indicates better performance as the target appears closer to the top of the retrieval list. Mean Reciprocal Rank (MRR) computes the average inverse rank of the first relevant result. Formally defined as:

$$\mathbf{MR} = \frac{1}{|Q|}\sum_{q=1}^{Q} r_q, \ \mathbf{MRR} = \frac{1}{|Q|}\sum_{q=1}^{Q}\frac{1}{r_q}, \tag{11}$$

where Q is the set of queries, and r_q is the rank position of the first relevant result for the q-th query. MRR ranges from 0 to 1, with higher values indicating better performance. Hit Rate at k (HR@k) measures the percentage of queries where at least one relevant result appears in the top-k retrieved sequences. We report HR@1 and HR@5 to evaluate the model's ability to retrieve relevant sequences within the top-1 and top-5 results respectively.

For condition-based retrieval, we evaluate the retrieval performance using Containment Rate (CR@k), which measures the proportion of query elements present in the top-K retrieved sequences. Specifically, CR@1 calculates the containment rate for the top-1 retrieved sequence, while CR@5 considers the union of elements from the top-5 retrieved sequences. Specifically, we evaluate the retrieval results using Containment Rate (CR), which measures the proportion of query elements present in the retrieved sequence:

$$\mathbf{CR@k} = \frac{|Q \cap \mathcal{R}_\mathbf{k}|}{|Q|}, \tag{12}$$

where Q denotes the query set and $\mathcal{R}_\mathbf{k}$ represents the union of elements from the top-k retrieved sequences. A higher **CR@K** indicates better coverage of the query elements in retrieved results.

C Supplementary Experiments

Two-Stage Retrieval. We conduct our experiments using a two-stage approach for trajectory retrieval. First, we employ coarse-grained modalities (such as roads and regions) to quickly filter and narrow down the dataset. Next, we apply topology-based fine-grained matching. We performed the experiments separately with different data sizes, ultimately reducing them to subsets of 200 for roads and 500 for regions. As illustrated in 7, we observe no significant difference in subset retrieval after applying coarse-grained modal filtering compared to retrieval from the complete dataset. Notably, even as the data size increases to 200k, our method maintains strong robustness. The above results show that our coarse-grained-based retrieval is highly flexible and is able to significantly reduce computational overhead and improve overall efficiency with this two-stage approach.

Conditional Retrieval. Finally, we further explore the conditional retrieval flexibility and robustness of the OmniTraj. The visualization results in Figure 8 demonstrate two aspects: the first is the impact of the desired sequence length on the model, and the second is the range accuracy of the model retrieval. It is clear from this result that the OmniTraj is sufficiently robust to the increase in retrieval length and does not sacrifice accuracy due to its increase. Considering that our approach is not limited to exact sequence retrieval, it should also support queries such as retrieving trajectories that pass through or cover a specific sequence of road segments. We significantly increase the range (from 1 to 100) of the retrieval, and the results show that the sequences it retrieves can be well covered. Note that longer retrieved sequences imply more precise matches, and the fewer in the set to be retrieved satisfy this requirement, thus resulting in lower coverage.