UniTraj: Learning a Universal Trajectory Foundation Model from Billion-Scale Worldwide Traces

Yuanshao Zhu[1,2,3,*], **James Jianqiao Yu**[4,†], **Xiangyu Zhao**[2,†], **Xun Zhou**[4], **Liang Han**[4], **Xuetao Wei**[1], **Yuxuan Liang**[3,†]

[1] Southern University of Science and Technology, [2] City University of Hong Kong
[3] The Hong Kong University of Science and Technology (Guangzhou)
[4] Harbin Institute of Technology, Shenzhen

yuanshao@ieee.org, jqyu@ieee.org, xianzhao@cityu.edu.hk
zhouxun2023@hit.edu.cn, han.liang@hit.edu.cn
weixt@sustech.edu.cn, yuxliang@outlook.com

Abstract

Building a universal trajectory foundation model is a promising solution to address the limitations of existing trajectory modeling approaches, such as task specificity, regional dependency, and data sensitivity. Despite its potential, data preparation, pre-training strategy development, and architectural design present significant challenges in constructing this model. Therefore, we introduce **UniTraj**, a Universal Trajectory foundation model that aims to address these limitations through three key innovations. First, we construct **WorldTrace**, an unprecedented dataset of 2.45 million trajectories with billions of GPS points spanning 70 countries, providing the diverse geographic coverage essential for region-independent modeling. Second, we develop novel pre-training strategies—Adaptive Trajectory Resampling and Self-supervised Trajectory Masking—that enable robust learning from heterogeneous trajectory data with varying sampling rates and quality. Finally, we tailor a flexible model architecture to accommodate a variety of trajectory tasks, effectively capturing complex movement patterns to support broad applicability. Extensive experiments across multiple tasks and real-world datasets demonstrate that UniTraj consistently outperforms existing methods, exhibiting superior scalability, adaptability, and generalization, with WorldTrace serving as an ideal yet non-exclusive training resource. The implementation codes and full dataset are available at https://github.com/Yasoz/UniTraj.

1 Introduction

Trajectory data, as the digital footprints of human movement, is becoming a fundamental data source for understanding mobility patterns and transforming urban intelligence [3]. These spatio-temporal sequences unlock critical insights across diverse applications: from optimizing transportation networks that alleviate congestion in megacities, enhancing location-based services that personalize user experiences [18, 2], to powering logistics systems that determine the efficiency of global supply chains [12, 24, 36]. Despite their significance, extracting meaningful patterns (from statistical methods to deep learning [25]) of trajectory data presents profound challenges due to their inherent complexity, varying lengths, irregular sampling rates, and region-specific characteristics.

As trajectory data continues to expand exponentially, three critical limitations in current approaches have become increasingly apparent: (1) **Task Specificity**: Current approaches are typically designed

[*] Work done during the internship at HKUST(GZ)
[†] Corresponding author

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

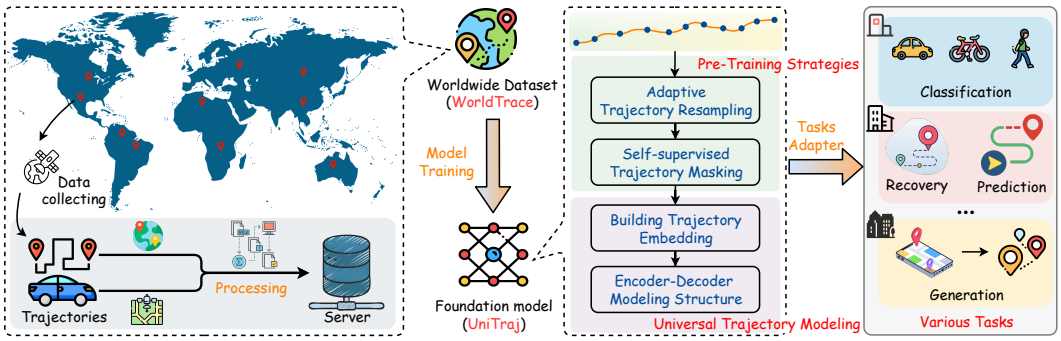


Figure 1: Overview of this work, we propose a trajectory foundation model and also collect a worldwide trajectory dataset. The pre-trained UniTraj can be used as a backbone while adapters are trained for different regions and tasks.

for single-purpose applications, limiting their generalizability and requiring substantial re-engineering for new tasks. (2) **Regional Dependency**: Many models are developed and trained on data from specific geographic regions, making them ineffective when applied to different locations with distinct mobility patterns and infrastructure. (3) **Data Sensitivity**: Real-world trajectory data often contains noise, irregular sampling, or missing entries, making models highly sensitive to data quality and necessitating extensive preprocessing, which reduces robustness. These limitations point to a fundamental gap: the absence of a universal foundation model capable of operating across diverse tasks, geographic regions, and data quality levels. While foundation models have revolutionized NLP [1, 7] and CV [9, 13] by providing versatile, pre-trained architectures that generalize across domains, trajectory analysis has not yet benefited from this paradigm shift. Creating this model would transform trajectory intelligence from its current fragmented state to a unified approach with significantly enhanced generalization capabilities [27, 39].

However, building such a model presents key challenges: (1) **Data preparation:** The first challenge is to prepare a sufficiently diverse trajectory dataset that spans different geographic regions and appropriate sampling rates. Existing datasets lack sufficient geographic diversity and scale, also limited by proprietary restrictions and collection costs. This data scarcity severely hampers model generalizability and cross-regional research efforts on a global scale. (2) **Pre-training Strategy:** Developing robust and scalable pre-training strategies is another challenge. Real-world trajectory data exhibits heterogeneous quality with noising, varying sampling rates, and missing points. Effective pre-training must accommodate these inconsistencies while learning robust representations that transfer across diverse contexts. (3) **Model Design:** The last challenge involves selecting and tailoring an effective model architecture. A universal foundation model requires an architecture that balances adaptability across tasks with computational efficiency, capturing complex spatio-temporal dependencies without overfitting to specific regional information or trajectory patterns.

To address these challenges, we introduce **Uni**versal **Traj**ectory foundation model (**UniTraj**) supported by three key innovations. As shown in Figure 1, we firstly construct **WorldTrace**, the first trajectory dataset with large-scale, high-quality, and global distribution, which provides the essential foundation for region-agnostic modeling. Then, we design several novel pre-training strategies—adaptive resampling and self-supervised masking—that enable robust learning from heterogeneous trajectory data with varying sampling rates and quality, bridging the gap between regional variations and inconsistent data. Finally, we design a flexible model architecture that captures complex spatio-temporal dependencies while adapting to diverse trajectory tasks, creating a versatile backbone for trajectory modeling. Collectively, UniTraj achieves *task-adaptive, region-independent, and data quality resilience*, delivering a scalable and efficient solution for trajectory analysis applications. In summary, our research makes the following key contributions:

- We introduce WorldTrace, a pioneering trajectory dataset spanning 70 countries with 2.45 million trajectories and billions of GPS points. Its unprecedented global diversity and quality overcome the limitations of existing region-specific datasets, offering a comprehensive and open groundwork for facilitating trajectory modeling research.
- We propose UniTraj, trained on WorldTrace and equipped with novel pre-training and masking strategies that effectively capture complex spatio-temporal dependencies. This model significantly

enhances generalizability across tasks and geographical contexts, adapts to the heterogeneity of data, and provides a scalable and efficient solution for a wide range of trajectory analysis applications.

- We demonstrated the effectiveness of UniTraj through comprehensive experiments on multiple trajectory analysis tasks. The results show significantly improved performance of zero-shot and fine-tuning settings, confirming its potential as a versatile backbone for diverse trajectory modeling tasks, performing optimally when trained on diverse and high-quality datasets like WorldTrace.

2 Related Work

Trajectory Datasets. Trajectory datasets are foundational for advancing mobility research, yet existing collections vary (geographic coverage, data quality, and granularity) considerably in their utility and limitations. Well-known datasets, such as GeoLife [46], collected over five years by 182 users, has contributed significantly to fields like travel mode detection [5] and traffic flow analysis [19]. However, its limited geographic coverage and participant diversity restrict its generalizability. ehicle-focused datasets such as Porto [28], T-drive [42], and Electric Vehicle Data [34] provide valuable mobility insights but frequently exhibit low or inconsistent sampling rates that complicate analysis. Synthetic alternatives like SynMob [49] offer uniform sampling but lack the regional diversity and quality variations essential for robust model development. Proprietary collections including GAIA [8] and Grab-Posisi [15] contain high-quality data but remain largely inaccessible due to regulatory and commercial constraints. These limitations—geographic constraints, sampling irregularities, and access restrictions—collectively impede the development of universal trajectory models. The community urgently needs comprehensive, openly accessible datasets with global coverage to advance trajectory modeling research and enable effective model generalization.

Foundation Models. The success of foundation models in natural language processing and computer vision, exemplified by BERT [7], GPT-3 [1], and Vision Transformers [9], has demonstrated how large-scale pretraining can yield highly generalizable representations across diverse tasks. This paradigm has recently extended to time series and spatio-temporal domains, with models like TST [44], TimeFM [6], and Moirai [38] leveraging Transformer architectures to capture temporal dependencies. In spatio-temporal prediction specifically, approaches such as UniST [43], Opencity [20], and ClimaX [29] have shown promise in traffic flow and climate modeling, respectively. However, these models often remain tailored to specific tasks or regions, limiting their broader applicability. Trajectory-specific models like TrajGDM [4], BigCity [41], and TrajFM [23] address certain tasks but lack the scalability and robustness needed for cross-task or cross-region applications. While unsupervised learning approaches like MAE [13] and TimeFM [6] have proven effective for images and time series, trajectory modeling presents unique challenges that demand greater flexibility to accommodate diverse mobility patterns, geographic contexts, and sampling characteristics without extensive task-specific modifications. To summary, there remains a pressing need for trajectory foundation models that unify multiple tasks within a single framework, providing robust, transferable representations that generalize across tasks and handle data variability while maintaining computational efficiency.

3 Preliminary

Definition 1: (**Trajectory**). A trajectory represents the sequential record of movement through space over time. Formally, a trajectory $\boldsymbol{\tau}$ of length n is expressed as a sequence of continuously sampled GPS points: $\boldsymbol{\tau} = \{p_1,\ p_2,\ \ldots,\ p_n\}$, where each point $p_i = \langle \text{lng}_i, \text{lat}_i, t_i \rangle$ denotes the spatial coordinates (longitude and latitude) at timestamp t_i. The *sampling interval* between consecutive points is defined as $\Delta t_i = t_i - t_{i-1}$, for $i = 2, \ldots, n$. These intervals may be uniform within or across trajectories, or vary significantly based on data collection methods and environmental factors.

Definition 2: (Trajectory Dataset). A trajectory dataset comprises multiple trajectories, each capturing the movement of an object over time. Formally, it is given by $\mathcal{D} = \{\boldsymbol{\tau}_1, \boldsymbol{\tau}_2, \ldots, \boldsymbol{\tau}_{|\mathcal{D}|}\}$, where $|\mathcal{D}|$ denotes the total number of trajectories in the dataset. These collections may vary in geographic coverage, sampling rates, and quality depending on their source and application scenario.

Problem Statement: (Universal Trajectory Modeling). Building upon the above definitions, this study aims to develop a universal foundation model for trajectory data that can adapt to diverse tasks and geographic contexts while accommodating heterogeneous data sources. Formally, consider a set

of trajectories $\mathcal{D} = \{\boldsymbol{\tau}_i\}_{i=1}^{|\mathcal{D}|}$, where each $\boldsymbol{\tau}_i$ is defined as in Definition 1. The goal is:

$$F : \boldsymbol{\tau} \mapsto \mathbf{h} \in \mathbb{R}^d, \tag{1}$$

which projects a raw trajectory $\boldsymbol{\tau}$ into a d-dimensional representation $\mathbf{h}$. This function $F(\cdot)$must capture intrinsic spatio-temporal patterns within trajectories while demonstrating three key capabilities: (1) task adaptability across various applications including classification, prediction, and anomaly detection; (2) region independence, enabling zero-shot generalization to different geographic contexts; and (3) resilience to data quality variations, effectively handling inconsistent sampling rates, varying trajectory lengths, and noise without extensive preprocessing or task-specific re-engineering.

4 Methodology

In this section, we describe the methodology for developing UniTraj, addressing the key challenges outlined in the introduction. Our approach is structured around answering three fundamental questions: (1) *How to construct a diverse and high-quality trajectory dataset that enables cross-regional generalization*? (2) *How to develop robust and scalable pre-training strategies that accommodate heterogeneous trajectory data*? and (3) *How to design an effective model architecture that adapts across diverse trajectory tasks*?

4.1 WorldTrace Dataset Construction

To address the data preparation challenge, we introduce WorldTrace, a large-scale, globally distributed trajectory dataset specifically designed to support universal trajectory modeling. Below, we introduce our data acquisition process, preprocessing pipeline, and key dataset statistics, demonstrating WorldTrace's suitability as a foundation for developing robust and generalizable trajectory foundation models. Detailed information on processing, analysis, and copyright can be found in **Appendix A**. The full dataset is available on the Hugging Face[3] and ModelScope[4] platforms.

Data Acquisition. We sourced raw trajectory data from OpenStreetMap (OSM) GPS traces [30], focusing on contributions uploaded between 2021-2023 and tagged for motorized movement to ensure data currency and relevance. This approach minimizes device heterogeneity and outdated data impacts. All collected data is stored in the standardized GPX format (an XML schema), containing latitude, longitude, timestamps, and optional metadata, providing a uniform structure that simplifies parsing and preprocessing. During acquisition, we implemented preliminary filtering to exclude trajectories with obvious anomalies such as coordinates outside valid ranges or duplicate entries.

Table 1: Summary statistics of WorldTrace.

Statistic	**Value**
Number of Trajectories	2.45 Million
Total Raw Points	8.8 Billion
Geographical Covered	70 Countries
Sampling Interval	1 sec (normalized)
Time Span	08/2021 – 12/2023
Avg. Duration	6 min
Avg. Distance	5.73 km
Avg. Speed	48.0 km/h

Data Preprocessing. Our preprocessing pipeline balances preserving authentic movement patterns with removing noise and inconsistencies, which includes the following steps:

1. **Normalization**: The original data had a high sampling frequency of up to 10 Hz, causing redundancy and increased storage demands. We therefore resampled trajectories to a uniform rate of one point per second (1 Hz), preserving essential motion details while reducing data size. In addition, by standardizing trajectories to 1s/point, we can perform better resampling during subsequent model training to accommodate frequency inconsist issues.
2. **Filtering**: We discarded trajectories with fewer than 32 points or covering distances below 100 meters, as such short trips often lack meaningful patterns and introduce noise. Following established practices [5], we also removed trajectories containing implausible speeds (e.g., exceeding 120 km/h), typically caused by GPS errors or anomalies. We also apply distance- and loop-based outlier detection to identify and remove trajectories that deviate markedly from the expected path.

[3]https://huggingface.co/datasets/OpenTrace/WorldTrace
[4]https://modelscope.cn/datasets/OpenTrace/WorldTrace

3. **Calibration**: Given that GPS signals can suffer from errors due to building obstructions, multipath effects, and receiver noise [14], we applied map-matching techniques [40] to align raw GPS points with underlying road networks. This calibration step is common practice in trajectory data processing and is widely used in data collection and related research to correct positioning errors [8, 37], improve spatial accuracy, and make trajectory analysis more reliable.

Data Analysis and Statistics. After acquiring and preprocessing the raw trajectory data, we conducted an in-depth analysis to examine the characteristics and quality of the WorldTrace dataset. Table 1 summarizes key statistics of WorldTrace. Overall, the dataset contains approximately 2.45 million trajectories and 8.8 billion raw GPS points, covering 70 countries across all inhabited continents. The data spans August 2021 to December 2023, with an average trajectory duration of about six minutes (with normalized to a 1-second sampling interval), an average distance of 5.73 km, and an average speed of 48.0 km/h. The number of points per trajectory ranges from 32 to more than 600, averaging around 358 points. Collectively, these attributes confirm WorldTrace's suitability for developing universal trajectory models that can address varied spatiotemporal patterns and broad geographical contexts.

4.2 Pre-Training Strategies

Having established a diverse trajectory dataset, we develop robust pre-training strategies to learn robust and transferable spatio-temporal representations. Rather than relying on task-specific supervision, we leverage unannotated trajectory data to capture both local and global movement patterns. To address the heterogeneous data quality challenges (varying sampling rates, differing lengths, and missing points) posed by real-world trajectory, we propose two strategies tailored specifically for trajectory: *Adaptive Trajectory Resampling* and *Self-supervised Trajectory Masking*. Due to space limitations, more details and analysis about pre-training strategies can be found in **Appendix** B.

Adaptive Trajectory Resampling (ATR). Real-world trajectory data often exhibits inconsistent sampling intervals and lengths due to diverse collection standards, device capabilities, and user behaviors. Such discrepancies challenge model generalization, as features learned under one sampling regime may not transfer to another. Inspired by common practice of multi-scale representation learning, ATR strategy addresses these issues through two complementary components:

- **Dynamic Multi-Scale Resampling**. This approach dynamically adjusts sampling frequency based on trajectory length, ensuring shorter trajectories retain fine-grained detail while longer ones are efficiently compressed. Specifically, we design a logarithmic resampling function $R(n)$ to implement this strategy:

$$R(n) = R_{\text{min}} + (1 - R_{\text{min}}) \cdot \frac{\ln(n - n_{\text{min}} + 1)}{\ln(n_{\text{max}} - n_{\text{min}} + 1)}, \tag{2}$$

  where n_{min} and n_{max} define thresholds for trajectory lengths considered "short" or "long", and R_{min} is the minimum sampling ratio. This logarithmic function creates a smooth transition in sampling density ($n_{\text{min}} < n < n_{\text{max}}$), providing three key benefits: (1) preserving critical motion patterns across trajectory lengths, (2) reducing overfitting by limiting redundancy in densely sampled data, and (3) exposing the model to diverse temporal resolutions during training.

- **Interval Consistent Resampling**. This component focuses on the sampling rate, imposing a uniform time interval Δt between consecutive points within each track:

$$\boldsymbol{\tau}' = \{\, p_{k_j} \mid k_j = 1 + (j-1)\Delta t,\ j = 1, 2, \ldots, m \}. \tag{3}$$

  By ensuring consistent spacing, this approach simplifies downstream modeling by creating regular temporal structures that make time-dependent patterns easier to learn, while mitigating complications from missing data or irregular sampling.

Combining these approaches, ATR enables models to learn representations that generalize across varying sampling rates and trajectory lengths (analysis presented in **Appendix B.1**), which is a critical capability for universal trajectory modeling.

Self-supervised Trajectory Masking (STM). Trajectory data is often incomplete or irregular due to device limitations, communication failures, and environmental factors. Motivated by masked auto-encoding methods from visual and language models, we introduce a tailored self-supervised

trajectory masking strategy, in which part of the input trajectory is hidden, forcing the model to infer local and global dependencies. Given a resampled trajectory $\boldsymbol{\tau}' = \{p_1, p_2, \ldots, p_n\}$, we define a masking function $\mathcal{M}(\boldsymbol{\tau}', r)$ that replaces a fraction r of points with a [MASK] tokens:

$$\tilde{\boldsymbol{\tau}} = \mathcal{M}(\boldsymbol{\tau}', r) = \{p_1, \ldots, [\text{MASK}]_{i \in \mathbf{I}}, \ldots, p_n\}, \tag{4}$$

where $\mathbf{I} \subseteq \{1, 2, \ldots, n\}$ and $r = |\mathbf{I}|/n$. To comprehensively address different data incompleteness scenarios, (see **Appendix** B.2 for details) we employ four complementary masking strategies:

- **Random Masking:** Uniformly samples points to mask ($\mathbf{I}_{\text{rand}} \sim \text{Uniform}(\{1, 2, \ldots, n\})$), forcing the model to infer both short-range and long-range dependencies. By forcing the reconstruction of randomly omitted points, the approach enhances the model's ability to generalize to diverse gaps.
- **Block Masking:** Conceals consecutive points ($\mathbf{I}_{\text{block}} = \{k, k+1, \ldots, k+b-1\}$) to simulate sensor failures, encouraging reconstruction of continuous segments. This approach prompts the model to utilize surrounding context for reconstructing entire missing segments, encouraging it to capture longer-range dependencies.
- **Key Points Masking:** Identifies and masks critical turning points using the Ramer-Douglas-Peucker algorithm [10]: $\mathbf{I}_{\text{key}} = \{p_k \mid d_{\max}(p_k, \overline{p_1 p_n}) > \epsilon\}$ ($d_{\max}(\cdot)$ is the maximum perpendicular distance between point p_k and line $\overline{p_1 p_n}$, ϵ is the threshold). This focuses learning on structurally significant points (sharp turns or notable speed changes) that define the trajectory's shape.
- **Last N Masking:** Masks final trajectory points ($\mathbf{I}_{\text{last}} = \{n-N+1, n-N+2, \ldots, n\}$). This setting emulates real-world forecasting tasks where future data is unavailable and must be inferred from historical observations, making it particularly effective for prediction scenarios.

4.3 Universal Trajectory Modeling

To effectively leverage the diverse trajectory data and robust pre-training strategies described above, we need to design a model architecture that can capture local and global patterns while freeing itself from regional and task-specific constraints. Our motivation for adopting this structure design is as follows: (1) We need an architecture that can be generalized to a wide range of tasks without extensive restructuring. Therefore, we adopted minimal trajectory data information (latitude, longitude, and timestamp) and ignored other region-bound information such as POI and geographical context. (2) This structure uses the reconstruction of missing points in partial observations as a proxy task and can inherit the masking strategy introduced earlier. (3) The separation of encoding and decoding enables flexible application to various downstream tasks through transfer learning or fine-tuning. More details about the architecture and parameters can be found in **Appendix** C.

Building Trajectory Embedding. Effective trajectory modeling requires transforming raw spatial and temporal data into structured embeddings that capture both local and global movement patterns. To ensure the generality of the model, we only use the latitude, longitude, and time information of the trajectory, and embed the spatial and temporal components separately to form a unified representation. For the spatial component, we normalize trajectory and map them into a d-dimensional space using a 1D convolutional, yielding a spatial embedding $\boldsymbol{h}_i^{\text{s}}$. Similarly, the temporal component, based on the time intervals Δt_i, is embedded into the same d-dimensional space via a linear layer, resulting in a temporal embedding $\boldsymbol{h}_i^{\text{t}}$. This decoupled design enables the model to effectively learn relative movement and temporal dependencies, and also cope with situations where one component may be absent. Beyond point-wise embedding, modeling the relationships between trajectory points is critical for understanding movement patterns. We adopt Rotary Position Encoding (RoPE) [32], which applies rotational transformations in the embedding space. The advantage of RoPE is its ability to preserve relative positional relationships while allowing for flexible encoding of spatial-temporal patterns across varying trajectory scales.

Adaptive Representation Learning. Based on the trajectory embeddings, we use a encoder-decoder architecture with RoPE-enhanced attention mechanism to adaptively learn a general representation of trajectories. The encoder processes the visible points in a trajectory those that are unmasked during training. Given a masked trajectory $\tilde{\boldsymbol{\tau}} = \{p_1, \ldots, [\text{MASK}]_{i \in \mathbf{I}}, \ldots, p_n\}$, we first extract the embedding representations of the unmasked points $\mathbf{H} = \{\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_m\}$ (where $m \leq n$ and $i \notin \mathbf{I}$) through the embedding steps. The encoder, denoted as $\mathbf{E}_\theta$, processes these visible embeddings to generate latent representations: $\boldsymbol{z}_{\text{enc}} = \mathbf{E}_\theta(\mathbf{H})$. The decoder reconstructs masked trajectory points based on the latent embeddings produced by the encoder. It receives the visible embeddings and mask

Table 2: Performance comparison of UniTraj with trajectory recovery tasks. The results are reported in MAE and RMSE with meters. Bold denotes the best results and underline denotes the second-best results.

Methods	WorldTrace		Chengdu		Xi'an		GeoLife		Grab-Posisi		Porto	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Linear	427.68	516.15	205.74	258.52	176.49	220.87	196.85	249.76	507.41	617.28	396.61	482.39
DHTR	220.35	302.47	75.19	98.68	62.85	83.43	80.04	168.25	351.20	415.16	194.37	232.59
Transformer	130.82	147.62	55.23	62.85	45.85	51.96	94.68	113.77	136.58	163.29	104.36	126.96
DeepMove	51.16	62.29	29.32	39.02	27.31	35.67	86.38	107.78	126.93	168.07	136.66	174.96
TrajBERT	58.13	70.14	26.48	33.83	19.45	25.13	34.53	43.24	112.68	136.24	78.77	99.23
TrajFM	47.64	58.92	19.10	25.09	18.86	24.13	59.34	64.24	107.64	130.69	71.15	92.96
UniTraj (zero-shot)	10.22	13.56	11.98	20.94	8.93	13.83	37.21	63.89	114.07	167.01	78.28	100.14
Improvement(%)	↑78.55	↑76.99	↑37.28	↑16.54	↑52.65	↑42.69	↓7.76	↓47.46	↓5.97	↓27.79	↓10.02	↓7.72
UniTraj (fine-tune)	**6.94**	**9.67**	**6.92**	**10.41**	**6.50**	**9.93**	**23.23**	**34.70**	**48.95**	**69.23**	**60.18**	**79.76**
Improvement(%)	↑85.43	↑83.59	↑63.77	↑58.51	↑65.54	↑58.85	↑32.73	↑19.75	↑54.52	↑47.03	↑15.42	↑14.20

tokens, which are initialized as learnable vectors representing missing positions. The full sequence is created by merging the encoded visible embeddings with the mask tokens, preserving the original structure of the trajectory:

$$\mathbf{z}_{\text{dec}} = \text{Reorder}\left(\left\{ \begin{array}{ll} \boldsymbol{z}_i = \boldsymbol{z}_{\text{enc},j} & \text{if } i = \text{Index}(j),\ i \notin \mathbf{I} \\ [\text{MASK}] & \text{if } i \in \mathbf{I} \end{array} \right\}\right), \tag{5}$$

where $\boldsymbol{z}_{\text{enc},j}$ corresponds to the j-th encoder output. The decoder then processes the reordered sequence to predict the missing trajectory points: $\hat{\boldsymbol{\tau}} = \text{Linear}(\mathbf{D}_\phi(\mathbf{z}_{\text{dec}}))$. The model is trained to minimize the reconstruction loss between the predicted and original points at the masked positions:

$$\mathcal{L} = \frac{1}{|\mathbf{I}|} \sum_{i \in \mathbf{I}} \left\| f_{\theta,\phi}(\tilde{\boldsymbol{\tau}})_i - \boldsymbol{\tau}_i \right\|^2, \tag{6}$$

where $f_{\theta,\phi}(\tilde{\boldsymbol{\tau}})$ represents the encoder-decoder network, and i refers to the masked positions.

5 Experiments

5.1 Experimental Setups

Datasets. We evaluate UniTraj on six diverse real-world trajectory datasets representing different collection scenarios, quality levels, motion patterns, and geographic regions. These include WorldTrace, Chengdu, Xi'an, GeoLife, Grab-Posisi, and Porto. Detailed summary are provided in **Appendix** D.1.

5.2 Task Applicability Analysis

We explore the applicability and generalizability of UniTraj to various data and downstream tasks, e.g., trajectory recovery, prediction, classification, and generation tasks. Due to space constraints, we provide the detailed setup and the results of generation task in **Appendix** D.2. It is important to clarify that our work aims to develop a general-purpose trajectory foundation model that generalizes across diverse geographic regions without region-specific dependencies, validating its effectiveness as a backbone supporting real-world trajectory applications across geographical contexts. Existing trajectory representation learning methods inherently rely on region-bound information (POIs, road networks, etc.)[16, 22, 26, 48], which contradicts our initial goal of region-independent modeling. UniTraj extracts meaningful representations solely from trajectory points without requiring auxiliary geographic context. Therefore, we deliberately excluded these methods from our baseline comparison as their architectural dependency on regional knowledge fundamentally diverges from our objective of developing a globally deployable model.

Trajectory Recovery. Table 2 presents a comprehensive comparison of UniTraj against established baselines across six datasets, revealing patterns that illuminate fundamental capabilities in trajectory reconstruction. The performance disparity between UniTraj and previous methods is particularly pronounced in geographically diverse and quality-variable datasets, where it demonstrates substantial resilience to regional variations. In the zero-shot setting, UniTraj achieves remarkable results, confirming it effectively captures transferable spatio-temporal patterns without requiring additional fine-tuning. The performance difference becomes particularly instructive when analyzing low-quality datasets like GeoLife and Grab-Posisi, with their highly irregular sampling intervals and multiple

travel modes. It demonstrate the effectiveness of our adaptive resampling strategy in handling temporal heterogeneity. The Chengdu and Xi'an datasets reveal another critical aspect of UniTraj's capabilities, models trained on high-quality data exhibit reliable transferability and achieve optimal results even in zero-shot scenarios. When fine-tuned, UniTraj achieves the lowest error scores across all datasets, demonstrating UniTraj's superior generalizability across diverse geographic regions. For instance, on GeoLife, UniTraj's fine-tuned performance (MAE 23.23) reduces error by 32.73% compared to TrajBERT, showcasing its effectiveness with complex travel patterns and lower-quality data. These results validate WorldTrace's potential as a foundation dataset and UniTraj's consistent superiority in trajectory recovery tasks, with substantial improvements through fine-tuning reinforcing its adaptability and robustness.

Trajectory Prediction. Table 3 shows UniTraj's exceptional performance in trajectory prediction, a different task requiring forward inference rather than reconstruction. The zero-shot results merit particular attention, as they represent the most challenging scenario for trajectory models. On WorldTrace, UniTraj's zero-shot MAE significantly outperforms all baselines, underscoring the model's versatility in capturing universal motion patterns. When fine-tuned, the performance further improves, consistently achieving the best results across all evaluated datasets. This generalization capability stems from our Last-N masking strategy, which explicitly shapes the embedding space to support predictive inference. These results further confirm that UniTraj not only generalizes remarkably well across diverse datasets but also benefits considerably from fine-tuning, making it highly adaptable for real-world applications requiring accurate trajectory predictions.

Table 3: Performance comparison of UniTraj with trajectory prediction tasks.

Methods	WorldTrace		Chengdu		GeoLife	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
Linear	153.12	159.65	156.85	164.58	189.02	201.34
DHTR	146.48	151.63	123.47	129.73	180.32	187.59
Transformer	114.25	117.07	67.38	70.86	165.02	170.84
DeepMove	55.69	58.67	36.31	39.10	116.46	123.20
TrajBERT	80.57	86.36	64.73	68.92	113.68	121.18
TrajFM	75.45	81.32	77.82	80.48	121.94	128.16
UniTraj (zero-shot)	49.85	55.02	42.75	45.93	108.35	133.60
Improvement(%)	↑10.49	↑6.22	↓17.74	↓17.46	↑4.69	↓10.25
UniTraj (fine-tune)	**30.10**	**34.46**	**28.78**	**32.44**	**90.97**	**102.88**
Improvement(%)	↑45.95	↑41.27	↑20.74	↑17.03	↑19.98	↑15.10

Trajectory Classification. Figure 2 presents classification accuracy results that reveal UniTraj's capacity to learn discriminative representations of movement modalities. Notably, even without fine-tuning, UniTraj achieves 71.3% accuracy on GeoLife, outperforming several supervised baselines. This zero-shot performance demonstrates that the pre-trained representations inherently capture transportation mode signatures, where movement modality emerges as a natural organizing principle. On the Grab-Posisi dataset, which presents additional challenges due to similar motion patterns for mixed travel modes (car and motorcycle). UniTraj achieves 79.3% accuracy after fine-tuning with a substantial improvement over the best baseline. This improvement emphasizes UniTraj's ability to capture subtle kinematic signatures that differentiate travel modes with complex or similar patterns.

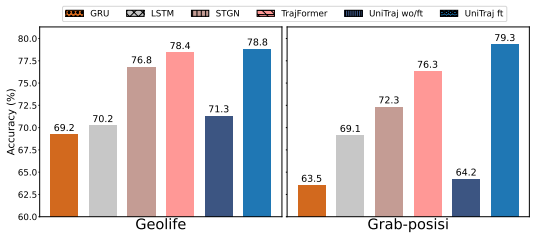


Figure 2: Performance comparison of classification task with GeoLife and Grab-posisi dataset.

5.3 Dataset Study

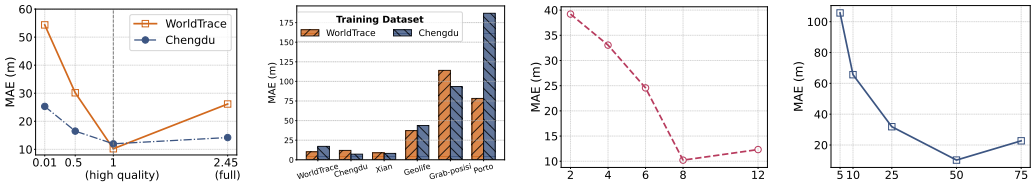
This section analyze the impact of dataset scale, quality, and diversity on model performance of UniTraj, particularly its generalization capability across different data sources. We focus on two main experiments: (1) examining the effect of dataset scale and quality within WorldTrace, with varying data volumes ($\{0.01,\ 0.5,\ 1\}$ millions) and a high-quality (obtained by further removing loops, staying dense trajectories) subset, and (2) assessing UniTraj's adaptability and effectiveness by training it on these datasets beyond WorldTrace, thus showing its potential as a foundation model.

Effect of Dataset Scale and Quality. Figure 3(a) illustrates the relationship between training data volume and model performance, revealing a phenomenon that goes beyond simple scaling laws. With increasing trajectory count from WorldTrace (from 0.5M to 2.45M), the MAE on the in-domain test set decreases dramatically, showing substantial improvement up to approximately 1M trajectories before beginning to exhibit diminishing returns. The above result indicates that larger datasets enable

Table 4: Ablation study on different resampling and masking strategies on six datasets.

Methods	WorldTrace		Chengdu		Xi'an		GeoLife		Grab-posisi		Porto	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
w/o Dynamic Multi-scale resampl.	426.80	482.37	192.54	272.42	157.85	223.96	499.95	671.69	1933.28	2504.16	93.14	119.93
w/o Interval Consistent resampl.	21.30	24.76	12.98	20.61	9.34	13.90	69.41	115.33	102.45	149.60	1724.12	2016.61
w/o Key points masking	25.49	28.91	14.46	21.98	11.10	15.17	45.94	72.84	113.65	162.57	**76.51**	101.18
w/o Block masking	**7.79**	**10.47**	**9.22**	**15.36**	**7.16**	**11.18**	48.59	77.73	**89.34**	**128.72**	198.41	238.88
UniTraj	10.22	13.56	11.98	20.94	8.93	13.83	**37.21**	**63.89**	114.07	167.01	78.28	**100.14**

the model to capture a wider range of spatio-temporal patterns. However, while increasing the dataset size from 1 million to 2.45 million trajectories results in better coverage, the model's MAE slightly increases due to the introduction of more noise in the full dataset. In contrast, training on a high-quality subset of 1 million trajectories, which includes curated, noise-free data, yields more reliable and consistent learning. This highlights the importance of both dataset scale and quality, with quality being especially crucial when data volume is limited.



(a) Training data volume. (b) Training dataset. (c) Number of Encoders (d) Mask ratio (%).

Figure 3: The effect of amount of data volume, diversity dataset, and different parameter settings.

Effect of Dataset Diversity. In Figure 3(b), we compare UniTraj 's zero-shot performance when trained on WorldTrace and Chengdu (the highest quality dataset available), evaluated across multiple real-world datasets. Models trained on WorldTrace exhibit superior generalization across diverse datasets (e.g., GeoLife and Porto), reflecting the broad geographic and contextual coverage of WorldTrace. Conversely, models trained on Chengdu perform best on datasets with similar density and travel modes, such as Xi'an. However, proprietary datasets like Chengdu, while offering high quality, are not publicly available, limiting their applicability for universal tasks. These results demonstrate UniTraj's robustness and adaptability, validating WorldTrace as an ideal training resource for building a universal trajectory foundation model. At the same time, the findings confirm that UniTraj can effectively leverage other datasets when necessary, further enhancing its versatility.

5.4 Model Study

We investigate architectural components, parameter settings, and pre-training strategies to assess sensitivity to parameter choices and the contributions of their core components.

Effect of Parameter Settings. Figure 3(c) Figure 3(d) and presents the results of our parameter sensitivity analysis, examining how the number of encoder blocks and the masking ratio influence model performance. As shown in Figure 3(c), increasing the number of encoder blocks from 2 to 8 significantly reduces MAE, with performance plateauing beyond 8 blocks. This plateau suggests that while deeper architectures can improve model capacity, the benefits diminish without corresponding adjustments in data or hyperparameters [17]. Figure 3(d) demonstrates that a masking ratio of 50% yields the best performance. Low masking ratios (e.g., 5%-10%) result in higher MAE due to insufficient training signal, while higher ratios (e.g., 75%) lead to increased MAE from excessive information loss. A 50% masking ratio strikes a balance, providing the model with a strong training signal without sacrificing the context needed for effective trajectory reconstruction.

Ablation Study. Table 4 presents an ablation study, showing how different pre-training strategies affect UniTraj's performance across datasets. The performance varies across datasets, indicating the effectiveness and limitations of them depending on the specific data and task scenarios. *Dynamic Multi-scale Resampling* significantly improves performance across most datasets, especially GeoLife and Grab-Posisi, which have inconsistent sampling intervals and lower data quality. This suggests that dynamic resampling helps the model to adapt to heterogeneous dataset scenarios and to be adaptive for information preservation (see **Appendix B.1.1** for more details). The *Interval Consistent Resampling* has a notable positive effect on datasets with consistent sampling rates, such as Porto

and WorldTrace. It indicates that the integration of this strategy strategy can effectively separate the temporal sampling pattern from the region, it enhances the generalization of the model to data sets with different sampling rates (analysis presented in **Appendix B.1.2**). *Key Points Masking* leads to substantial performance drops on high-quality datasets like Chengdu and Xi'an but appears to offer minimal benefits, or even slight disadvantages, for certain datasets. This finding suggests that adjusting adaptive masking strategies based on trajectory complexity, potentially applying it selectively to trajectories with significant directional changes, while using alternative strategies for smoother paths. *Block Masking* shows significant effects on GeoLife and Porto, where it helps the model handle low sampling frequencies. However, its impact on other datasets is more inconsistent, suggesting that it introduces an artificial challenge that may increase complexity in high-frequency datasets. (we provide a robustness analysis in **Appendix B.2**) Overall, the varying impact of UniTraj's pre-training strategies across datasets highlights its adaptability to different tasks and scenarios. While not all of them universally enhance performance, their combined use provides a balanced training strategy, allowing for flexible configuration depending on specific dataset requirements. Fine-tuning further optimizes performance, ensuring stability and robustness across diverse tasks.

6 Conclusion

In this work, we presented UniTraj, a universal trajectory foundation model designed to overcome the task specificity, regional dependency, and data quality limitations of current approaches. UniTraj acts as a robust backbone that generalizes effectively across diverse tasks and regions. To support its development, we introduced WorldTrace, a high-quality global dataset with 2.45 million trajectories from 70 countries, offering broad geographic coverage, varied sampling rates, and open accessibility. Together, UniTraj and WorldTrace provide a versatile, high-performing foundation for trajectory analysis, paving the new solution for more adaptable and efficient models in trajectory-based research. Future work will focus on expanding the geographic and modal diversity of the WorldTrace dataset to better cover underrepresented regions and non-motorized travel. We also aim to enhance the UniTraj model by integrating contextual information, such as road networks and points of interest, to improve its predictive accuracy and real-world applicability. Further optimizations to the model architecture and pre-training strategies will also be explored to boost performance and efficiency.

7 Acknowledgement

This work was mainly supported by the National Natural Science Foundation of China under Grant No. 62506097, No. 62402414 and No.62502404. This work is also supported by the Guangdong Basic and Applied Basic Research Foundation (No. 2025A1515011994), Tencent (CCF-Tencent Open Fund, Tencent Rhino-Bird Focused Research Program), Didi (CCF-DiDi GAIA Collaborative Research Funds), Guangzhou Municipal Science and Technology Project (No. 2023A03J0011) and Guangzhou-HKUST(GZ) Joint Funding Program (No. 2024A03J0620), Hong Kong Research Grants Council's Research Impact Fund (No.R1015-23), Research Grants Council's Collaborative Research Fund (No.C1043-24GF), Research Grants Council's General Research Fund (No.11218325), Institute of Digital Medicine of City University of Hong Kong (No.9229503), Huawei (Huawei Innovation Research Program), 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), Didi (CCF-Didi Gaia Scholars Research Fund), Kuaishou, and Bytedance.

References

[1] T. B. Brown. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[2] Y. Chang, E. Tanin, G. Cong, C. S. Jensen, and J. Qi. Trajectory similarity measurement: An efficiency perspective. *arXiv preprint arXiv:2311.00960*, 2023.

[3] W. Chen, Y. Liang, Y. Zhu, Y. Chang, K. Luo, H. Wen, L. Li, Y. Yu, Q. Wen, C. Chen, et al. Deep learning for trajectory data management and mining: A survey and beyond. *arXiv preprint arXiv:2403.14151*, 2024.

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

[5] S. Dabiri and K. Heaslip. Inferring transportation modes from gps trajectories using a convolutional neural network. *Transportation research part C: emerging technologies*, 86:360–371, 2018.

[6] A. Das, W. Kong, R. Sen, and Y. Zhou. A decoder-only foundation model for time-series forecasting. *arXiv preprint arXiv:2310.10688*, 2023.

[7] J. Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[8] Didi Chuxing. Gaia open datasets., 2018.

[9] A. Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[10] D. H. Douglas and T. K. Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: the international journal for geographic information and geovisualization*, 10(2):112–122, 1973.

[11] 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*, pages 1459–1468, 2018.

[12] C. Guo, B. Yang, J. Hu, and C. Jensen. Learning to route with sparse trajectory sets. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, pages 1073–1084. IEEE, 2018.

[13] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[14] B. Hofmann-Wellenhof, H. Lichtenegger, and E. Wasle. *GNSS–global navigation satellite systems: GPS, GLONASS, Galileo, and more*. Springer Science & Business Media, 2007.

[15] X. Huang, Y. Yin, S. Lim, G. Wang, B. Hu, J. Varadarajan, S. Zheng, A. Bulusu, and R. Zimmermann. Grab-posisi: An extensive real-life gps trajectory dataset in southeast asia. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Prediction of Human Mobility*, page 1–10, 2019.

[16] J. Jiang, D. Pan, H. Ren, X. Jiang, C. Li, and J. Wang. Self-supervised trajectory representation learning with temporal regularities and travel semantics. In *2023 IEEE 39th international conference on data engineering (ICDE)*, pages 843–855. IEEE, 2023.

[17] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[18] H. Lan, J. Xie, Z. Bao, F. Li, W. Tian, F. Wang, S. Wang, and A. Zhang. Vre: a versatile, robust, and economical trajectory data system. *Proceedings of the VLDB Endowment*, 15(12):3398–3410, 2022.

[19] L. Li, R. Jiang, Z. He, X. M. Chen, and X. Zhou. Trajectory data-based traffic flow studies: A revisit. *Transportation Research Part C: Emerging Technologies*, 114:225–240, 2020.

[20] Z. Li, L. Xia, L. Shi, Y. Xu, D. Yin, and C. Huang. Opencity: Open spatio-temporal foundation models for traffic prediction. *arXiv preprint arXiv:2408.10269*, 2024.

[21] Y. Liang, K. Ouyang, Y. Wang, X. Liu, H. Chen, J. Zhang, Y. Zheng, and R. Zimmermann. Trajformer: Efficient trajectory classification with transformers. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 1229–1237, 2022.

[22] Y. Lin, H. Wan, S. Guo, J. Hu, C. S. Jensen, and Y. Lin. Pre-training general trajectory embeddings with maximum multi-view entropy coding. *IEEE Transactions on Knowledge and Data Engineering*, 36(12):9037–9050, 2023.

[23] Y. Lin, T. Wei, Z. Zhou, H. Wen, J. Hu, S. Guo, Y. Lin, and H. Wan. Trajfm: A vehicle trajectory foundation model for region and task transferability. *arXiv:2408.15251*, 2024.

[24] Y. Lin, Z. Zhou, Y. Liu, H. Lv, H. Wen, T. Li, Y. Li, C. S. Jensen, S. Guo, Y. Lin, et al. Unite: A survey and unified pipeline for pre-training st trajectory embeddings. *arXiv e-prints*, pages arXiv–2407, 2024.

[25] M. Luca, G. Barlacchi, B. Lepri, and L. Pappalardo. A survey on deep learning for human mobility. *ACM Computing Surveys (CSUR)*, 55(1):1–44, 2021.

[26] Z. Ma, Z. Tu, X. Chen, Y. Zhang, D. Xia, G. Zhou, Y. Chen, Y. Zheng, and J. Gong. More than routing: Joint gps and route modeling for refine trajectory representation learning. In *Proceedings of the ACM Web Conference 2024*, pages 3064–3075, 2024.

[27] G. Mai, W. Huang, J. Sun, S. Song, D. Mishra, N. Liu, S. Gao, T. Liu, G. Cong, Y. Hu, et al. On the opportunities and challenges of foundation models for geospatial artificial intelligence. *arXiv preprint arXiv:2304.06798*, 2023.

[28] W. K. Meghan O'Connell, moreiraMatias. Ecml/pkdd 15: Taxi trajectory prediction (i), 2015.

[29] T. Nguyen, J. Brandstetter, A. Kapoor, J. K. Gupta, and A. Grover. Climax: A foundation model for weather and climate. *arXiv preprint arXiv:2301.10343*, 2023.

[30] OpenStreetMap Contributors. Openstreetmap, 2024.

[31] J. Si, J. Yang, Y. Xiang, H. Wang, L. Li, R. Zhang, B. Tu, and X. Chen. Trajbert: Bert-based trajectory recovery with spatial-temporal refinement for implicit sparse trajectories. *IEEE Transactions on Mobile Computing*, 2023.

[32] J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, and Y. Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[33] 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*, 30, 2017.

[34] G. Wang, X. Chen, F. Zhang, Y. Wang, and D. Zhang. Experience: Understanding long-term evolving patterns of shared electric vehicle networks. In *The 25th Annual international conference on mobile computing and networking*, pages 1–12, 2019.

[35] J. Wang, N. Wu, X. Lu, W. X. Zhao, and K. Feng. Deep trajectory recovery with fine-grained calibration using kalman filter. *IEEE Transactions on Knowledge and Data Engineering*, 33(3):921–934, 2019.

[36] S. Wang, Z. Bao, J. S. Culpepper, and G. Cong. A survey on trajectory data management, analytics, and learning. *ACM Computing Surveys (CSUR)*, 54(2):1–36, 2021.

[37] Z. Wang, K. Fu, and J. Ye. Learning to estimate the travel time. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, page 858–866, London, United Kingdom, 2018. Association for Computing Machinery.

[38] G. Woo, C. Liu, A. Kumar, C. Xiong, S. Savarese, and D. Sahoo. Unified training of universal time series forecasting transformers. *arXiv preprint arXiv:2402.02592*, 2024.

[39] H. Yan and Y. Li. Generative ai for intelligent transportation systems: Road transportation perspective. *ACM Computing Surveys*, 2025.

[40] C. Yang and G. Gidofalvi. Fast map matching, an algorithm integrating hidden markov model with precomputation. *International Journal of Geographical Information Science*, 32(3):547 – 570, 2018.

[41] X. Yu, J. Wang, Y. Yang, Q. Huang, and K. Qu. Bigcity: A universal spatiotemporal model for unified trajectory and traffic state data analysis. *arXiv preprint arXiv:2412.00953*, 2024.

[42] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, and Y. Huang. T-drive: Driving directions based on taxi trajectories. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '10, page 99–108, New York, NY, USA, 2010. Association for Computing Machinery.

[43] Y. Yuan, J. Ding, J. Feng, D. Jin, and Y. Li. Unist: a prompt-empowered universal model for urban spatio-temporal prediction. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4095–4106, 2024.

[44] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, and C. Eickhoff. A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 2114–2124, 2021.

[45] P. Zhao, A. Luo, Y. Liu, J. Xu, Z. Li, F. Zhuang, V. S. Sheng, and X. Zhou. Where to go next: A spatio-temporal gated network for next poi recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 34(5):2512–2524, 2020.

[46] Y. Zheng, H. Fu, X. Xie, W.-Y. Ma, and Q. Li. *Geolife GPS trajectory dataset - User Guide*, July 2011.

[47] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma. Mining interesting locations and travel sequences from gps trajectories. In *Proceedings of the 18th international conference on World wide web*, pages 791–800, 2009.

[48] S. Zhou, S. Shang, L. Chen, C. S. Jensen, and P. Kalnis. Red: Effective trajectory representation learning with comprehensive information. *arXiv preprint arXiv:2411.15096*, 2024.

[49] Y. Zhu, Y. Ye, Y. Wu, X. Zhao, and J. J. Yu. Synmob: Creating high-fidelity synthetic GPS trajectory dataset for urban mobility analysis. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.

[50] 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*, pages 4676–4687, 2024.

SUPPLEMENTARY MATERIAL

UNITRAJ: LEARNING A UNIVERSAL TRAJECTORY FOUNDATION MODEL FROM BILLION-SCALE WORLDWIDE TRACES

TABLE OF CONTENTS

A Details of WorldTrace Dataset

In this section, we detail the collection of the dataset, the processing, and provide a detailed analysis of the resulting dataset.

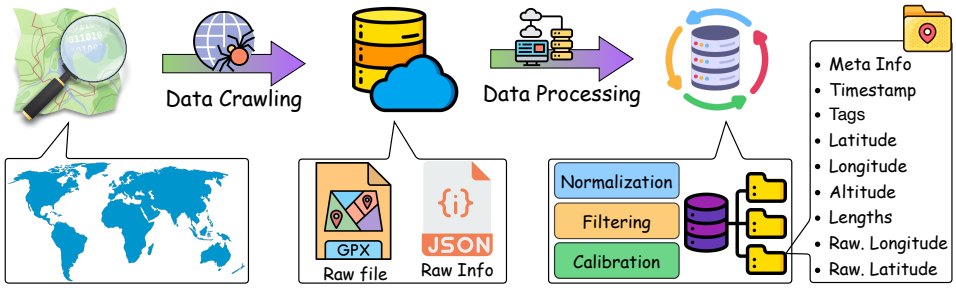


Figure 4: The process pipeline of WorldTrace dataset construction.

A.1 Data Collection

Data Source. As shown in Figure 4, the raw data for WorldTrace is sourced from the shared trajectory data platform on OpenStreetMap (OSM) [30][5]. This platform, a public sharing project, hosts over 11 million GPS trajectories uploaded by contributors worldwide from 2004 to the present. To ensure data quality and reliability, we specifically targeted contributions tagged for motorized movement to ensure data currency and relevance to modern transportation networks. This approach helps minimize device heterogeneity and avoids outdated data that might not reflect current infrastructure. The raw data is stored in the standardized GPX (GPS Exchange Format), an XML schema designed for exchanging GPS data between applications and web services https://www.topografix.com/GPX/1/1/. Each GPX file contains sequences of trackpoints with the following attributes:

- Latitude (decimal degrees)
- Longitude (decimal degrees)
- Altitude (decimal numbers)
- Timestamp (ISO 8601 format)
- Optional metadata (version, tags, etc.)

In addition, while crawling the original trajectory, we also crawled the basic information about the trajectory descriptions, such as the starting point, markers, time, creator, etc., which was saved as a JSON file.

Collection Process. Prior to integration, our collection pipeline involved the following steps: Our collection pipeline involved the following steps:

1. **API-based Retrieval**: We use the OSM API to systematically query and download GPX traces based on selected filters to ensure global coverage. In order not to increase the burden on server providers, we did not use concurrent crawling, and the whole collection process lasted about 6 months, yielding about 4.5 million raw traces.
2. **Initial Filtering**: During acquisition, we implemented preliminary filtering to exclude trajectories with obvious anomalies such as: Coordinates outside valid ranges (-90° to 90° for latitude, -180° to 180° for longitude); Duplicate or long duration consecutive points; Empty or near-empty traces (fewer than 60 seconds).
3. **Format Standardization**: All collected data was parsed from the original GPX format and converted to a unified internal format for subsequent processing.

[5]https://www.openstreetmap.org/traces

A.2 Data Processing

Our preprocessing pipeline was designed to balance preserving authentic movement patterns with removing noise and inconsistencies. The process consists of three main stages:

Normalization. The raw data exhibited highly variable sampling frequencies, ranging from sub-second intervals up to several seconds between consecutive points. This heterogeneity creates challenges for modeling and increases storage requirements unnecessarily. We therefore applied the following normalization procedures:

- **Temporal Resampling**: We resampled all trajectories to a uniform rate of one point per second (1 Hz). For segments with sampling rates higher than 1 Hz, we select the first occurrence of a trajectory point within each one-second window. For segments with lower sampling rates, we used linear interpolation between available points to estimate positions at one-second intervals.
- **Coordinate Standardization**: All coordinates were converted to the WGS84 datum for consistency, and we ensured uniform precision across the dataset (6 decimal places for both latitude and longitude, providing 0.1m precision at the equator).

Filtering. After normalization, we implemented a multi-stage filtering process to meticulously remove trajectories that were deemed unsuitable for our analysis. This comprehensive filtering approach involved several key steps:

- **Length-based Filtering**: We discarded trajectories with fewer than 32 points (equivalent to 32 seconds after resampling) or covering distances below 100 meters, as these typically represent stationary periods or very short movements with limited analytical value.
- **Speed-based Filtering**: We calculated point-to-point speeds and removed trajectories containing implausible values (e.g., exceeding 120 km/h or lower 0.5 km/h in urban environments), typically caused by GPS errors or anomalies.
- **Distance-based Outlier Detection**:We calculated the distance between the original trajectory and the map-matched trajectory. Trajectories that were too far away (indicating large deviations in motion) were flagged for further inspection or removal.
- **Loop Detection**: We identify and remove trajectories that form perfect or near-perfect loops with no apparent destination by their geometry, which usually indicates the presence of clearly anomalous patterns.

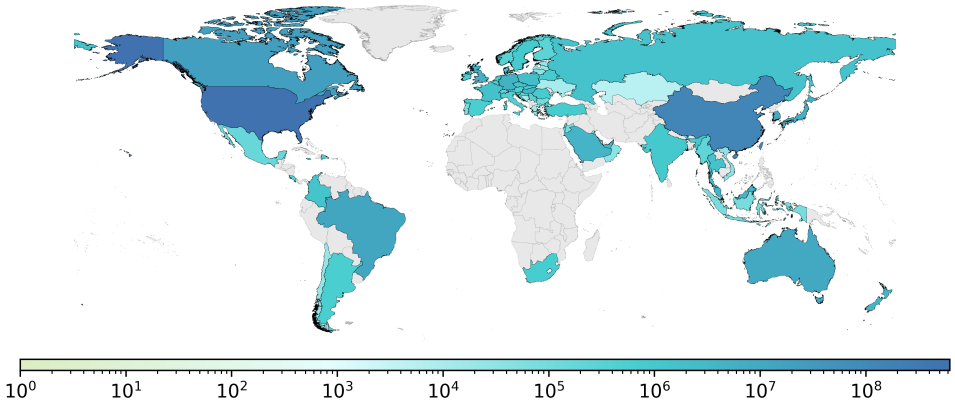
Calibration. GPS signals can suffer from various errors due to atmospheric conditions, satellite geometry, and physical obstructions. To improve data quality, we applied map-matching techniques to align raw GPS points with underlying road networks, using a Hidden Markov Model-based approach (or using online API) with a custom emission probability function that accounts for both point-to-road distance and heading consistency. Besides, each trajectory point was enriched with derived attributes.

A.3 Data Statistics and Analysis

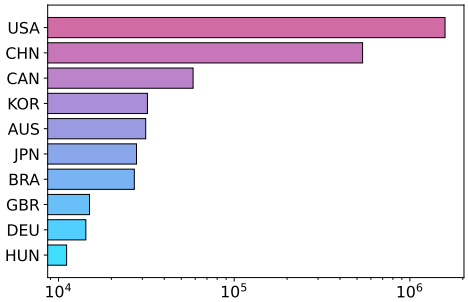
Overall Statistics. The final WorldTrace dataset contains:

- Approximately 2.45 million trajectories.
- 8.8 billion raw GPS points (before normalization).
- Coverage across 70 countries on all inhabited continents.
- Temporal span from August 2021 to December 2023.
- Average trajectory duration of approximately 6 minutes.
- Average trajectory distance of 5.73 kilometers.
- Average travel speed of 48.0 km/h.
- Points per trajectory ranging from 32 to over 600, with an average of 358 points.

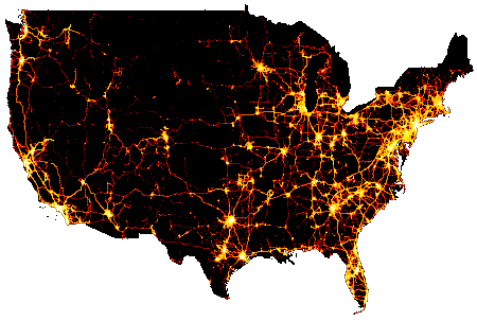
Geographic Distribution. WorldTrace offers extensive geographic coverage, as illustrated in Figure 5, encompassing trajectory data from 70 countries and spanning diverse environments and



(a) Geographic distribution across the world.



(b) Trajectory counts of top 10 countries.



(c) Distribution within contiguous USA.

Figure 5: The distribution details of WorldTrace dataset.

infrastructure types. This global distribution is visualized in Figure 5(a), highlighting dense concentrations in North America, East Asia, and parts of Europe, with trajectory counts exceeding in the most represented regions. Figure 5(b) further details the top 10 countries by trajectory counts, with the United States, China, and Canada leading in data volume. Notably, it exhibits substantial geographic diversity, with varying densities across urban, suburban, and rural environments. The top 10 countries by trajectory count, namely, the USA, China, Canada, Germany, UK, Japan, Brazil, Australia, South Korea, and Hungary, represent a wide range of urban forms, road networks, and mobility cultures. Additionally, Figure 5(c) provides a closer look at the data density within the contiguous United States, demonstrating high-resolution coverage along major road networks and urban centers. This detailed distribution underscores the dataset's ability to capture nuanced variations in trajectory data across different regions. Collectively, these figures emphasize the potential of WorldTrace to serve as a robust foundation for developing region-independent and universal trajectory models. Its extensive geographic coverage and diverse environmental representation make it well-suited for applications that require broad and adaptable trajectory data.

A.4 Data Privacy and Copyright

To protect privacy and comply with international data protection regulations, all data collection adhered strictly to privacy regulations and ethical guidelines. Trajectories were anonymized, and any personally identifiable information was excluded to protect user privacy. In addition, all raw data follows the Open Data Commons Open Database License (**ODbL**) license from OSM: http://opendatacommons.org/licenses/odbl/1.0/. We will share derived datasets under the same license terms to respect the data use policies of the community.

B Pre-training Strategies

In this section, we provide specific details on the adaptive trajectory resampling strategy and the self-supervised trajectory masking strategy, and we will provide the design motivation and theoretical analysis for these two strategies.

B.1 Adaptive Trajectory Resampling

Trajectory data heterogeneous is one of the main challenges in cross-regional and cross-device trajectory modeling. The Adaptive Trajectory Resampling strategy solves this problem through two complementary components: Dynamic Multi-Scale Resampling and Interval Consistent Resampling. We designed these two strategies with the motivation of fitting different regions and dataset qualities through diversified trajectory sampling frequencies and motion patterns. *Dynamic Multi-Scale Resampling ensures an optimal balance between information preservation and computational efficiency across different trajectory lengths, prioritizing the retention of key motion patterns. Interval Consistent Resampling enhances the model's generalization ability across datasets with different sampling rates by normalizing the time dimension.*

B.1.1 Dynamic Multi-Scale Resampling

As discussed in Section 4.2, we adopted a logarithmic resampling ratio that adjusts the sampling rate according to the trajectory length. The resampling ratio function $R(n)$ is designed to decrease logarithmically as the trajectory length n increases:

$$R(n) = \begin{cases} R_{\text{min}}, & n \geq n_{\text{max}} \\ 1 - (1 - R_{\text{min}})\phi(n), & n_{\text{min}} < n < n_{\text{max}} \\ 1, & n \leq n_{\text{min}} \end{cases} \tag{7}$$

where R_{min} is the minimum sampling ratio, and n_{min} and n_{max} denotes the shortest and longest length thresholds, respectively. The normalization factor $\phi(n)$ is computed as follows:

$$\phi(n) = \frac{\ln(n - n_{\text{min}} + 1)}{\ln(n_{\text{max}} - n_{\text{min}} + 1)}. \tag{8}$$

Formal Definition. Here, we provide a formal definition and theoretical analysis of the above empirical results through information theory and computational efficiency perspectives. For any trajectory $\boldsymbol{\tau} = \{p_1,\ p_2,\ \ldots,\ p_n\}$ consist of n spatio-temporal points. the number of points for resampled trajectory $\boldsymbol{\tau}' = \{p_1, p_2, \ldots, p_m\}$ is:

$$m = R(n) \cdot n, \tag{9}$$

where function $R(n)$ that determines what proportion of points to retain. The logarithmic sampling strategy guarantees bounded sample sizes for arbitrarily long trajectories while preserving critical minimum information content. Specifically:

- For $n \leq n_{\text{min}}$: $R(n) = 1$, so $m = n$;
- For $n \geq n_{\text{max}}$: $R(n) = R_{\text{min}}$, we set $m = m_{\max}$ as a constant. Clearly, the number of sampled points is bounded above by $m_{\max}$.

To ensure boundedness, we analyze $m(n)$ in the intermediate domain $n \in (n_{\text{min}}, n_{\text{max}})$.

$$m = \left[1 - (1 - R_{\min}) \cdot \phi(n)\right] \cdot n. \tag{10}$$

Taking derivative:

$$\frac{d(R(n) \cdot n)}{dn} = 0 \tag{11}$$

Solving this equation yields a value $n^* < n_{\max}$, ensuring that $m_{\max}$ is bounded. Since $R(n)$ becomes constant for $n \geq n_{\max}$, and m increases linearly in that region, the global maximum occurs at either n^* or $n_{\max}$. However, due to the logarithmic decay of $R(n)$, the growth of m slows, and the maximum value is achieved at a finite $n^* < n_{\max}$. Hence, m is bounded for all n.

Corollary 1: Information Preservation and Computing Efficiency Optimization

Standpoint: The logarithmic sampling function provides an optimal balance between information preservation and computational efficiency across varying trajectory lengths.

Proof: Let $I(\boldsymbol{\tau})$ represent the information content of trajectory $\boldsymbol{\tau}$. Empirical studies in spatio-temporal data analysis suggest that information content typically scales sub-linearly with trajectory length, following approximately:

$$I(\boldsymbol{\tau}) \propto n^{\alpha}, \tag{12}$$

where $0 < \alpha < 1$. For example, $\alpha \approx 0.7$ indicates that only 70% of the trajectory points contain valid feature information, and the remaining 30% are redundant. For a resampled trajectory $\boldsymbol{\tau'}$ with $m = R(n) \cdot n$ points, the information preservation ratio η can be approximated as:

$$\eta = \frac{I(\boldsymbol{\tau'})}{I(\boldsymbol{\tau})} \approx (\frac{m}{n})^{\alpha} = R(n)^{\alpha}. \tag{13}$$

The computational cost C of processing trajectory typically scales linearly with length:

$$C(\boldsymbol{\tau}) \propto n^{\beta}, \tag{14}$$

where $\beta \geq 1$, typically $\beta \approx 2$ for Transformer-based models. After resampling, the computational efficiency gain γ is:

$$\gamma = \frac{C(\boldsymbol{\tau})}{C(\boldsymbol{\tau'})} \approx (\frac{n}{m})^{\beta} = \frac{1}{R(n)^{\beta}}. \tag{15}$$

The optimal sampling function maximizes the product of information preservation and computational efficiency:

$$\max_{R(n)} \eta \cdot \gamma = \max_{R(n)} R(n)^{\alpha} \cdot \frac{1}{R(n)^{\beta}} = \max_{R(n)} R(n)^{\alpha-\beta}. \tag{16}$$

Since $\alpha < \beta$ for typical trajectory data, this is a decreasing function $R(n)$. However, we must maintain a minimum level of information, hence the constraint $R(n) \geq R_{\min}$.

When we examine the information density:

$$D(n) = \frac{I(\boldsymbol{\tau})}{n} \propto n^{\alpha-1}, \tag{17}$$

we observe that it decreases as n increases, indicating diminishing information return per point in longer trajectories. An optimal sampling ratio should proportionally track this information density:

$$R_{\text{opt}}(n) \propto D(n) \propto n^{\alpha-1}. \tag{18}$$

Our logarithmic resampling function's derivative in the intermediate domain ($n_{\min} < n < n_{\max}$) is:

$$\frac{dR(n)}{dn} = -\frac{1 - R_{\min}}{\ln(n_{\max} - n_{\min} + 1)} \cdot \frac{1}{n - n_{\min} + 1} \propto \frac{1}{n} \tag{19}$$

As n increases, the growth rate of the logarithmic function slows down, causing the rate at which the sampling rate $R(n)$ decreases to also slow down. This is closely proportional to the derivative of the theoretical optimal sampling rate:

$$\frac{dR_{\text{opt}}(n)}{dn} \propto (\alpha - 1)n^{\alpha-2} \propto \frac{1}{n^{2-\alpha}}. \tag{20}$$

For example, when $\alpha \approx 0.7$, we have $\frac{dR_{\text{opt}}(n)}{dn} \propto \frac{1}{n^{1.3}}$ This property of logarithmic functions (their rate of change is inversely proportional to the input value), making them naturally suited to this task. Therefore, Logarithmic resampling provides a theoretically reasonable compromise: it preserves almost all of the information from short trajectories (where every point may be significant) while reducing redundancy in long trajectories (where redundancy is highest). Compared to linear functions, logarithmic functions can more naturally adapt to the information density curve across the entire trajectory length range.

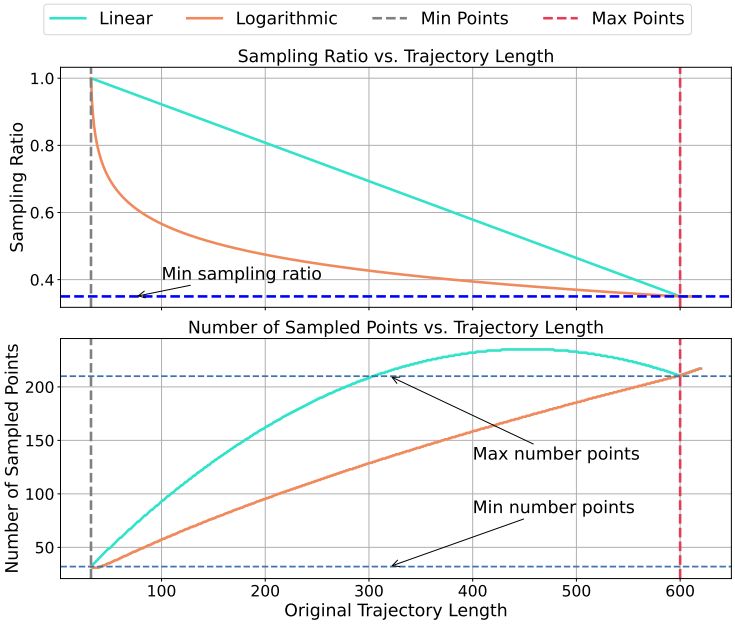


Figure 6: Illustration of the difference between dynamic resampling strategies with linear method.

Visualization. As shown in Figure 6, we compare in detail the proposed dynamic resampling strategy with a linear resampling strategy (where the sampling ratio $R(n)$ decreases linearly with the length of the trajectory) regarding the sampling ratio and the sampled points. Specifically, this figure illustrates the dynamic resampling strategy compared to a linear resampling approach. The top plot displays how the sampling ratio $R(n)$ decreases with trajectory length n. The dynamic strategy (orange curve) follows a logarithmic decrease, ensuring a smoother transition from retaining all points for short trajectories ($n \leq n_{\text{min}}$) to reducing redundancy for long trajectories ($n \geq n_{\text{max}}$), with a minimum sampling ratio R_{min}. In contrast, the linear resampling strategy (blue curve) decreases the sampling ratio at a constant rate. The bottom plot shows the relationship between the number of sampled points and trajectory length for both strategies. The dynamic approach adjusts sampling more gradually, preserving detail for intermediate trajectories while minimizing redundancy in longer trajectories. However, linear sampling methods instead suffer from redundancy of sampling points due to the smoothly decreasing sampling rate. This dynamic resampling strategy ensures a balance between data volume reduction and the retention of critical movement details. The visual comparison highlights the adaptive nature of the dynamic strategy.

B.1.2 Interval Consistent Resampling

Consider different cities may exhibit drastically different sampling intervals due to: Varying data collection protocols (e.g., 1s in City A vs. 5s in City B) and technical limitations or regional preferences in tracking technologies. This heterogeneity poses a serious challenge for developing universal trajectory models, as models trained on data from one region may fail to generalize to regions with different sampling characteristics. Therefore, we performed consistent interval sampling (at random time intervals) on the original dataset to ensure its generalizability across different datasets. Specifically, ICR standardizes the temporal intervals between trajectory points, transforming a trajectory $\boldsymbol{\tau} = \{(x_1, y_1, t_1), (x_2, y_2, t_2), \ldots, (x_n, y_n, t_n)\}$ with irregular time intervals into a trajectory $\boldsymbol{\tau'} = \{(x_1, y_1, \Delta t), (x_2, y_2, \Delta t), \ldots, (x_m, y_m, \Delta t)\}$ with uniform time intervals $\Delta t = t_{i+1} - t_i$, for all $i \in [1, m-1]$.

***Corollary 2**: Temporal Regularity for Cross-Dataset Generalization*

Standpoint: Interval consistent resampling regularizes the temporal dimension of trajectory samples, enhancing the model's ability to generalize across datasets with heterogeneous sampling rates.

Proof: Let $\mathcal{D}_1$ and $\mathcal{D}_2$ be two dataset of region with average sampling intervals $\mu_{\Delta t}^{(1)}$ and $\mu_{\Delta t}^{(2)}$. Assume the temporal pattern recognition task can be formalized as learning a function $f_\theta : \boldsymbol{\tau} \to Y$ where the learned parameters θ should ideally be robust to sampling rate variations. For trajectories with irregular sampling, the model must learn the relationship:

$$y = f_\theta((x_1, y_1, t_1), (x_2, y_2, t_2), \ldots, (x_n, y_n, t_n)). \tag{21}$$

This requires implicitly learning the distribution of time intervals $P(\Delta T)$, which varies across datasets. With ICR, the learning problem becomes:

$$y = f_\theta((x'_1, y'_1, t'_1), (x'_2, y'_2, t'_2), \ldots, (x_m, y_m, t_m)), \quad \text{with } t'_{i+1} - t'_i = \Delta t_{\text{fixed}} \tag{22}$$

where temporal intervals are now consistently fixed, eliminating the need to learn dataset-specific temporal distributions.

Information Entropy Analysis. From the entropy perspective, consider trajectories from different regions r with characteristic sampling intervals Δt^r, where the distribution of intervals can be modeled as:

$$P(\Delta t \mid r) \sim \mathcal{N}(\mu_r, \sigma_r^2), \tag{23}$$

where $\mathcal{N}$ is a dataset distribution with region-specific mean μ_r and variance σ_r^2. The entropy of the joint distribution of regions (or dataset) and sampling intervals is:

$$H(\mathcal{D}, \Delta T) = H(\mathcal{D}) + H(\Delta T \mid \mathcal{D}). \tag{24}$$

This high conditional entropy $H(\Delta T \mid \mathcal{D})$ creates a strong statistical correlation between regions and temporal patterns, forcing region-specific model adaptations. Interval Consistent Resampling transforms the original trajectory $\boldsymbol{\tau}$ into $\boldsymbol{\tau}'$ where $t'_{i+1} - t'_i = \Delta t_{\text{fixed}} \quad \forall i \in [1, m-1]$ This transformation minimizes the conditional entropy:

$$H(\Delta T' \mid \mathcal{D}) \approx 0, \tag{25}$$

which effectively decoupling the temporal sampling pattern from the region. This transformation reduces dataset-specific temporal variability, thereby bringing the conditional distributions of trajectories across datasets closer in distributional space:

$$P(\boldsymbol{\tau}' \mid \mathcal{D}_1) \approx P(\boldsymbol{\tau}' \mid \mathcal{D}_2). \tag{26}$$

The reduction means the model sees more consistent input distributions, thus reducing the domain gap in learning.

For trajectory modeling tasks that focus on spatial patterns rather than absolute temporal dynamics, information loss is minimal when resampling preserves relative temporal order and approximate speed relationships. For a trajectory with velocity profile $v(t) = (p_{i+1} - p_i)/(t_{i+1} - t_i)$, the constraint:

$$\frac{\|p'_{i+1} - p'_i\|}{\|p_{i+1} - p_i\|} \approx \frac{\Delta t_{\text{fixed}}}{\Delta t_i} \tag{27}$$

ensures that relative speed information is preserved even as absolute time intervals are normalized.

B.2 Self-supervised Trajectory Masking

Self-supervised Trajectory Masking (STM) forms a critical component of UniTraj's pre-training strategy, enabling the model to learn robust representations from incomplete trajectory data. While we introduced the concept in the main paper, this appendix provides a more detailed examination of the theoretical foundations, implementation details, and empirical justifications for our masking approach. Our Self-supervised Trajectory Masking framework implements four complementary masking strategies (as illustrated in Figure 7), each designed to simulate different types of real-world data incompleteness and encourage specific learning objectives:

B.2.1 Random Masking

Random Masking applies a uniform probability distribution to select trajectory points for masking, where each point has an equal chance of being masked regardless of its position or significance. Formally, we select a subset of indices:

$$\mathbf{I}_{\text{rand}} \sim \text{Uniform}(\{1, 2, \ldots, n\}). \tag{28}$$

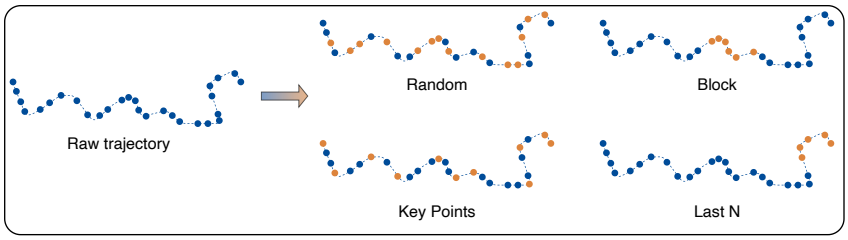


Figure 7: Illustration of the difference masking strategies.

to mask. This strategy forces the model to develop both local and global dependencies, as it must learn to infer missing points from surrounding context without relying on predictable patterns. Random masking is a general masking strategy used to simulate sensor failures or temporary GPS signal loss that often occur in random trajectories.

B.2.2 Block Masking

Block Masking conceals consecutive segments of the trajectory by selecting a starting point k and masking b consecutive points:

$$\mathbf{I}_{\text{block}} = \{k, k+1, \ldots, k+b-1\}, \text{ for some } k. \tag{29}$$

This approach simulates extended sensor failures, tunnels, or urban canyons where trajectory data may be unavailable for continuous periods. The strategy challenges the model to reconstruct substantial missing segments by understanding the broader movement context, encouraging the development of long-range dependencies and trajectory continuity reasoning.

B.2.3 Key Points Masking

Algorithm 1 Ramer–Douglas–Peucker (RDP) Algorithm

RDP($\boldsymbol{\tau}$, s, e, ϵ)
Initialize max distance $d_{\text{max}} \leftarrow 0$
Initialize index $k \leftarrow -1$
for $i = s+1$ to $e-1$ **do**
  Calculate the distance from p_i to $\overline{p_s p_e}$: d_i
  if $d_i > d_{\text{max}}$ **then**
    Update max distance $d_{\text{max}} \leftarrow d_i$
    Update index $k \leftarrow i$
  end if
end for
if $d_{\text{max}} > \epsilon$ **then**
  $\boldsymbol{\tau}_{\text{left}} \leftarrow$ RDP($\boldsymbol{\tau}$, s, k, ϵ)
  $\boldsymbol{\tau}_{\text{right}} \leftarrow$ RDP($\boldsymbol{\tau}$, k, e, ϵ)
  return $\{p_k\} \cup \boldsymbol{\tau}_{\text{left}} \cup \boldsymbol{\tau}_{\text{right}}$
else
  return $\{p_s, p_e\}$
end if

The key points masking adopt the Ramer-Douglas-Peucker (RDP) algorithm [10], which simplifies a trajectory by retaining points that are farthest from the line $\overline{p_1 p_n}$ connecting the first and last points. The indices are determined by

$$\mathbf{I}_{\text{key}} = \{p_k \mid d_{\text{max}}(p_k, \overline{p_1 p_n}) > \epsilon\}, \tag{30}$$

where ϵ is a predefined threshold, and $d_{\text{max}} = \max\{d(p_k, \overline{p_1 p_n}) \mid 2 \le k \le n-1\}$ is the maximum distance measures deviation from this line. As summarized in Algorithm 1, the RDP algorithm iteratively identifies the point p_k that maximizes $d_{\text{max}} = d(p_k, \overline{p_1 p_n})$. If $d_{\text{max}} > \epsilon$, the corresponding point p_k is treated as a key point and included in the mask set **I**key. This process is recursively applied to the trajectory segments $\boldsymbol{\tau}_{\text{left}} = \{p_1, \ldots, p_k\}$ and $\boldsymbol{\tau}_{\text{right}} = \{p_k, \ldots, p_n\}$, isolating critical points

for masking. By focusing on these pivotal points, the model is challenged to reconstruct essential trajectory segments, reinforcing its understanding of key structural patterns within trajectories.

B.2.4 Last N Masking

Last N Masking systematically removes the final N points of each trajectory:

$$\mathbf{I}_{\text{last}} = \{n - N + 1, n - N + 2, \ldots, n\}. \tag{31}$$

This strategy explicitly simulates trajectory prediction scenarios where future positions must be forecasted based on historical observations. By incorporating this masking approach during pre-training, the model develops capabilities directly applicable to trajectory prediction tasks, creating a natural bridge between self-supervised pre-training and downstream forecasting applications.

Corollary 3: Robustness through Comprehensive Masking

Standpoint: Self-supervised Trajectory Masking improves the robustness and generalization ability of the model to incomplete and heterogeneous trajectory data through a comprehensive masking strategy, enabling the model to learn more effective trajectory representations.

Proof: Let the trajectory data space be $\mathcal{D}$, with a true data distribution denoted as $P(\boldsymbol{\tau})$. In real-world applications, due to device limitations, communication failures, and environmental factors, the observed trajectories are often incomplete or irregular, and their distribution is denoted as $P(\tilde{\boldsymbol{\tau}})$. The incompleteness of trajectory data can be formalized as a conditional distribution $P(\tilde{\boldsymbol{\tau}} \mid \boldsymbol{\tau})$, representing the probability of observing an incomplete $\tilde{\boldsymbol{\tau}}$ given a complete trajectory $\boldsymbol{\tau}$.

STM can be formalized as a set of masking functions $\{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_k\}$, each corresponding to a different masking strategy. For a resampled trajectory $\boldsymbol{\tau}' = \{p_1, p_2, \ldots, p_n\}$, the masking function $\mathcal{M}_i$ transforms it as:

$$\tilde{\boldsymbol{\tau}}_i = \mathcal{M}_i(\boldsymbol{\tau}', r_i) = \{p_1, \ldots, [\text{MASK}]_{j \in \mathbf{I}_i}, \ldots, p_n\} \tag{32}$$

where $\mathbf{I}_i \subseteq \{1, 2, \ldots, n\}$ is the index set of masked positions and $r_i = |\mathbf{I}_i|/n$ is the masking ratio.

Information-Theoretic Analysis. From an information-theoretic perspective, STM introduces an artificial information bottleneck that forces the model to learn efficient representations. We define the model objective as minimizing the reconstruction loss:

$$\mathcal{L}(\theta) = \mathbb{E}_{\boldsymbol{\tau} \sim P(\boldsymbol{\tau}), i \sim \mathcal{U}(1,k)} \left[d(f_\theta(\mathcal{M}_i(\boldsymbol{\tau}, r_i)), \boldsymbol{\tau}) \right], \tag{33}$$

where d is a chosen distance metric.

During training, the model needs to learn the joint distribution $P(\boldsymbol{\tau}, \tilde{\boldsymbol{\tau}}_i)$ and estimate the conditional distribution $P(\boldsymbol{\tau} \mid \tilde{\boldsymbol{\tau}}_i)$. By Bayes' theorem:

$$P(\boldsymbol{\tau} \mid \tilde{\boldsymbol{\tau}}_i) = \frac{P(\tilde{\boldsymbol{\tau}}_i \mid \boldsymbol{\tau}) P(\boldsymbol{\tau})}{P(\tilde{\boldsymbol{\tau}}_i)}. \tag{34}$$

By using diverse masking strategies, the model learns to estimate $P(\boldsymbol{\tau} \mid \tilde{\boldsymbol{\tau}}_i)$ across different types of masked trajectories, which is equivalent to learning the true trajectory distribution $P(\boldsymbol{\tau})$ and the various degradation mechanisms $P(\tilde{\boldsymbol{\tau}}_i \mid \boldsymbol{\tau})$.

Optimality Theory of Diversity Complementary Masking Strategies. A key innovation in STM is the use of multiple complementary masking strategies. We define the *coverage region* of the union of masking strategies as:

$$\mathcal{C}(\{\mathcal{M}_1, \ldots, \mathcal{M}_k\}) = \int_{\tilde{\boldsymbol{\tau}} \in \mathcal{D}} \max_{i \in \{1,\ldots,k\}} P_{\mathcal{M}_i}(\tilde{\boldsymbol{\tau}}) \, d\tilde{\boldsymbol{\tau}}, \tag{35}$$

where $P_{\mathcal{M}_i}(\tilde{\boldsymbol{\tau}})$ denotes the distribution of incomplete trajectories generated by masking strategy $\mathcal{M}_i$.

We assert that for a suitable masking ratio and a diverse set of masking strategies $\{\mathcal{M}_1, \ldots, \mathcal{M}_k\}$, the combined coverage region satisfies:

$$\mathcal{C}(\{\mathcal{M}_1, \ldots, \mathcal{M}_k\}) > \max_{i \in \{1,\ldots,k\}} \mathcal{C}(\{\mathcal{M}_i\}). \tag{36}$$

This inequality indicates that the joint use of diverse masking functions provides strictly better coverage over possible incomplete trajectories than any individual strategy.

The advantage of combining multiple masking strategies in STM over using a single masking strategy can also be theoretically justified by comparing the expected reconstruction error. Assume the real-world conditional distribution of incomplete trajectories is $P_{\text{real}}(\tilde{\boldsymbol{\tau}} \mid \boldsymbol{\tau})$. For a single masking strategy $\mathcal{M}_i$, let the generated distribution be $P_{\mathcal{M}_i}(\tilde{\boldsymbol{\tau}} \mid \boldsymbol{\tau})$. Then the expected reconstruction error under this distribution is:

$$\mathbb{E}_{\boldsymbol{\tau}\sim P(\boldsymbol{\tau}),\tilde{\boldsymbol{\tau}}\sim P_{\text{real}}(\tilde{\boldsymbol{\tau}}|\boldsymbol{\tau})}\left[d(f_\theta(\tilde{\boldsymbol{\tau}}),\boldsymbol{\tau})\right] \tag{37}$$

It can be shown that training with a mixture of multiple masking strategies leads to a lower bound on this error compared to using any single strategy. This is because the mixture of diverse masking strategies better approximates the true real-world distribution of incomplete trajectories:

$$KL\left(P_{\text{real}}(\tilde{\boldsymbol{\tau}} \mid \boldsymbol{\tau}) \,\middle\|\, \frac{1}{k}\sum_{i=1}^{k} P_{\mathcal{M}_i}(\tilde{\boldsymbol{\tau}} \mid \boldsymbol{\tau})\right) < \min_{i\in\{1,\ldots,k\}} KL\left(P_{\text{real}}(\tilde{\boldsymbol{\tau}} \mid \boldsymbol{\tau}) \,\|\, P_{\mathcal{M}_i}(\tilde{\boldsymbol{\tau}} \mid \boldsymbol{\tau})\right) \tag{38}$$

Here, $KL(\cdot\|\cdot)$ denotes the Kullback–Leibler divergence.

C Details of UniTraj

In this section, we provide a detailed implementation of UniTraj, including the architecture and parameter settings.

C.1 Overall Architecture

The UniTraj model adopts an encoder-decoder architecture based on transformer blocks, designed to process trajectory data with minimal regional dependency and maximum task adaptability. Figure 8 illustrates the overall framework of UniTraj, which consists of several key components: spatio-temporal tokenization, encoder, decoder, and rotary embedding layers.

Our model takes trajectory points that have already undergone adaptive resampling (ATR) and masking (STM) as described in Appendix B. The input trajectories are represented as sequences of latitude-longitude coordinates and timestamps: $\boldsymbol{\tau} = \{(\text{lng}_i, \text{lat}_i, t_i) | i = 1,\ 2, \ldots, n\}$, where n is the total number of points after resampling. Unlike previous approaches that rely on region-specific features or road network information, UniTraj operates solely on these basic coordinates, enhancing its universal applicability across diverse geographic contexts.

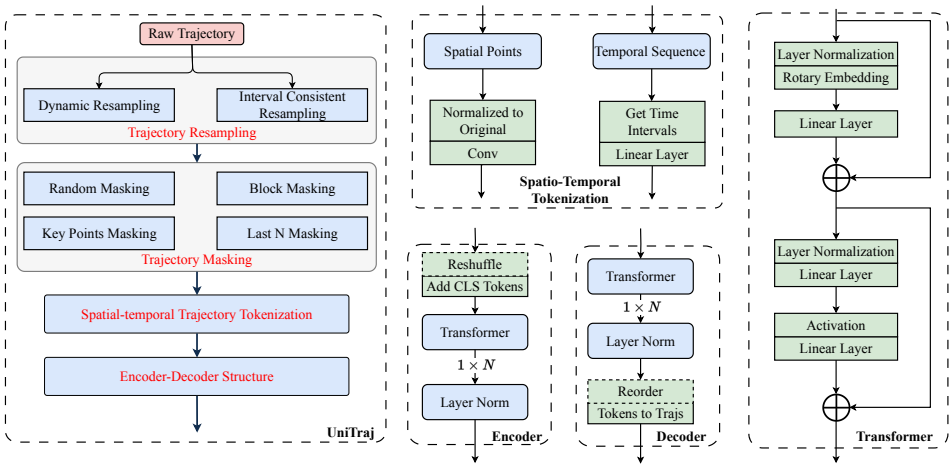


Figure 8: The main architecture and components of UniTraj.

C.2 Input Representation and Embedding

Spatio-Temporal Tokenization. To enhance numerical stability and generalization, all input coordinates are normalized relative to the first point in the trajectory $(x_i, y_i) = (\text{lng}_i - \text{lng}_1, \text{lat}_i - \text{lat}_1)$.

For the spatial component, we project the normalized coordinates into a d-dimensional space using a 1D convolutional neural network, yielding a spatial embedding

$$\boldsymbol{h}_i^s = \text{Conv1D}([x_i, y_i]; \theta_s), \tag{39}$$

where θ_s represents the learnable parameters of the convolutional layer. We use a kernel size of 1 with no stride to capture local spatial dependencies. Similarly, the temporal component, based on the time intervals Δt_i, is embedded into the same d-dimensional space via a linear layer, resulting in a temporal embedding:

$$\boldsymbol{h}_i^t = W_t \cdot \Delta t_i + b_t, \tag{40}$$

where $W_t \in \mathbb{R}^{d\times 1}$ and $b_t \in \mathbb{R}^d$ are learnable parameters. The final embedding for each trajectory point is obtained by element-wise addition of the spatial and temporal components:

$$\boldsymbol{h}_i = \boldsymbol{h}_i^s + \boldsymbol{h}_i^t \tag{41}$$

This dual-tokenization captures both spatial and temporal dynamics, enabling the model to learn relative movement and temporal dependencies effectively.

Rotary Positional Encoding (RoPE). In addition to encoding the spatial and temporal details of each trajectory point, it is essential to capture the relative positional relationships between points. These relationships enable the model to comprehend the movement sequence and the timing between points, both crucial for accurate trajectory modeling. To achieve this, we employ Rotary Position Encoding (RoPE) [32], which maintains the relative positional information between points by rotating the trajectory embedding vectors. Given the combined spatial-temporal embeddings $\boldsymbol{h}_i$ for point i in the trajectory, RoPE applies a rotational transformation:

$$\text{RoPE}(\boldsymbol{h}_i) = \begin{pmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{pmatrix} \begin{pmatrix} \boldsymbol{h}_i^{(1)} \\ \boldsymbol{h}_i^{(2)} \end{pmatrix}, \tag{42}$$

where $\boldsymbol{h}_i^{(1)}$ and $\boldsymbol{h}_i^{(2)}$ are the first and second halves of the embedding $\boldsymbol{h}_i$, and θ_i is a rotation angle that varies proportionally with the position index i. Specifically, θ_i is calculated as $\theta_i = \frac{i}{10000^{2k/d}}$, where k is the index of the embedding dimension, and d is the total dimension of the embedding.

The main advantage of RoPE is its ability to preserve relative positional information through rotational symmetry. This ensures that the relative distance and directional relationships between points are maintained, enabling the model to capture both local patterns (e.g., short-term movements) and global patterns (e.g., long-range directionality) within a trajectory. By encoding these relative positions, RoPE strengthens the model's capacity to understand movement dynamics across varying scales.

C.3 Adaptive Representation Learning

The UniTraj employs an encoder-decoder architecture [13] tailored for trajectory data. The encoder and decoder use Transformer blocks [33] with RoPE-powered self-attention mechanisms to capture dependencies within trajectory embeddings.

Encoder. Given a masked trajectory $\tilde{\boldsymbol{\tau}} = \{p_1, \ldots, [\text{MASK}]_{i\in\mathbf{I}}, \ldots, p_n\}$, we first extract the embedding representations of the unmasked points $\mathbf{H} = \{\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_m\}$ (where $m \leq n$ and $i \notin \mathbf{I}$) through the tokenizer and positional encoding steps. The encoder $\mathbf{E}_\theta$ processes the visible (unmasked) points in a trajectory to generate contextualized representations. It consists of L_e transformer blocks, each incorporating:

1. **Multi-head Self-attention with RoPE:** As described previously, we apply RoPE to the self-attention mechanism:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{Q_{\text{RoPE}} \cdot K_{\text{RoPE}}^T}{\sqrt{d_k}}\right) \cdot V \tag{43}$$

   where Q_{RoPE} and K_{RoPE} are the query and key matrices with RoPE applied.

2. **Feed-forward Network (FFN):** A two-layer FFN with GELU activation:

$$\text{FFN}(x) = W_2 \cdot \text{GELU}(W_1 \cdot x + b_1) + b_2 \tag{44}$$

3. **Layer Normalization and Residual Connections:** Each sub-block is wrapped with layer normalization (Pre-LN) and residual connections:

$$\mathbf{H}' = \text{LayerNorm}(\mathbf{H} + \text{Attention}(\mathbf{H})) \tag{45}$$

$$\mathbf{H}' = \text{LayerNorm}(\mathbf{H}' + \text{FFN}(\mathbf{H}')) \tag{46}$$

The encoder's output is a set of hidden representations $\mathbf{H}^e = \{\boldsymbol{h}_i^e | i = 1, 2, \ldots, m\}$ for the m visible points.

Decoder. The decoder reconstructs the masked points based on the contextualized representations from the encoder. It operates by combining the encoder's embeddings with mask tokens and processing them through L_d transformer layers:

1. **Input Combination:** The decoder input consists of both the encoder outputs for visible points and the mask token embeddings for masked positions:

$$\mathbf{H}_0^d = \text{Reorder}\left(\left\{ \begin{array}{ll} \boldsymbol{h}_i = \mathbf{h}_j^e & \text{if } i = \text{Index}(j),\ i \notin \mathbf{I} \\ \boldsymbol{h}^{\text{mask}} & \text{if } i \in \mathbf{I} \end{array} \right\}\right), \tag{47}$$

   where $\boldsymbol{h}^{\text{mask}}$ represents the mask token embeddings for all masked positions.

2. **Decoder Transformer Blocks:** The combined input is processed through L_d transformer blocks, each with the same structure as the encoder blocks (self-attention with RoPE, FFN, layer normalization, and residual connections). The self-attention mechanism allows information to flow between visible and masked positions:

$$\mathbf{H}_l^d = \text{TransformerBlock}(\mathbf{H}_{l-1}^d) \tag{48}$$

   for $l \in \{1, 2, \ldots, L_d\}$.

3. **Output Projection:** The final layer projects the decoder's representations for the masked positions back to coordinate space:

$$(\hat{x}_j, \hat{y}_j) = W_o \cdot \boldsymbol{h}_{L_d,j}^d + b_o \tag{49}$$

   where j indexes the masked positions, and $W_o \in \mathbb{R}^{2\times d}$ and $b_o \in \mathbb{R}^2$ are learnable parameters. These projected coordinates are then transformed back to the original coordinate system $(\hat{\text{lng}}_j, \hat{\text{lat}}_j) = (\hat{x}_j + \text{lng}_1, \hat{y}_j + \text{lat}_1)$.

UniTraj is trained using a self-supervised learning approach with a reconstruction loss function. For each trajectory, we apply our masking strategies (random, block, key points, or last N), and the model is trained to reconstruct these masked points:

$$\mathcal{L} = \frac{1}{|\mathbf{I}|} \sum_{j \in \mathbf{I}} \|(\hat{x}_j, \hat{y}_j) - (x_j, y_j)\|_2^2, \tag{50}$$

where $\mathbf{I}$ is the set of masked positions, and $\|\cdot\|$ denotes the L2 norm.

C.4 Task-Specific Adaptation

For downstream applications, UniTraj can be used in two primary ways:

1. **Zero-shot Transfer:** The pre-trained model's encoder can be directly applied to extract trajectory representations for various tasks without further training. We use the pre-trained UniTraj as a backbone and attach task-specific Multi-Layer Perceptron (MLP) adapters to the output:

$$\mathbf{H}^{\text{final}} = \text{MLP}(\text{UniTraj}_{\text{encoder}}(\boldsymbol{\tau})) \tag{51}$$

   The MLP adapter typically consists of 2-3 layers with non-linear activations:

$$\mathbf{H}^{\text{ada}} = W_2 \cdot \text{ReLU}(W_1 \cdot \mathbf{H}^e + b_1) + b_2 \tag{52}$$

   where $\mathbf{H}^e$ is the output from the UniTraj encoder.

2. **Fine-tuning:** Update all parameters of the backbone and adapters with specific dataset.

For different downstream tasks, we design specific adapter architectures:

- **For Trajectory Recovery/Prediction:** We can directly use UniTraj's decoder as an adapter without any additional modifications.
- **For Trajectory Classification:** The adapter includes pooling operations followed by fully connected layers to produce class logits.
- **For Trajectory Generation:** The adapter interfaces with generative models by providing conditioned trajectory embeddings.

C.5 Implementation Details

Additionally, we summarize the list of key hyperparameters and implementation-specific settings that may be used in the implementation of UniTraj in Table 5. Specifically, our model contains 8 encoders and 4 decoders, each using 4 heads in the attention layer. The model has approximately 2.38 million parameters, allowing it to balance complexity and computational efficiency. We set the embedding dimension to 128 and employ RoPE to capture spatial and temporal relationships effectively. Our model can handle an arbitrary length of the number of trajectory points and pad it to a length of 200. Naturally, due to the use of rotational positional embedding, our model holds extension capability and supports a maximum length of 512. In addition, when performing the dynamic resampling strategy, we set the minimum number of sampling points to 36 and the maximum to 600, and its minimum sampling rate is 0.35. Finally, we provide the probability of using various masking strategies during training, which can be further adapted to the specific task as we discussed in Section 5.4 and Table 4.

Table 5: General parameters setting for UniTraj.

Parameter	Setting value	Refer range
Encoder Blocks	8	≥ 2
Decoder Blocks	4	≥ 2
Attention Heads	4	≥ 1
Encode Dim	128	$64 \sim 256$
Parameters of Model (Millions)	2.38	–
Mask ratio	0.5	$0.25 \sim 0.75$
Trajectory Length Padding	200	$36 \sim 256$
Maximum Length Padding	512	–
Minimum Trajectory Points	36	–
Maximum Trajectory Points	600	–
Minimum Sampling ratio	0.35	–
Random Masking	0.7	–
Key Points Masking	0.15	–
Block Masking	0.05	–
Last N Masking	0.1	–

D Experiments Details

We use the Adam optimizer and mean square error loss with an initial learning rate of 1×10^{-3} with a learning rate scheduler. The model is trained for 200 epochs with a batch size of 1024, and early stopping is applied based on validation performance. All experiments were conducted using PyTorch, where the foundation model is trained on NVIDIA A100/L40s 40GB GPUs and the baseline experiments are performed on RTX 2080 Ti.

D.1 Datasets

We evaluate the performance of the proposed model using six diverse real-world trajectory datasets. Each dataset represents different data collection scenarios, quality levels, motion patterns, and geographic regions, providing a comprehensive test of the capabilities of UniTraj.

- **WorldTrace**: WorldTrace is our proposed large-scale, globally distributed dataset, which we describe in detail in Section 4.1. We curated a high-quality subset of 1.1 million trajectories from

the original dataset, which have been filtered to remove long stops and loops. Of this subset, 1 million trajectories are designated for model training combined with resampling or masking strategies, with the remaining 100,000 reserved for testing without any operation. To ensure consistency and enable independent **zero-shot evaluations**, the testing dataset is normalized to a sampling interval of 3 seconds per point.

- **Chengdu** [8]: The Chengdu dataset comprises over one million urban mobility trajectories collected from taxis operating in Chengdu, China, reflecting daily commuting and transportation patterns in a densely urbanized area. It features dense, high-frequency (3-second for most trajectories) sampling points that provide detailed insights into active urban environments.
- **Xi'an** [8]: Similar to Chengdu, the Xi'an dataset includes millions of taxi trajectories gathered in Xi'an, China, focusing on movement patterns within another densely populated Chinese city. The data, collected during November 2016, captures the traffic dynamics and urban mobility behaviors specific to this region.
- **GeoLife** [47]: The GeoLife dataset is a widely used trajectory dataset collected over three years by 182 users, primarily in Beijing, China. It is mainly distinguished by a wide variety of travel modes, including walking, cycling and driving. With this data, we can study the trajectory movement patterns and behavioral habits of different travel modes. Besides, this dataset suffers from irregular and often long sampling intervals, which limit its granularity and quality for trajectory analysis.
- **Grab-Posisi** [15]: Sourced from Southeast Asia, this dataset contains 84,000 ride-hailing trajectories, predominantly from the Grab service in cities such as Jakarta and Singapore. The variable sampling intervals across these trajectories provide insights into urban mobility patterns unique to Southeast Asian metropolises.
- **Porto** [28]: The Porto dataset consists of taxi trajectories collected in Porto, Portugal, capturing trips between different areas of the city. Although it provides valuable insight into taxi mobility within the city, the dataset has a relatively low sampling frequency, with long intervals (15 seconds) between data points.

D.2 Tasks Applicability Study Settings

D.2.1 Trajectory Recovery

In this experiment, we randomly mask 50% of trajectory points and test the recovery performance. Specifically, we evaluate UniTraj in both **zero-shot** (trained solely on WorldTrace) and **fine-tuned** settings (trained on WorldTrace and then fine-tuned on each respective dataset), aiming to understand its adaptability with and without task-specific training. Additionally, we compare UniTraj against a diverse range of baselines, including traditional deep learning models (Linear, DHTR [35], Transformer [33], and DeepMove [11]) and pre-trained models (TrajBERT [31] and TrajFM [23]). Performance metrics include Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) with meters, computed based on geographic distance:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i}^{n} |y_i - \hat{y}_i|\,, \tag{53}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i}^{n} (y_i - \hat{y}_i)^2} \tag{54}$$

where y_i and $\hat{y}_i$ are the real and recovered coordinates, respectively.

D.2.2 Trajectory Prediction

In this task, we focus on predicting future trajectories based on historical trajectory points. Following the setup [23] in previous work, we predicted the locations of five future points. The baseline settings and evaluation metrics are consistent with those used for the trajectory recovery task, and experiments were conducted on WorldTrace, Chengdu, and GeoLife datasets.

D.2.3 Trajectory Classification

The Trajectory Classification task is conducted on two datasets, GeoLife and Grab-Posisi. In this task, we will only use the encoder module of the UniTraj as a backbone and then add a classification header.

We compare UniTraj in two settings: without fine-tuning (wo/ft), where only the classifier head is trained, and with fine-tuning (ft), where the entire model is updated. For baselines, we following prior literature [21] use representative classification models including GRU, LSTM, STGN [45], and TrajFormer [21]. Performance is reported by classification accuracy:

$$\text{Acc} = \frac{1}{n}\sum_{i}^{n}\mathbf{I}(y_i, \hat{y}_i), \tag{55}$$

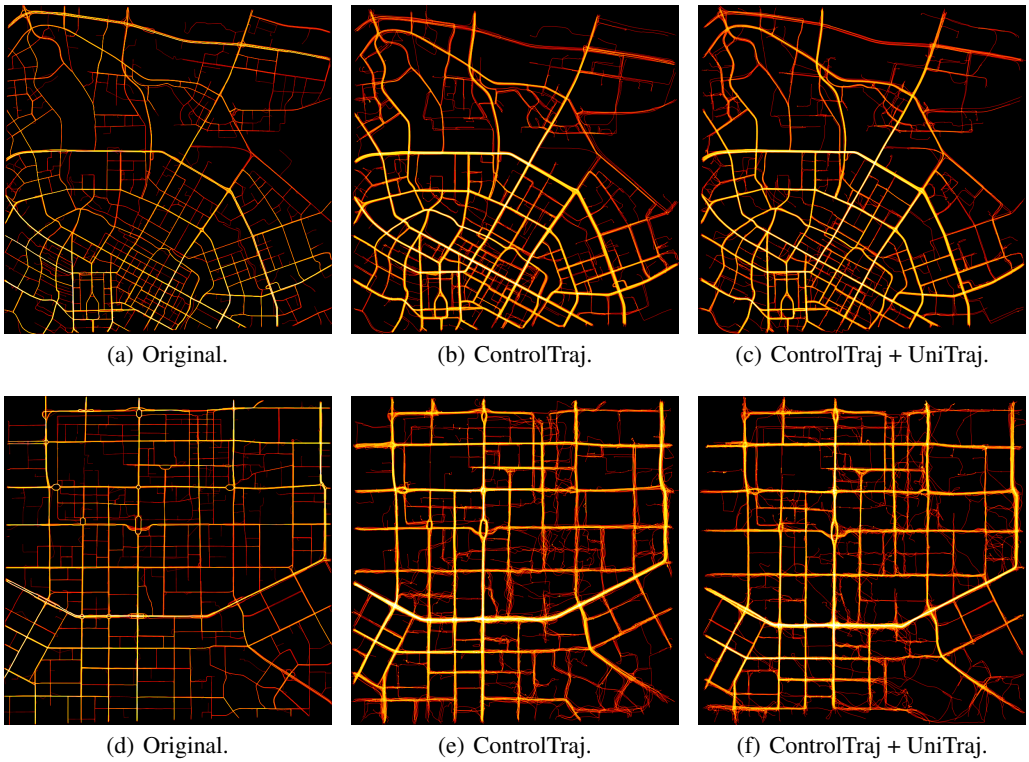
where y_i and $\hat{y}_i$ are the predicted and true labels, respectively, and $\mathbf{I}(\cdot)$ is a indicator function. Following the general settings of previous work, we selected four travel modes from the Geolife dataset, namely walking, bus, bike, and driving. For the Grab-Posisi dataset, there are two travel modes: car and motorcycle.

D.2.4 Trajectory Generation

In this task, we follow the approach in prior work [50], assessing trajectory generation using sequences of road segments that represent trajectories without explicit temporal attributes. Specifically, we use ControlTraj as a downstream task for trajectory generation, where we replace the road segment extraction component (RoadMAE) of the ControlTraj with UniTraj's encoder, testing the effectiveness of the embedded representation. The evaluation includes **density error** metrics [50]:

$$\text{Density Error} = \text{JSD}(G\|O) = \frac{1}{2}\mathbb{D}(\|\frac{(G+O)}{2}) + \frac{1}{2}\mathbb{D}(G\|\frac{(G+O)}{2}), \tag{56}$$

where G is the distribution of the generated trajectories in the city (which divides each city into grids of 16×16 size and calculates the count of trajectory points associated with each grid), and O is the distribution of the original trajectories. $\text{JSD}(\cdot)$ is the Jenson-Shannon divergence for two distributions.



(a) Original. (b) ControlTraj. (c) ControlTraj + UniTraj.

(d) Original. (e) ControlTraj. (f) ControlTraj + UniTraj.

Figure 9: Performance comparison of trajectory generation task with Chengdu dataset (first row), and transfer to Xi'an dataset (second row).

For the this task, UniTraj demonstrates its versatility through integration with existing generative frameworks. By replacing ControlTraj's road segment extraction module with UniTraj, we achieved

a 5.1% reduction in density error (from 0.0039 to 0.0037) when trained and generated on the Chengdu dataset. This improvement, though modest in magnitude, represents a significant advance in trajectory fidelity. More impressively, when transferring the generation capability to Xi'an without retraining—a challenging cross-region scenario—the UniTraj-enhanced generator maintains a density error of 0.0152. In contrast, the baseline ControlTraj experiences a 0.0171 density error when transferred across regions. This cross-region resilience further validates UniTraj's ability to capture universal trajectory patterns that transcend specific geographic contexts. We also show the heatmap visualizations to measure the accuracy and realism of generated trajectories in Figure 9, where brighter regions indicate denser trajectories and darker regions indicate sparser ones. Detailed analysis of the generated trajectories reveals that UniTraj-enhanced generation produces more realistic speed variations, particularly in complex road segments such as intersections, sparse or dense areas. In summary, the above results underscore UniTraj's potential for robust and transferable trajectory generation, proving its effectiveness in both familiar and novel geographic settings.

D.3 Dataset Study Settings

Effect of Dataset Scale and Quality. This task focuses on the impact of dataset size and quality on UniTraj performance. We analyze WorldTrace for the effects of different amounts and qualities of training data. Specifically, we further process the complete WorldTrace dataset by removing cyclic trajectories, removing trajectories with too many stopping points and sparse trajectories. In total, we partitioned a subset of high-quality trajectory data numbering 1 million items, and further partitioned a subset of 10,000, 500,000 trajectory data for UniTraj training.

Effect of Dataset Diversity. The task assessed the impact of using different data coverage (i.e., geographic diversity) on the model. We evaluate the zero-shot performance of UniTraj trained on the WorldTrace and Chengdu datasets, respectively, and tested on multiple real-world trajectory datasets. We chose the Chengdu dataset for comparison because it has very high data quality and has the identical me collection standards as the Xi'an dataset.

D.4 Model Study Settings

For setting the number of encoders decoders for the model, we adopt the following scheme {encoders: 2,4,6,8,12}, {decoders:2,2,4,4,6}, {attention heads:2,2,2,4,8}. We believe that an asymmetric encoder-decoder architecture can significantly reduce the number of parameters while maximizing the performance of the model. And the scaling law between the number of model parameters and the size of the data will be one of the considerations in our future research and model architecture design.

E More Discussion

E.1 Limitation

While UniTraj represents a significant advancement in universal trajectory modeling, several limitations remain that warrant acknowledgment and future investigation. Despite WorldTrace's unprecedented geographic coverage spanning 70 countries, data distribution remains uneven, with certain regions (particularly in Africa and parts of Asia) underrepresented, potentially limiting model performance in these areas. Additionally, our focus on motorized movement may restrict generalization to non-motorized mobility patterns, such as pedestrian trajectories with distinctly different motion properties. The computational resources required for training and deploying UniTraj at scale present practical challenges for resource-constrained environments, necessitating more efficient architectures or distillation approaches. From a technical perspective, UniTraj relies solely on coordinate and temporal information, lacking integration of contextual features like road networks, traffic conditions, and points of interest that could further enhance predictive accuracy. Addressing these limitations represents promising directions for future research, potentially through expanded geographic coverage, multimodal trajectory integration, architecture optimization, context-aware modeling, and continual learning techniques. Nonetheless, we believe that the proposed UniTraj and WorldTrace datasets will contribute to the development of the entire community towards a more generalized, global view of trajectory analysis.

E.2 Broader Impact

This work presents both promising opportunities and notable concerns for society. Positively, this universal trajectory model could popularize mobility intelligence across diverse regions, enabling improved transportation systems in underserved areas without extensive local data collection. The model could drive more efficient urban planning, reduce traffic congestion and emissions, and enhance logistics optimization globally. However, this technology could also enable more pervasive monitoring capabilities, raising surveillance concerns if misused. Additionally, there exists potential for widening technological disparities between resource-rich and resource-constrained organizations. Balancing these implications requires commitment to privacy-preserving techniques and equitable access policies to ensure this technology advances social welfare while minimizing potential harms.

E.3 Ethics Issues

The development and application of large-scale trajectory foundation models raise important ethical considerations, which have been carefully addressed throughout this research. A primary concern is data privacy and the risk of re-identification. To mitigate this, we implemented a robust anonymization protocol: during data collection from OpenStreetMap, we deliberately excluded user identifiers, and the data was subsequently processed through filtering, resampling, and map-matching. This multi-step process significantly abstracts the trajectories from their original form, ensuring the publicly shared dataset is thoroughly anonymized.

Another critical issue is the potential for misuse, particularly for large-scale surveillance. The design of UniTraj inherently addresses this risk at the model level by learning from aggregated, generalized movement patterns rather than individual user paths, making it incapable of reconstructing original trajectories or tracing personal identities. To further prevent the learning of localized patterns, we will not release any model versions fine-tuned on private datasets and will advocate for clear guidelines for responsible use.

Finally, we acknowledge the potential for biases arising from the uneven geographical distribution of training data. To address this, we are employing data augmentation techniques to mitigate the effects of data scarcity in certain regions. We analyze these inherent biases and their potential societal consequences, and we are committed to ongoing efforts to enhance data diversity and model fairness in future work to ensure our research aligns with best practices in responsible AI.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

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

   Answer: [Yes]

   Justification: All claims have been confirmed in the methods and experiments section.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We have discussed this in detail in **Appendix E** of the supplementary materials.

Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: We provide analysis and proof in **Appendix** of the supplementary materials.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We provide code and sample data in the supplementary materials. In addition, we provide detailed parameter settings in the **Appendix C.5**.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

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

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

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

   Answer: [Yes]

   Justification: We provide code and data at https://github.com/Yasoz/UniTraj.

   Guidelines:

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

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: We provide detailed data and hyperparameters settings in the **Appendix C.5**.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [NA]

   Justification: We report results by taking the average of multiple rounds of experiments.

   Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We declare the computing resources used in the experiments in **Appendix D**.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We ensure that no personally identifiable information is included in the data.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We discussed this topic in Discussion at Appendix.

    Guidelines:

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

11. **Safeguards**

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

    Answer: [NA]

    Justification: This paper is not related to this risk.

    Guidelines:

    - The answer NA means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We declare the license of the used data in **Appendix A**.

    Guidelines:

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

13. **New assets**

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

    Answer: [Yes]

    Justification: licensing, acquisition, and processing of the dataset introduced in this paper, and ensure that there are no privacy risks.

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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

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

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [Yes]

    Justification: We use LLM to improve language expression and visualization results.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.