TEMPT: Trajectory Mixing Decomposition for GPS Trajectory Prediction

Anonymous Author
Anonymous Affiliation
anonymous@ijcai.com

Abstract

Trajectory prediction is a critical task with applications spanning intelligent transportation systems, urban planning, and autonomous navigation. Existing approaches often struggle to balance computational efficiency, predictive accuracy, and adaptability to varying trajectory lengths. In this work, we propose TEMPT (TrajEctory Mixing decomPosiTion), a novel framework that pioneers the integration of adaptive patch-based input representation, multi-scale decomposition, and lightweight MLP-based architecture for trajectory prediction. Unlike conventional approaches that rely on rigid architectures or computationally intensive mechanisms like attention, TEMPT introduces a simple-yet-effective dynamic weighted aggregation mechanism to seamlessly process variable-length trajectories while preserving critical spatial-temporal dependencies. By lightweight MLP mixing for trajectory data and a prediction-residual-based loss function, TEMPT effectively captures spatial-temporal dependencies and ensures robust predictions. Extensive experiments on real-world traffic datasets demonstrate that TEMPT not only achieves state-of-the-art accuracy but also significantly reduces computational costs, making it ideal for real-time applications on resource-constrained platforms. This work sets a new benchmark for efficient, adaptive, and scalable trajectory prediction.

1 Introduction

Trajectory prediction is a fundamental task with significant importance in modern intelligent systems [Shi *et al.*, 2023; Gu *et al.*, 2023]. With the proliferation of GPS-enabled devices, vast amounts of trajectory data are being generated daily, presenting both opportunities and challenges for understanding and utilizing this information. Accurate trajectory prediction plays a crucial role in numerous applications, such as navigation systems [Xu *et al.*, 2022a], urban planning [Ma *et al.*, 2019], and traffic management [Li *et al.*, 2021]. For instance, predicting the future movements of vehicles can help optimize traffic flow, reduce congestion, and enhance safety. In addition, trajectory prediction enables personalized location-based services, improving user experiences in areas like ride-hailing and delivery logistics [Shi *et al.*, 2022; Xu *et al.*, 2022b]. Despite its broad applicability, the task remains challenging due to the complexity and variability of human mobility patterns, which are influenced by dynamic factors such as time, weather, and personal preferences.

Recent advancements in GPS trajectory prediction research have explored various methodologies to address the challenges of this task. Early approaches to trajectory prediction heavily relied on statistical methods, such as Hidden Markov Models [Mathew *et al.*, 2012] and Gaussian Processes [Wang *et al.*, 2007]. These models are effective in capturing spatial and temporal correlations in structured data. Nonetheless, their limited scalability and inability to handle complex, nonlinear patterns in real-world trajectories have motivated the search for more advanced solutions. With the rise of deep learning, neural networks have become a dominant approach in trajectory prediction. Recurrent neural networks have been widely used for modeling sequential data [Zhou *et al.*, 2018]. Recently, attention mechanisms and Transformer-based models have further improved the ability to capture long-range dependencies in trajectory data [Liang *et al.*, 2022]. These models excel in scenarios where large-scale data with intricate spatial-temporal patterns is available.

Despite the progress made in GPS trajectory prediction, several open challenges hinder its practical application. **First**, the general trend of current models is being increasingly complex and computationally intensive, making them unsuitable for deployment on edge devices such as low-resource robots or mobile systems. These models often require significant computational resources and power, making real-time prediction infeasible in constrained environments. Achieving a balance between computational efficiency and predictive accuracy is essential for practical applications. **Second**, most existing models rely on processing trajectories of a fixed, predefined length. This rigidity introduces inefficiencies: overly long trajectories may lose critical semantic details during preprocessing (truncation), leading to degraded predictive performance, while shorter trajectories often result in unnecessary computation, reducing model efficiency. Furthermore, this fixed-length assumption fails to accommodate the dynamic and heterogeneous nature of real-world trajectory data, where variability in trajectory length is common. Addressing these challenges requires innovative model architectures that are lightweight, adaptive, and capable of dynamically managing trajectory lengths to maximize both computational efficiency and predictive effectiveness.

To address these challenges, we propose **T**raj**E**ctory **M**ixing decom**P**osi**T**ion (TEMPT), a lightweight and adaptive solution for trajectory prediction. Following the principle of time-series decomposition [Huang *et al.*, 1998; Wen *et al.*, 2019], we intensively utilize the straightforward multi-layer perceptron (MLP) structure for trajectory representation learning and predictive decoding, such that the training and inference processes can be greatly accelerated. Additionally, we propose a simple-yet-highly-effective trainable weighted representation aggregation mechanism to project the variable-length trajectory embeddings into a unified representation space, i.e., the model takes trajectories with variable length and generate high quality predictions. The proposed TEMPT aims to strike a balance between efficiency, scalability, and accuracy, making it suitable for diverse real-world applications, from autonomous systems to smart city infrastructures.

The main contributions of this paper are threefold:

- **Lightweight Model Design**: We present a lightweight GPS trajectory prediction model, TEMPT, that maintains high prediction accuracy while being computationally efficient, enabling real-time application with limited resources.
- **Adaptive Handling of Variable-Length Trajectories**: The proposed TEMPT dynamically processes variable-length input data using a patch-based approach, performing mixing operations within each patch, thereby improving prediction and facilitating an understanding of the contribution of trajectory patches to the final forecast.
- **State-of-the-Art Performance**: The proposed TEMPT demonstrates superior performance compared to both complex state-of-the-art architectures and naïve baseline predictions, validating its effectiveness and practicality.

The remainder of this paper is structured as follows. Section 2 presents a brief review of related work on trajectory prediction and Section 3 introduces the preliminaries. In Section 4, we elaborate on the design and details of the proposed TEMPT. Empirical results and comparisons to state-of-the-art baselines are summarized and discussed in Section 5. Finally, Section 6 concludes this paper.

2 Related Work

In this section, we present a summary on the related work of trajectory prediction. Trajectories, as a special form of time-series, may have their analytical methods inspired from time-series mining approaches. Therefore, we also briefly summarize existing time-series prediction methods. Finally, one of the major techniques adopted by TEMPT is introduced, namely, MLP-Mixer.

2.1 Trajectory prediction

As a type of multivariate time series data, trajectories play a fundamental role in motion-related scenarios. Trajectory prediction is crucial for many applications, particularly in autonomous driving, logistics and delivery optimization, and drone navigation. Trajectory prediction tasks can be categorized into three main types based on the raw data utilized: vision-based, GPS-based, and multi-modal. Vision-based prediction – as explored in works like [Gu *et al.*, 2023; Gu *et al.*, 2021; Liao *et al.*, 2024] – relies on the autonomous vehicle's perception system, where the basic unit is the pixel in an image. This approach mirrors a human driver's perspective, enabling intelligent driving control through visual interactions. GPS-based prediction, as seen in studies like [Liang and Zhao, 2021; Ip *et al.*, 2021; Wang and Feng, 2024], uses GPS sensors to capture the movement of vehicles on roads. Here, the basic unit is a 2D coordinate on a map, offering a straightforward view of driving behavior[1]. Multi-modal prediction combines different data modalities, such as graphs [Bae *et al.*, 2022], language [Bae *et al.*, 2024], and cross-modal models [Choi *et al.*, 2021], to improve prediction accuracy and robustness.

Apart from the types of input data, the prediction models themselves also vary. RNN-based methods are commonly used for sequential data processing and are effective for trajectory prediction [Zhou *et al.*, 2018]. However, they suffer from vanishing gradient issues, making it challenging to capture long-range dependencies, and their sequential nature leads to slower training and inference compared to parallelizable architectures. In recent years, Transformers have been widely adopted for trajectory analysis tasks due to their ability to capture long-range dependencies in data [Zhou *et al.*, 2021]. Despite their advantages, transformers are computationally intensive, requiring significant memory and processing power, which can limit their feasibility for resource-intensive applications [Zeng *et al.*, 2023]. Trajectory prediction, like other time series tasks, benefits from simple yet efficient models, especially in scenarios with limited computational resources or where quick feedback is essential.

2.2 Time-series analysis

Indeed, the past decade witnessed a plethora of efforts on time-series data analysis, and trajectory prediction, as a specialized time-series task, may benefit from these developments. Recurrent Neural Networks (RNN) [Bai *et al.*, 2018], Convolutional Neural Networks (CNN) [Liu *et al.*, 2022; Wu *et al.*, 2023], Transformers [Nie *et al.*, 2023; Zhang and Yan, 2023; Zhou *et al.*, 2022], and their variants [Wang *et al.*, 2023] have been widely explored to model time series data for tasks such as forecasting, classification, and imputation. These methods have contributed significantly to advancing the understanding and application of time-series models in various domains.

However, as model architectures become increasingly complex, efficiency concerns have emerged. Deep learning models often require extensive computational resources [Menghani, 2023], which may not be practical for real-world trajectory prediction tasks when latency and resource constraints are critical. Furthermore, the interpretability of complex models remains an ongoing challenge, limiting their adoption in applications requiring high levels of trans-

[1]We use "trajectory" and "GPS trajectory" interchangably in the sequel without loss of generality.

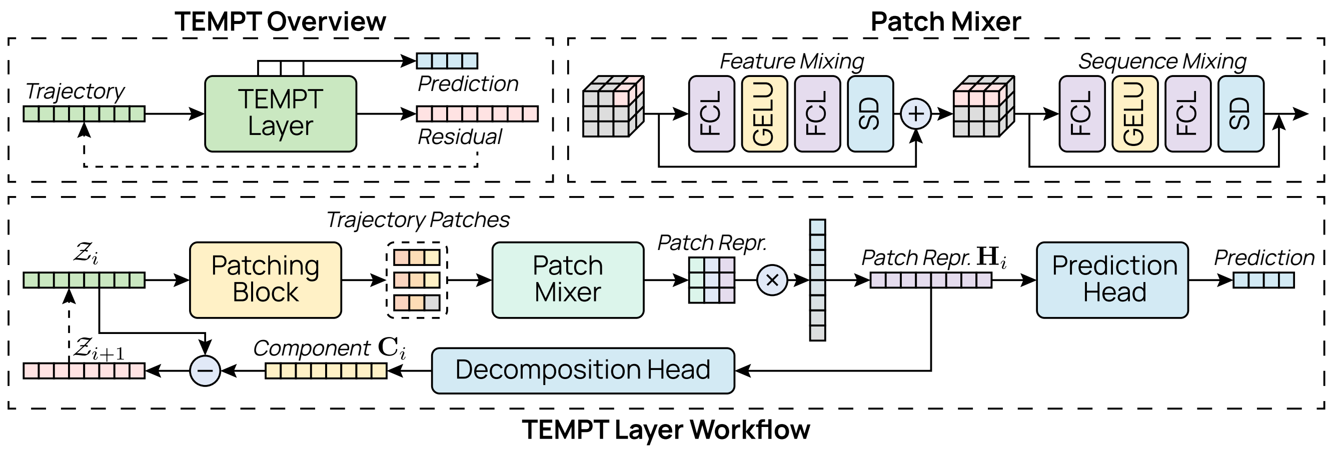

Figure 1: The proposed TEMPT framework.

parency. Interestingly, simpler methods, such as MLP-based approaches [Zeng *et al.*, 2023], have demonstrated that it is possible to achieve effective performance with minimal complexity. These approaches leverage the inherent patterns in time-series data without the overhead of sophisticated architectures. Their success highlights the potential for rethinking model design principles to prioritize simplicity and efficiency without sacrificing accuracy.

2.3 MLP-Mixer

Recent years have seen a resurgence of interest in MLP-based methods (e.g., MLP-Mixer) because of their simplicity and effectiveness [Zhong *et al.*, 2024; Nie *et al.*, 2024]. MLP-based mixers, such as the MLP-Mixer, utilize linear layers to perform mixing operations across specific dimensions of input data, such as spatial or temporal dimensions. These mixers are designed to handle structured data by alternating operations that separately process different dimensions, preserving the inherent relationships within the data. This design has regained attention due to its simplicity, scalability, and ability to deliver competitive performance in tasks like time series analysis and image recognition, without relying on complex mechanisms such as convolutions or attention.

While recent studies have demonstrated that MLP-based architectures can serve as strong baselines for time-series analysis [Zeng *et al.*, 2023; Chen *et al.*, 2023; Zhong *et al.*, 2024], applying them directly to trajectory prediction faces significant challenges due to that trajectory prediction involves unique characteristics, e.g., variable length, that standard MLPs are not designed to address effectively.

3 Preliminaries

In this section, we formally define the concepts related to trajectory prediction and describe the problem in detail.

Definition 1 (GPS Trajectory)**.** *A GPS trajectory is a sequence of consecutively sampled GPS points, denoted as* $\mathcal{X} = \{p_1, \dots, p_M\}$. *Each point* $p_i = [lon_i, lat_i]$, $i \in \{1, \dots, M\}$, *represents the longitude and latitude of a vehicle's location. The trajectory captures the spatial movement of the vehicle as it progresses through a series of geographic positions, providing a fundamental representation of movement in spatial analysis.*

In this paper, we consider trajectories sampled at evenly spaced intervals. For non-evenly sampled trajectories, the timestamps can be incorporated as additional features in the input data to account for temporal irregularities.

Problem 1 (Trajectory Prediction)**.** *Given a sequence of historical GPS points* $\mathcal{X}_{hist} = \{p_1, \dots, p_T\}$, *where* T *is the number of historical points, the goal of trajectory prediction is to forecast the future GPS points* $\mathcal{X}_{future} = \{p_{T+1}, \dots, p_{T+H}\}$. *Here,* H *represents the number of future points to predict. This task involves learning the spatial dependencies and movement patterns present in the historical trajectory data to generate accurate predictions for future positions.*

4 Trajectory Mixing Decomposition

This section presents the technical details of the proposed trajectory prediction framework, referred to as TEMPT, illustrated in Figure 1. TEMPT introduces a simple yet powerful architecture based on the MLP structure. One of TEMPT's key features is its reliance on structured mixing operations to accomplish the trajectory prediction task, effectively eliminating the need for complex temporal techniques (e.g., RNNs, LSTMs, or self-attention) while maintaining lower complexity with respect to sequence length.

4.1 TEMPT Overview

TEMPT consists of a stack of L TEMPT Layers, learning to decompose the input trajectory $\mathcal{X}$ into L decomposed components $\{\mathbf{C}_1, ..., \mathbf{C}_L\}$. Formally, let $\mathcal{Z}_0 = \mathcal{X}$, we have

$$\mathcal{Z}_i = \mathcal{Z}_{i-1} - \sum_{j=1}^{i} \mathbf{C}_j, \quad i = 1, ..., L, \tag{1}$$

such that $\mathcal{Z}_i$ specifies the intermediate residual of the trajectory after the first i components have been decomposed (removed) from the input $\mathcal{X}$. From the data decomposition perspective, the whole process is depicted in Figure 2 and can be viewed as

$$\mathcal{X} = \sum_{i=1}^{L} \mathbf{C}_i + \mathbf{R}, \tag{2}$$

where $\mathbf{R}$ denotes the final residual, or data noise. Indeed, decomposing trajectory data into multiple orthogonal components provides a robust mechanism for extracting multi-scale representations, enabling the model to capture both coarse and fine-grained patterns effectively. This decomposition facilitates learning by reducing the complexity inherent in raw trajectory data, allowing the model to focus on disentangled, meaningful features. This hypothesis is also verified by empirical results in Section 5.

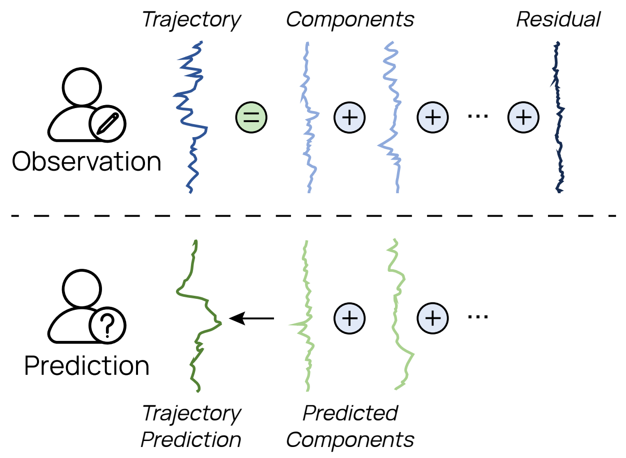

Figure 2: Illustrative example of trajectory prediction with decomposition.

Taking this principle of trajectory decomposition, TEMPT extracts the multi-scale representation $\mathbf{H}_i$ of input trajectory by patching and mixing blocks, and decompose the corresponding principal trajectory components accordingly using a decomposition projection head. With the latent representation, the trajectory prediction task can thereby be accomplished with another prediction projection head:

$$\begin{aligned}\mathbf{H}_i &= \textsc{PatchMix}(\mathcal{Z}_i), \quad i = 1, \dots, L, && (3)\\ \mathbf{C}_{i,t+1:t+H} &= \textsc{PredHead}(\mathbf{H}_i), && (4)\\ \mathbf{C}_i &= \textsc{ReconHead}(\mathbf{H}_i), && (5)\\ \mathcal{Z}_{t+1} &= \mathcal{Z}_t - \mathbf{C}_i. && (6)\end{aligned}$$

In the sequel, we introduce the detailed implementation of TEMPT.

4.2 Trajectory Patching Block

Given input $\mathcal{Z}_i$, each TEMPT Layer first transforms the sequence of length m into trajectory patches with patch size P_i. The patching operation is achieved by segmenting $\mathcal{Z}_i$ along the temporal dimension into non-overlapping patches with stride P_i. The result of this transformation is a high dimensional tensor $\mathbf{P}_i$ of shape $N_i \times P_i \times F$, where $N_i = \lceil M/P_i \rceil$ and $F = 2$ is the feature dimensionality.

4.3 Feature and Sequence Mixing

After the transformation, $\mathbf{P}_i$ is passed to the patch mixer block, a fundamental component of TEMPT designed to process trajectory data by integrating both feature and sequence mixing operations. Either operation is based exclusively on MLPs along the feature and sequence dimension, respectively. The design of mixing operation is depicted in Figure 1, where two fully connected layers are stacked with a GELU non-linear activation in between, a final Stochastic Depth to deactivate selected layers during training for robustness and acceleration, and a residual link governing the identity mapping of the operation. The final output after both mixing is denoted by $\mathbf{M}_i$

While these two mixing operations share the same architecture, they are designed to capture the latent representation from different perspectives. In particular, the first feature mixing operates within each time step of trajectory patches to capture feature-wise dependencies. This step is crucial for extracting localized spatial features and ensuring that the positioning information are effectively modeled. Further, the second sequence mixing operation focuses on intra-sequence (trajectory patch) interactions by capturing temporal dependencies across the sequence. This step ensures that the model considers the sequential nature of the data, which is essential for trajectory prediction tasks.

After the mixing operations, TEMPT is capable of effectively extracting both localized and global features, ensuring robust performance on diverse trajectory datasets. As we are adopting the straightforward MLP structure, the first challenge in Section 1, i.e., computationally intensive prediction, can be resolved. Nonetheless, the second one, i.e., variable trajectory length, remains open. Inspecting into the tensor shape $N_i \times P_i \times F$ after patching, one may figure that N_i is indeed variable subject to the length of input trajectories. A third MLP mixing operation, as how previous MLP-Mixers did, obviously cannot address this problem. Therefore, TEMPT adopts a simple-yet-highly-effective trainable weighted aggregation mechanism to fuse the output of mixing operations as follows:

$$\mathbf{H}_i = \mathbf{W}_{i,1:N_i}\mathbf{M}_i, \tag{7}$$

where $\mathbf{W}_i$ is a learnable weighting vector regulating the influence of each trajectory patch on the representation. The length of this vector is set to a larger number than $\max N_i$. This design, though seems rough at first sight, indeed accords with its rationale. With a trained TEMPT, values in $\mathbf{W}_i$ is a straightforward indication of the importance of trajectory patches for subsequent prediction. During back-propagation, $\mathbf{W}_i$ values correspond to the near past are better trained than distant ones, which also reflects their intuitive importance.

4.4 Prediction Projection Heads

With the extracted latent representation of input trajectories, the prediction task is accomplished by a decoder-like prediction projection head. Following the lightweight implementation principle, TEMPT reuses the mixing structure introduced in Section 4.3 and outputs the prediction locations in one go. Specifically, the aforementioned weighting aggregation generates one trajectory representation in each TEMPT Layer. The representation is subsequently fed into a standalone mixing operation, whose output dimension is $H \times F$. This output $\mathbf{C}_{i,t+1:t+H}$ is considered as the prediction of the corresponding decomposed $\mathbf{C}_i$ c.f. (2). The final prediction is the naïve aggregation $\sum_i \mathbf{C}_{i,t+1:t+H}$.

One may note that in TEMPT, the prediction is conducted in batch instead of the more common auto-regressive manner. The design is primarily for reducing the computational footprint for long-horizon forecast scenarios. Indeed, TEMPT can be naturally adapt to auto-regressive prediction by taking the output values as the next-step input.

4.5 Decomposition and Training

Intuitively, TEMPT can be trained by comparing the prediction and ground truth values in a self-supervised manner. However, we figure that incorporating an additional residual loss [Zhong *et al.*, 2024] alongside the prediction loss is instrumental in ensuring the preservation and utilization of the principal components of the trajectory data. This residual loss serves as a regularization and guides the model to focus on the most informative and meaningful aspects of the data, reinforcing the learning of robust representations. This dual-objective training framework encourages a deeper alignment between the learned representations and the underlying dynamics of the trajectory data, making the predictive process more effective and reliable.

To achieve this objective, TEMPT employs another mixing operation after the representation learning with its output designed to decompose the remaining principal component of the trajectory, i.e., $\mathbf{C}_i$. By (6), TEMPT removes the extracted principal component from the input trajectory data, and recursively calculates the final residual $\mathbf{R} = \{r_{ij}\}$ from any input trajectory $\mathcal{X}$. According to [Zhong *et al.*, 2024], training against the residual loss aims at developing a zero-mean, non-autocorrelated noise signal from the raw data. Correspondingly, the loss is defined as

$$\mathcal{L}_{\text{res}} = \frac{\sum r_{ij}}{MF} + \frac{\sum(\text{ReLU}(|a_{ij} - \alpha/\sqrt{M}|))^2}{(M-1)F}, \qquad (8)$$

$$a_{ij} = \frac{\sum_{k=j+1}^{L}(r_{ik} - \bar{r}_i)(r_{i,k-j} - \bar{r}_i)}{\sum_{k=1}^{L}(r_{ik} - \bar{r}_i)^2}, \qquad (9)$$

where (9) defines the autocorrelation of $\mathbf{R}$ and α is the maximum tolerance of autocorrelation coefficient. The first term in (8) minimizes the mean of $\mathbf{R}$, while the second minimizes the autocorrelation.

The training pipeline of TEMPT involves feeding input sequences into the model, where each TEMPT Layers generates both predictive outputs and decomposed trajectory components. The former is aggregated and tested against the ground truth future trajectory points, while the latter is reduced from the original input and compute the residual. The training objective is formulated as

$$\mathcal{L} = \mathcal{L}_{\text{pred}} + \lambda\mathcal{L}_{\text{res}}, \qquad (10)$$

where $\mathcal{L}_{\text{pred}}$ is the L2 prediction loss, $\mathcal{L}_{\text{res}}$ is the residual loss, and λ is a loss balancing parameter.

5 Experiment

In this section, we present a series of comprehensive experiments on two real-world datasets to show the efficacy of TEMPT on trajectory prediction. We first introduce the experimental configurations, including the datasets, implementation details of TEMPT, and state-of-the-art base- line methods. Subsequently, we conduct an extensive comparative study to assess TEMPT's predictive performance. Further, we analyze the impact of key components on TEMPT's performance. Finally, we study the structural design of TEMPT.

5.1 Experiment Settings

To evaluate the effectiveness of TEMPT, we conducted experiments on two real-world taxi trajectory datasets collected from Chengdu and Xi'an[2]. The sampling rate for these datasets are 3 seconds. Based on the trajectory data, we formalize three trajectory prediction scenarios: *Short* for predicting the trajectory in the next 15, 30, and 60 seconds ($\{5, 10, 20\}$-step prediction) with past 30 observations; *Long* for predicting the trajectory in the next 30, 60, 120, and 600 seconds ($\{10, 20, 40, 100\}$-step prediction) with past 400 observations. A third *All* scenario does not limit the number of past observations, i.e., variable-length input trajectory, and predicts for 15, 30, 60, 120 seconds into the future. For all scenarios, we train TEMPT with $L = 4$. Patch sizes P_i are $\{30, 10, 5, 1\}$ for short-term prediction and otherwise $\{40, 20, 10, 1\}$.

We compare TEMPT with the following baselines:

- NexuSQN (2024) [Nie *et al.*, 2024]: An MLP-based spatial and temporal mixer for general traffic forecasting.
- MSD-Mixer (2024) [Zhong *et al.*, 2024]: A trajectory prediction model that integrates multi-scale decomposition and mixing mechanisms.
- MLPST (2023) [Zhang *et al.*, 2023]: A lightweight MLP-based spatiotemporal model for trajectory prediction.
- PatchTST (2023) [Nie *et al.*, 2023]: A transformer-based model for long-sequence time-series forecasting.
- DLinear (2023) [Zeng *et al.*, 2023]: A linear model optimized for efficiency in time-series forecasting.
- Naïve: A baseline that assumes no change from the last observed value.

All models are implemented and evaluated in a Python environment equipped with PyTorch on an NVIDIA 2080 Ti GPU. The primary evaluation metrics are Root Mean Squared Error (RMSE) between prediction results and real trajectories, together with Mean Absolute Error (MAE) measuring the geographical distance error between prediction and ground truth trajectory.

5.2 Overall Performance

Table 1 presents the overall performance of TEMPT and the baseline models across short-term and long-term prediction tasks. For short-term predictions, TEMPT consistently achieves the lowest MSE and MAE values across all horizons, indicating its superior accuracy in capturing fine-grained temporal dependencies. On average, TEMPT reduced the MAE by a consistent remarkable 25.14% over any second-best performing baselines. For instance, in the Chengdu dataset, TEMPT attains an MAE of 2.34 for 15-second prediction, significantly outperforming the second-best model, MSD-Mixer, by a 59.72% error reduction. Similarly, in the Xi'an dataset, TEMPT demonstrates outstanding performance with an MAE of 8.33 for 30-second prediction, a 43.0% error reduction over the second-best MLPST. The results suggest that TEMPT's multi-scale patching and weighted aggregation mechanisms effectively capture the complex temporal dependencies in the trajectory data, leading to superior prediction accuracy.

[2]https://outreach.didichuxing.com/

Table 1: Trajectory prediction results. The best results are in **bold** and the second bests are underlined.

Models			TEMPT (Ours)		NexuSQN (2024)		MSD-Mixer (2024)		MLPST (2023)		PatchTST (2023)		DLinear (2023)		Naïve	
			RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
Chengdu	Short	15s	**2.67**	**2.34**	11.58	9.77	6.68	5.81	12.74	10.08	15.46	11.46	23.86	16.02	70.69	47.26
		30s	**5.80**	**5.26**	21.92	16.39	8.86	7.93	17.31	14.17	14.45	12.11	26.86	19.65	217.49	145.91
		60s	**15.74**	**12.84**	39.34	30.17	19.31	15.60	37.40	29.13	31.63	25.42	43.19	32.83	330.06	248.91
	Long	30s	**14.85**	**13.37**	24.13	18.36	20.24	17.05	23.39	18.28	22.09	18.16	16.78	15.10	64.64	44.93
		60s	**31.62**	**27.59**	50.88	36.92	45.53	35.60	41.68	33.30	63.64	45.81	68.77	48.27	136.61	93.57
		120s	**68.09**	**54.39**	96.82	76.46	123.09	89.09	79.61	63.07	110.48	84.41	134.59	95.30	288.22	193.59
		300s	**255.87**	**198.45**	333.03	239.80	521.56	364.31	287.59	210.81	441.57	314.63	566.50	384.20	2046.59	1370.74
	All	15s	**2.73**	**2.40**	11.68	9.93	6.73	6.07	12.99	10.25	16.06	12.15	24.79	17.02	68.74	45.11
		30s	**5.89**	**5.34**	22.24	16.68	9.03	8.16	18.04	14.95	15.36	13.20	27.59	20.43	226.30	158.12
		60s	**15.88**	**13.06**	40.87	32.01	19.93	16.00	38.26	30.03	32.59	26.77	44.40	33.68	345.57	255.34
		120s	**65.40**	**50.77**	93.35	74.12	115.83	83.49	73.96	60.21	105.20	82.29	130.91	92.62	283.49	189.96
Xi'an	Short	15s	**3.13**	**2.53**	8.00	6.01	7.82	5.88	4.31	3.39	7.15	5.61	8.80	6.59	47.64	33.28
		30s	**9.60**	**8.33**	24.14	17.25	28.52	19.35	17.70	14.62	20.65	16.80	35.31	23.87	103.41	69.53
		60s	**24.48**	**21.67**	33.54	27.31	43.18	33.18	49.89	38.08	62.14	44.26	71.75	50.61	204.49	142.20
	Long	30s	**18.30**	**12.55**	33.57	22.43	19.27	14.20	21.69	15.89	24.43	18.46	26.87	19.07	106.55	62.04
		60s	**25.28**	**20.05**	41.47	31.97	34.81	26.67	36.21	29.70	35.94	28.73	36.43	29.89	187.57	115.29
		120s	**62.19**	**50.44**	101.87	68.01	81.83	63.82	72.52	57.40	78.98	62.33	91.16	65.46	327.66	210.07
		300s	**225.09**	**193.30**	367.49	280.26	350.29	278.60	415.77	295.41	292.76	250.35	400.15	290.72	698.94	482.44
	All	15s	**3.15**	**2.60**	8.43	6.35	8.11	6.34	4.60	3.62	7.47	5.92	9.36	6.94	45.43	30.81
		30s	**9.68**	**8.65**	24.38	17.89	28.80	20.14	18.11	15.04	21.19	17.72	36.60	24.77	105.22	71.80
		60s	**24.56**	**21.79**	33.83	27.51	43.54	33.49	50.31	38.99	63.31	45.09	72.87	51.17	210.61	147.72
		120s	**62.54**	**50.71**	102.00	68.87	82.25	64.29	73.36	58.31	79.70	63.43	92.37	66.18	306.83	196.98
Avg. Error ↓ (%)			-	-	49.10	32.21	40.04	23.45	40.00	23.49	47.93	33.74	57.37	42.06	87.02	80.58

In long-term predictions, TEMPT also excels, particularly for prediction horizons up to 120 seconds. For instance, in the Chengdu dataset, TEMPT achieves an MAE of 54.39 for 2-minute prediction, while the next-best model, MLPST, records an MAE of 33.30. In the Xi'an dataset, TEMPT achieves an MAE of 50.44 for the same horizon, outperforming PatchTST and MLPST. However, for extremely long horizons (6 minutes), while TEMPT still performs competitively, the gap with transformer-based baselines narrows slightly, suggesting potential room for optimization in capturing ultra-long trajectory dependencies.

The superiority of TEMPT is further validated by the *All* scenario, where it consistently outperforms all baselines across all prediction horizons with larger improvements over the previous fix-length scenarios. The results demonstrate the effectiveness of the proposed weighted aggregation mechanism on learning and condensing trajectory representations. TEMPT's robustness and generalization capabilities are also thereby validated, showcasing its ability to handle variable-length input trajectories and predict accurately into the future.

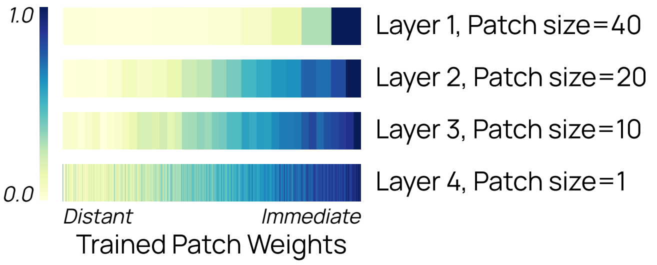

Figure 3: Trained patch weight values in weighted aggregation.

To further demonstrate the lightweight nature of TEMPT, we compare its training and prediction time with the baselines. The average training time for one epoch and average prediction time for one trajectory is summarized in Table 2, showcasing TEMPT's significantly lower computation footprint compared to all baselines, approximately 1.5 times faster than the performing baselines MSD-Mixer and MLPST. The efficiency advantage is attributed to the model's MLP-based compact architecture design and the use of adaptive patching and weighted aggregation mechanisms, which enable effective information aggregation and processing across multiple scales. Figure 3 illustrates the learnt values of the weighted aggregation mechanism in TEMPT. The result accords with the intuition that more immediate patches are assigned higher weights, i.e., more important, while patches further in the past are given lower weights. The visualization demonstrates the model's ability to adaptively assign importance to different trajectory patches based on their relevance to the prediction task.

5.3 Ablation Study

While the MLP-based architecture of TEMPT contribute to the training and inference efficiency, we figure that the effectiveness of TEMPT are rooted in the proposed adaptive patching and weighted aggregation. To validate the effect of the proposed modules, we implement the following variants of TEMPT:

- TEMPT-U: TEMPT without multi-scale patching. We use the same patch size for all layers and the patch size is set to 10, i.e., 30 seconds.
- TEMPT-S: TEMPT with static pooling. We replace learnable weighted pooling with static pooling which treats all patches with the same importance. The weights are frozen

Table 2: Time complexity study results. Training time is for one epoch, prediction time is for one trajectory.

Models		TEMPT		NexuSQN		MSD-Mixer		MLPST		PatchTST		DLinear	
		Train	Pred.	Train	Pred.	Train	Pred.	Train	Pred.	Train	Pred.	Train	Pred.
Pred. Len.	15s	**171s**	**0.06s**	273s	0.10s	250s	0.09s	261s	0.09s	425s	0.68s	378s	0.26s
	30s	**177s**	**0.08s**	281s	0.11s	257s	0.11s	273s	0.15s	451s	0.80s	380s	0.26s
	60s	**180s**	**0.09s**	286s	0.13s	261s	0.12s	279s	0.16s	465s	0.93s	384s	0.30s
	120s	**190s**	**0.11s**	289s	0.17s	265s	0.18s	289s	0.20s	490s	1.03s	397s	0.34s
	300s	**205s**	**0.18s**	311s	0.28s	288s	0.27s	304s	0.31s	510s	1.34s	423s	0.49s
Speed $\uparrow$		-	-	1.5×	1.6×	1.4×	1.5×	1.5×	1.7×	2.5×	9.6×	2.1×	3.3×

Table 3: Ablation study results on Xi'an long-term 30-second prediction.

Model	TEMPT	TEMPT-U	TEMPT-S	TEMPT-I
RMSE	15.46	18.95	18.51	16.57
MAE	12.55	13.82	13.60	12.59

during training so that the pooling process is static.

- TEMPT-I: inverted TEMPT. We arrange the layers with their patch sizes in ascending order to discover the inverted scale of patching.

We adopt Xi'an long-term 30-second prediction to validate the efficacy of the proposed modules. The results in Table 3 show that varying patching sizes play a critical role in capturing temporal dependencies, as evidenced by the performance drop in TEMPT-U. We hypothesize that the multi-scale patching mechanism enables TEMPT to capture both short-term and long-term dependencies effectively, essential in trajectory prediction task. Further, a non-negligible performance drop is observed in TEMPT-S, indicating the importance of the learnable weighted pooling mechanism in capturing the relevance of different trajectory patches. This observation is consistent with the intuition that different trajectory patches contribute differently to the prediction task, and the model should adaptively assign importance to each patch based on its relevance. What surprised us is that the inverted TEMPT, TEMPT-I, achieves competitive performance compared to the original TEMPT. This result suggests that the model is capable of learning the importance of different patches regardless of their order, indicating the robustness and flexibility of the proposed weighted aggregation mechanism.

5.4 TEMPT Structure Analysis

TEMPT incorporates L TEMPT Layers for multi-level trajectory decomposition, where each layer employs a different patch size P_i. To validate their combined impact on TEMPT's performance, we define several variants with varying numbers of layers and patch sizes as follows:

- Original TEMPT: four TEMPT Layers and $P_i = \{40, 20, 10, 1\}$.
- T-3A: three TEMPT Layers and $P_i = \{20, 10, 1\}$.
- T-3B: three TEMPT Layers and $P_i = \{100, 40, 20\}$.
- T-4A: four TEMPT Layers and $P_i = \{100, 40, 20, 10\}$.
- T-4B: four TEMPT Layers and $P_i = \{200, 100, 400, 20\}$.
- T-5A: five TEMPT Layers and $P_i = \{100, 40, 20, 10, 1\}$.

Similar to the previous ablation study, we also adopt Xi'an long-term 30-second prediction for comparison. The simulation results are summarized in Table 4.

Table 4: Hyper-parameter study results on Xi'an long-term 30-second prediction.

Model	TEMPT	T-3A	T-3B	T-4A	T-4B	T-5A
RMSE	15.46	18.43	17.61	16.38	16.74	15.84
MAE	12.55	13.78	13.41	12.73	12.96	12.59

From the table, we observe that the original TEMPT configuration achieves the best performance, indicating that the four-layer configuration with patch sizes of 40, 20, 10, and 1 strikes the best balance between accuracy and computational efficiency. This result aligns with the architecture design discussed in Section 4, where the multi-scale patching mechanism enables TEMPT to capture both short-term and long-term dependencies effectively. Further, the results also indicate that while achieves slightly worse performance, the performance gap of TEMPT variants are relatively small, suggesting that TEMPT is robust to variations in the number of layers and patch sizes. This robustness is crucial for practical applications where the model needs to adapt to different scenarios and data characteristics.

6 Conclusion

In this paper, we introduced TEMPT (Trajectory Mixing Decomposition), a novel framework designed for GPS trajectory prediction with a focus on computational efficiency, flexibility, and predictive accuracy. TEMPT leverages a lightweight MLP-based architecture that integrates adaptive patch-based input representation, multi-scale mixing mechanisms, and a robust decomposition strategy. TEMPT enables effective handling of trajectories of varying lengths while maintaining high prediction performance across both short-term and long-term forecasting tasks using only fractions of computation over state of the arts.

Extensive experiments on real-world traffic datasets, including Chengdu and Xi'an, demonstrate that TEMPT consistently outperforms state-of-the-art models in terms of both accuracy and efficiency. The model's low footprint nature with its remarkable prediction performance is particularly suitable for real-time applications in resource-constrained environments. Ablation studies further validate the effectiveness of key architectural components, namely, adaptive patching and weighted pooling. Future work will focus on extending TEMPT to ultra-long-term predictions and exploring its adaptability to other spatiotemporal prediction domains.

References

[Bae *et al.*, 2022] Inhwan Bae, Jin-Hwi Park, and Hae-Gon Jeon. Learning pedestrian group representations for multi-modal trajectory prediction. In *European Conference on Computer Vision*, pages 270–289. Springer, 2022.

[Bae *et al.*, 2024] Inhwan Bae, Junoh Lee, and Hae-Gon Jeon. Can language beat numerical regression? language-based multimodal trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 753–766, 2024.

[Bai *et al.*, 2018] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling, 2018.

[Chen *et al.*, 2023] Si-An Chen, Chun-Liang Li, Nate Yoder, Sercan O Arik, and Tomas Pfister. TSMixer: An all-MLP architecture for time series forecasting. *arXiv preprint arXiv:2303.06053*, 2023.

[Choi *et al.*, 2021] Chiho Choi, Joon Hee Choi, Jiachen Li, and Srikanth Malla. Shared cross-modal trajectory prediction for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 244–253, 2021.

[Gu *et al.*, 2021] Junru Gu, Chen Sun, and Hang Zhao. DenseTNT: End-to-end trajectory prediction from dense goal sets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15303–15312, 2021.

[Gu *et al.*, 2023] Junru Gu, Chenxu Hu, Tianyuan Zhang, Xuanyao Chen, Yilun Wang, Yue Wang, and Hang Zhao. ViP3D: End-to-end visual trajectory prediction via 3d agent queries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5496–5506, 2023.

[Huang *et al.*, 1998] Norden E. Huang, Zheng Shen, Steven R. Long, Manli C. Wu, Hsing H. Shih, Quanan Zheng, Nai-Chyuan Yen, Chi Chao Tung, and Henry H. Liu. The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 454(1971):903–995, 1998.

[Ip *et al.*, 2021] André Ip, Luis Irio, and Rodolfo Oliveira. Vehicle trajectory prediction based on LSTM recurrent neural networks. In *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*, pages 1–5. IEEE, 2021.

[Li *et al.*, 2021] Mingqian Li, Panrong Tong, Mo Li, Zhongming Jin, Jianqiang Huang, and Xian-Sheng Hua. Traffic flow prediction with vehicle trajectories. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 294–302, 2021.

[Liang and Zhao, 2021] Yuebing Liang and Zhan Zhao. NetTraj: A network-based vehicle trajectory prediction model with directional representation and spatiotemporal attention mechanisms. *IEEE Transactions on Intelligent Transportation Systems*, 23(9):14470–14481, 2021.

[Liang *et al.*, 2022] Yuxuan Liang, Kun Ouyang, Yiwei Wang, Xu Liu, Hongyang Chen, Junbo Zhang, Yu Zheng, and Roger Zimmermann. TrajFormer: Efficient trajectory classification with transformers. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 1229–1237, 2022.

[Liao *et al.*, 2024] Haicheng Liao, Shangqian Liu, Yongkang Li, Zhenning Li, Chengyue Wang, Yunjian Li, Shengbo Eben Li, and Chengzhong Xu. Human observation-inspired trajectory prediction for autonomous driving in mixed-autonomy traffic environments. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 14212–14219. IEEE, 2024.

[Liu *et al.*, 2022] Minhao Liu, Ailing Zeng, Muxi Chen, Zhijian Xu, Qiuxia Lai, Lingna Ma, and Qiang Xu. SCINet: Time series modeling and forecasting with sample convolution and interaction. *Advances in Neural Information Processing Systems*, 35:5816–5828, 2022.

[Ma *et al.*, 2019] Yuexin Ma, Xinge Zhu, Sibo Zhang, Ruigang Yang, Wenping Wang, and Dinesh Manocha. TrafficPredict: Trajectory prediction for heterogeneous traffic-agents. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 6120–6127, 2019.

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

[Menghani, 2023] Gaurav Menghani. Efficient deep learning: A survey on making deep learning models smaller, faster, and better. *ACM Comput. Surv.*, 55(12), March 2023.

[Nie *et al.*, 2023] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations*, 2023.

[Nie *et al.*, 2024] Tong Nie, Guoyang Qin, Lijun Sun, Wei Ma, Yu Mei, and Jian Sun. Contextualizing MLP-Mixers spatiotemporally for urban traffic data forecast at scale. *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[Shi *et al.*, 2022] Liushuai Shi, Le Wang, Chengjiang Long, Sanping Zhou, Fang Zheng, Nanning Zheng, and Gang Hua. Social interpretable tree for pedestrian trajectory prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2235–2243, 2022.

[Shi *et al.*, 2023] Liushuai Shi, Le Wang, Sanping Zhou, and Gang Hua. Trajectory unified Transformer for pedestrian trajectory prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9675–9684, October 2023.

[Wang and Feng, 2024] Yilin Wang and Yiheng Feng. IDM-Follower: A model-informed deep learning method for car-following trajectory prediction. *IEEE Transactions on Intelligent Vehicles*, 2024.

[Wang *et al.*, 2007] Jack M Wang, David J Fleet, and Aaron Hertzmann. Gaussian process dynamical models for human motion. *IEEE transactions on pattern analysis and machine intelligence*, 30(2):283–298, 2007.

[Wang *et al.*, 2023] Huiqiang Wang, Jian Peng, Feihu Huang, Jince Wang, Junhui Chen, and Yifei Xiao. MICN: Multi-scale local and global context modeling for long-term series forecasting, 2023.

[Wen *et al.*, 2019] Qingsong Wen, Jingkun Gao, Xiaomin Song, Liang Sun, Huan Xu, and Shenghuo Zhu. RobustSTL: A robust seasonal-trend decomposition algorithm for long time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5409–5416, July 2019.

[Wu *et al.*, 2023] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. TimesNet: Temporal 2D-variation modeling for general time series analysis, 2023.

[Xu *et al.*, 2022a] Pei Xu, Jean-Bernard Hayet, and Ioannis Karamouzas. SocialVAE: Human trajectory prediction using timewise latents. In *European Conference on Computer Vision*, pages 511–528. Springer, 2022.

[Xu *et al.*, 2022b] Yuan Xu, Jiajie Xu, Jing Zhao, Kai Zheng, An Liu, Lei Zhao, and Xiaofang Zhou. MetaPTP: An adaptive meta-optimized model for personalized spatial trajectory prediction. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2151–2159, 2022.

[Zeng *et al.*, 2023] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are Transformers Effective for Time Series Forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 11121–11128, 2023.

[Zhang and Yan, 2023] Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The eleventh international conference on learning representations*, 2023.

[Zhang *et al.*, 2023] Zijian Zhang, Ze Huang, Zhiwei Hu, Xiangyu Zhao, Wanyu Wang, Zitao Liu, Junbo Zhang, S Joe Qin, and Hongwei Zhao. MLPST: MLP is all you need for spatio-temporal prediction. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 3381–3390, 2023.

[Zhong *et al.*, 2024] Shuhan Zhong, Sizhe Song, Weipeng Zhuo, Guanyao Li, Yang Liu, and S-H Gary Chan. A multi-scale decomposition MLP-Mixer for time series analysis. *Proceedings of the VLDB Endowment*, 17(7):1723–1736, 2024.

[Zhou *et al.*, 2018] Fan Zhou, Qiang Gao, Goce Trajcevski, Kunpeng Zhang, Ting Zhong, and Fengli Zhang. Trajectory-user linking via variational autoencoder. In *IJCAI*, pages 3212–3218, 2018.

[Zhou *et al.*, 2021] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 11106–11115, 2021.

[Zhou *et al.*, 2022] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 27268–27286. PMLR, 17–23 Jul 2022.