Continuous-Time Transformer Based Channel Prediction with Non-Uniform Pilot Pattern

Yiliang Sang, Ke Ma, Lebin Yao, Pengyu Wang,
Zhaocheng Wang, *Fellow, IEEE*, Zhu Han, *Fellow, IEEE*, and Sheng Chen, *Life Fellow, IEEE*

***Abstract*—Deep learning based channel prediction has garnered significant attention to mitigate channel aging in high-mobility multiple-input multiple-output (MIMO) systems. However, existing channel prediction methods extract the temporal correlations from the channel sequences estimated at uniform pilots, which require dense pilot configuration to mitigate Doppler aliasing in high-mobility scenarios and incur substantial estimation overhead. To tackle this problem, we propose a channel prediction method based on continuous-time transformer with the non-uniform pilot pattern, thereby enabling accurate prediction across arbitrary time scales with only a small number of pilots. Specifically, we first design the non-uniform pilot pattern based on Chebyshev polynomial roots and then prove its optimality under Doppler-dominated channel variations with relatively stable user velocity, wherein a subset of pilots are densely configured to provide a finer resolution of Doppler phase estimation. To adapt to the non-uniform pattern, a continuous-time transformer is further proposed, which integrates the superior feature extraction capability of transformer with the continuous-time modeling strength of neural ordinary differential equation (ODE) for flexibly processing the estimated channel sequences with non-uniform time scales. More concretely, the attention mechanism is extended to the continuous-time domain by incorporating neural ODE, while a high-frequency temporal encoding is designed to fit rapidly time-varying channels. Besides, an element-wise prediction mechanism is proposed to efficiently capture temporal correlations and prevent overfitting. Simulation results demonstrate that our proposed method can realize accurate continuous-time channel prediction in high-mobility scenarios, and significantly outperforms existing channel prediction methods.**

***Index Terms*—Multiple-input multiple-output, channel prediction, high-mobility scenario, Doppler aliasing, deep learning.**

I. Introduction

Multiple-input multiple-output (MIMO) is one of the fundamental technologies in the 5G, 6G and beyond wireless communication systems [1]. To fully leverage the potential benefits of MIMO for downlink transmission, wireless channels need to be accurately estimated at the base station (BS). However, as the number of user equipments (UEs) and antennas grows exponentially [2], [3], channel estimation can lead to substantial pilot overhead. Besides, to mitigate channel aging in mobile scenarios [4], channel estimation is required to be performed periodically, which further exacerbates the pilot overhead issue. To solve the above issues, channel prediction has been widely adopted to predict the future channel sequence by exploiting the temporal correlations between the historical and future channels [5].

The conventional channel prediction methods mainly include the autoregressive (AR) model [6], [7], the sum-of-sinusoids model [8], and the linear extrapolation model [9]. Specifically, the study [6] designed a vector Kalman filter with AR parameter estimation, while the work [7] improved the performance of AR model by utilizing the channel sparsity in the angle-delay domain. The work [8] proposed a sum-of-sinusoids model with estimation of signal parameters via rotational invariance technique (ESPRIT). A Prony based linear extrapolation predictor was derived in [9]. Nevertheless, these conventional methods usually rely on strict assumptions, such as the number of paths, which may not match the complex multi-path channels in practical scenarios and could cause severe performance degradation.

To solve this problem, deep learning based channel prediction has been extensively studied [10]–[17], benefiting from the capability of deep learning to extract the sophisticated intricate correlations within channel sequences in a data-driven manner. Specifically, the classic recurrent neural network (RNN) was utilized to capture the temporal correlations of channel sequences in [10], while the study [11] proposed RNN variants, long short-term memory (LSTM) and gated recurrent unit (GRU), to achieve more accurate prediction. However, these RNN-like neural networks predict the future channel sequence in a sequential manner, which may struggle to accurately capture long-term dependencies [18] and suffer from error accumulation issues [19]. In contrast, owing to the capability to effectively extract long-term dependencies via attention mechanisms and avoid error accumulation through parallel prediction, transformer has been extensively applied in channel prediction [12]–[14], channel estimation [20]–[23], and predictive beamforming [24]–[27]. For instance, the work [12] proposed a generative transformer with parallel prediction framework, while a hybrid prediction model that integrates transformer with GRU was employed in [13]. The work [14] designed a linear based lightweight transformer with encoder-only architecture. The prediction performance of different neural networks was compared in [15]. Besides, the position information of UEs was utilized in [16], and a physics-inspired model named C-GRBFnet was proposed to achieve efficient channel representation and prediction. The work [17] derived

This work was supported in part by the National Natural Science Foundation of China under Grant 62471275, in part by NSF ECCS-2302469, in part by Japan Science and Technology Agency (JST) Adopting Sustainable Partnerships for Innovative Research Ecosystem (ASPIRE) JPMJAP2326. *(Corresponding author: Zhaocheng Wang.)*

Y. Sang, K. Ma, L. Yao, P. Wang, and Z. Wang are with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, China (e-mails: sangyl23@mails.tsinghua.edu.cn, make15@tsinghua.org.cn, yaolb24@mails.tsinghua.edu.cn, wangpengyu@mail.tsinghua.edu.cn, zcwang@tsinghua.edu.cn).

Z. Han is with the Department of Electrical and Computer Engineering at the University of Houston, Houston, TX 77004 USA (e-mail: hanzhu22@gmail.com).

S. Chen is with the School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K. (e-mail: sqc@ecs.soton.ac.uk).

TABLE I
COMPARISON BETWEEN PROPOSED AND EXISTING DEEP LEARNING BASED CHANNEL PREDICTION METHODS.

Methods	Attention mechanism	Continuous-time modeling	Pilot pattern	Prediction mechanism
RNN-like [10], [11]	×	×	Uniform	Vector-wise
Transformer [12]–[14]	√	×	Uniform	Vector-wise
Neural ODE [28]–[31]	×	√	Uniform	Vector-wise
Proposed	√	√	Non-uniform	Element-wise

a novel channel deduction framework, where the information of historical channel sequences is fused with the coarse estimation of current channels to enhance prediction robustness. Nonetheless, since the above methods do not model the specific time interval, they are restricted to predict the future channel sequence with uniform time scales. In other words, the periods of the historical and predicted channel sequences need to be strictly equal. Therefore, to mitigate severe channel aging in high-mobility scenarios, channel estimation needs to be frequently executed to obtain the future channel sequence with a short prediction period, which inevitably imposes enormous pilot overhead.

In order to achieve finer time-scale prediction without additional channel estimation, continuous-time channel prediction has gained widespread attention in recent years [28]–[32], which reconstructs the underlying continuous-time channels from the channel sequence estimated at uniform pilots. As a widely adopted method for continuous-time modeling, neural ordinary differential equation (ODE) fits channel dynamics by neural networks and applies numerical integration to implement arbitrary time-scale prediction. The work [28] proposed to use RNN to capture the temporal correlations from the historical channel sequence, whose output is utilized by neural ODE for continuous-time channel prediction. The integrations of neural ODE with LSTM and GRU were respectively designed in [29] and [30]. The physics-inspired neural ODE was proposed in [31] to effectively learn channel dynamics. A very recent work [32] proposed ODE-Former to support continuous-time prediction from channel sequences sampled at arbitrary time slots. This work integrates the transformer-like structure with neural ODE to enhance the learning capability, and utilizes the integration length instead of the attention mechanism to directly determine temporal relationships.

However, these existing neural ODE based methods face three problems, which may significantly degrade the continuous-time prediction performance. Firstly, the most crucial issue is Doppler aliasing. The temporal channel variations primarily originate from the rotation of Doppler phase, which is proportional to the UE velocity [12], [33]. As the UE velocity increases, the Doppler phase may rotate beyond one full cycle (2π) between adjacent channel estimations. In this case, Doppler aliasing occurs according to the Nyquist criterion, which implies that the accurate estimation of Doppler phase for reconstructing continuous-time channels becomes infeasible. To avoid Doppler aliasing, existing neural ODE based methods are required to densely insert a large number of uniform pilots, resulting in substantial estimation overhead. Secondly, existing neural ODE based methods generally rely on RNN-like neural networks to extract the temporal correlations within channel sequences, which makes them difficult to precisely capture long-term dependencies and leads to error accumulation issues. Thirdly, existing neural ODE based methods transform the channel vector into the angle domain, and subsequently predict all of its elements at once by simultaneously exploiting the spatial and temporal correlations of channels, called vector-wise prediction mechanism. This mechanism imposes a shared temporal correlation component (e.g., attention weights of transformer) across different elements within the angle-domain channel vector. However, considering that different elements generally correspond to distinct propagation paths and exhibit diverse Doppler shifts [7], the temporal dynamics of the elements within one angle-domain channel vector are usually inconsistent. Therefore, enforcing a shared temporal correlation component across different elements in the vector-wise prediction mechanism could impair the prediction performance. Besides, since an element can contain redundant information for other elements, the vector-wise prediction mechanism may inadvertently hamper the precise extraction of temporal correlations for each element and increase the risk of overfitting.

To address the aforementioned problems, we propose the element-wise continuous-time transformer with the non-uniform pilot pattern for channel prediction in this paper. Specifically, a non-uniform pilot pattern according to Chebyshev polynomial roots is proposed, where a subset of pilots are configured with high density. In this way, compared to the uniform pattern, the proposed non-uniform pattern can achieve a significantly finer estimation resolution of Doppler phase under the same pilot overhead. Furthermore, under the conditions of Doppler-dominated channel variations with relatively stable UE velocity, we prove that the proposed non-uniform pilot pattern optimally facilitates the precise reconstruction of continuous-time channels in terms of the maximum norm. Besides, we design a continuous-time transformer tailored for accurate arbitrary time-scale prediction, harnessing the powerful feature extraction capability of transformer in conjunction with the continuous-time modeling strength of neural ODE. Different from ODE-Former [32] that directly determines temporal relationships by the integration length, the proposed continuous-time transformer explicitly extends the attention mechanism to the continuous-time domain via neural ODE, thereby realizing adaptive extraction of continuous-time correlations. More concretely, the queries, keys, and values in the attention mechanism are generalized to their underlying continuous-time trajectories via neural ODE and closed-form interpolation functions. Then, the correlations between these trajectories are solved by numerical integration, thereby realizing the continuous-time attention. Meanwhile, we design

a high-frequency temporal encoding to replace the standard positional encoding in transformer, which explicitly embeds the time information into the network input and enables the effective fitting of rapidly time-varying channels in high-mobility scenarios. In addition, inspired by the well-known channel-independent framework in the area of time series modeling [19], [34], we propose to predict each element of the channel vector at once instead of the whole channel vector, while the prediction of different elements shares the same network parameters. This prediction mechanism is termed as element-wise prediction, which can generate element-specific attention weights to effectively capture temporal correlations and mitigate overfitting. Simulation results demonstrate that our proposed method can realize accurate continuous-time channel prediction in high-mobility scenarios and remarkably surpasses its conventional counterparts. Our code has been made publicly available in [35] to facilitate reproduction. The main contributions of this paper can be summarized as follows.

- We theoretically analyze that continuous-time channel prediction faces the issue of Doppler aliasing in high-mobility scenarios according to the Nyquist criterion. To solve this issue, a non-uniform pilot pattern based on Chebyshev polynomial roots is proposed, which can provide a significantly finer estimation resolution of Doppler phase compared to the uniform pilot pattern. Moreover, we prove that the proposed non-uniform pilot pattern is optimal for the reconstruction of continuous-time channels in the sense of maximum norm.
- We propose the continuous-time transformer, wherein the attention mechanism is integrated with neural ODE to accurately extract the correlations across arbitrary time scales, and a high-frequency temporal encoding is designed to effectively track the fast time-varying channels. On this basis, we further propose an element-wise prediction mechanism, which enables the continuous-time transformer to pay attention to efficiently extracting the temporal correlations of each element and reduce the risk of overfitting.

Table I briefly compares our proposed method with existing deep learning based channel prediction methods. In contrast to these existing methods, our method utilizes the non-uniform pilot pattern to effectively address Doppler aliasing, integrates transformer with neural ODE for accurate arbitrary time-scale prediction, and adopts the element-wise prediction mechanism to efficiently capture the temporal correlations.

This paper is organized as follows. Section II presents the system model. In Section III, the issue of Doppler aliasing for continuous-time channel prediction is analyzed and the proposed non-uniform pilot pattern is elaborated. The proposed continuous-time transformer and element-wise prediction mechanism are presented in Section IV. Section V provides the simulation results, and finally the conclusions are summarized in Section VI.

Notations: $\mathbb{C}^{m\times n}$ and $\mathbb{R}^{m\times n}$ represent the $m\times n$ complex and real spaces, respectively. $C^N\big([a,\ b]\big)$ denotes the space of N-times continuously differentiable real-valued functions defined on the closed interval $[a,\ b]$. $\mathrm{j}=\sqrt{-1}$ represents the imaginary unit. Vectors and matrices are separately denoted by boldface lower-case and capital letters. $[\cdot]_i$ denotes the i-th element of a vector, while $[\cdot]_{i,j}$ is the (i,j)-th element of a matrix. $(\cdot)^{\mathrm{T}}$ denotes the transpose. $\|\cdot\|_2$ represents the 2-norm of a vector. $|\cdot|$ denotes the modulus operator, while $\lceil\cdot\rceil$ represents the ceiling function. $\|f\|_\infty=\max_{x\in[a,b]}|f(x)|$ denotes the uniform norm of function f. $a\propto b$ means that a is proportional to b. $\mathrm{mod}(x,2\pi)$ represents x modulo 2π. $\mathbb{N}_+$ is the set of positive natural numbers, and $\mathcal{T}$ denotes the set of time slots.

II. System Model

We consider a BS serving one single-antenna UE in this paper for simplicity, whereas our proposed channel prediction method can be directly extended to the scenario of multiple-antenna UEs. We further assume that the BS is equipped with dual-polarized antennas in the uniform planar array (UPA) with the numbers of antennas in the horizontal and vertical directions being M_{h} and M_{v}, respectively. Consequently, the total antenna number at the BS is given by $M=2M_{\mathrm{h}}M_{\mathrm{v}}$.

A. Channel Model

The 3D time-varying multi-path channel model [36] is considered, which has been extensively adopted in existing channel prediction works [37], [38]. Specifically, let us denote the channel vector at time slot t as $\boldsymbol{h}(t)\in\mathbb{C}^{M\times 1}$. Then its m-th element $[\boldsymbol{h}(t)]_m$ can be expressed as

$$[\boldsymbol{h}(t)]_m=\sum_{l_{\mathrm{p}}=1}^{L_{\mathrm{p}}}\alpha_{l_{\mathrm{p}}}e^{\mathrm{j}2\pi\frac{(\boldsymbol{r}_{\mathrm{tx},l_{\mathrm{p}}})^{\mathrm{T}}\boldsymbol{d}_{\mathrm{tx},m}}{\lambda}}e^{\mathrm{j}2\pi f_{\mathrm{D},l_{\mathrm{p}}}t}e^{-\mathrm{j}2\pi f_{\mathrm{c}}\tau_{l_{\mathrm{p}}}},\qquad(1)$$

where L_{p} denotes the number of paths, $\alpha_{l_{\mathrm{p}}}$, $f_{\mathrm{D},l_{\mathrm{p}}}$ and $\tau_{l_{\mathrm{p}}}$ are the corresponding complex gain, Doppler shift and delay of the l_{p}-th path, respectively, λ is the wavelength and f_{c} denotes the carrier frequency, while $\boldsymbol{d}_{\mathrm{tx},m}$ represents the location vector of the m-th BS antenna, and $\boldsymbol{r}_{\mathrm{tx},l_{\mathrm{p}}}$ denotes the spherical unit vector at azimuth departure angle $\phi_{\mathrm{tx},l_{\mathrm{p}}}$ and elevation departure angle $\theta_{\mathrm{tx},l_{\mathrm{p}}}$, which can be written as

$$\boldsymbol{r}_{\mathrm{tx},l_{\mathrm{p}}}=\big[\sin\theta_{\mathrm{tx},l_{\mathrm{p}}}\cos\phi_{\mathrm{tx},l_{\mathrm{p}}},\sin\theta_{\mathrm{tx},l_{\mathrm{p}}}\sin\phi_{\mathrm{tx},l_{\mathrm{p}}},\cos\theta_{\mathrm{tx},l_{\mathrm{p}}}\big]^{\mathrm{T}}.\quad(2)$$

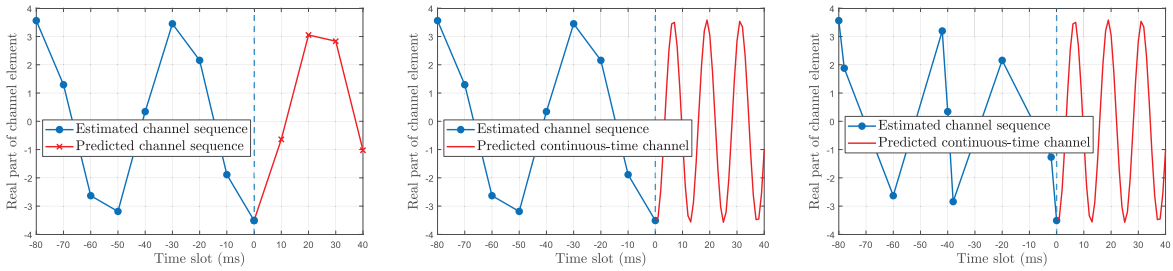
In (1), the temporal channel variations stem from the rotation of Doppler phase $2\pi f_{\mathrm{D},l_{\mathrm{p}}}t$, and the Doppler shift $f_{\mathrm{D},l_{\mathrm{p}}}$ can be further written as [7], [12], [33]

$$f_{\mathrm{D},l_{\mathrm{p}}}=\frac{f_{\mathrm{c}}(\boldsymbol{r}_{\mathrm{rx},l_{\mathrm{p}}})^{\mathrm{T}}\boldsymbol{v}}{c},\qquad(3)$$

where c is the light speed, $\boldsymbol{r}_{\mathrm{rx},l_{\mathrm{p}}}$ is the spherical unit vector at azimuth arrival angle $\phi_{\mathrm{rx},l_{\mathrm{p}}}$ and elevation arrival angle $\theta_{\mathrm{rx},l_{\mathrm{p}}}$, which can be expressed by substituting $\big\{\phi_{\mathrm{rx},l_{\mathrm{p}}},\theta_{\mathrm{rx},l_{\mathrm{p}}}\big\}$ for $\big\{\phi_{\mathrm{tx},l_{\mathrm{p}}},\theta_{\mathrm{tx},l_{\mathrm{p}}}\big\}$ in (2), and $\boldsymbol{v}$ is the UE speed vector at azimuth angle ϕ_{v} and elevation angle θ_{v}, which is given by

$$\boldsymbol{v}=v\big[\sin\theta_{\mathrm{v}}\cos\phi_{\mathrm{v}},\sin\theta_{\mathrm{v}}\sin\phi_{\mathrm{v}},\cos\theta_{\mathrm{v}}\big]^{\mathrm{T}}.\qquad(4)$$

In (4), v is the UE velocity. According to (3) and (4), as the UE velocity v increases, the Doppler shift $f_{\mathrm{D},l_{\mathrm{p}}}$ becomes larger, thereby leading to more drastic temporal channel variations.



(a) (b) (c)

Fig. 1. Different channel prediction frameworks: (a) uniform time-scale prediction with uniform pilot pattern; (b) continuous-time prediction with uniform pilot pattern; (c) continuous-time prediction with non-uniform pilot pattern. Current time slot is defined as 0 ms, estimation time slots of uniform and non-uniform pilot patterns are set to $\{-80\,\text{ms}, -70\,\text{ms}, -60\,\text{ms}, -50\,\text{ms}, -40\,\text{ms}, \cdots, 0\,\text{ms}\}$ and $\{-80\,\text{ms}, -78\,\text{ms}, -60\,\text{ms}, -42\,\text{ms}, -40\,\text{ms}, \cdots, 0\,\text{ms}\}$, respectively.

B. Problem Formulation

Let us define the estimated channel vector at time slot t as $\widetilde{\boldsymbol{h}}(t) \in \mathbb{C}^{M\times 1}$. To mitigate channel aging in mobile scenarios, channel prediction is essential to obtain the future channel sequence by utilizing the historical estimated channel sequence. Most existing channel prediction works [10]–[17] focus on the uniform time-scale prediction $g_{\text{uni}}(\cdot)$ as presented in Fig. 1 (a). By defining the current time slot as 0, the uniform time-scale prediction $g_{\text{uni}}(\cdot)$ can be formulated as

$$\left\{\widehat{\boldsymbol{h}}(pT_{\text{e}})\right\}_{p=1}^{P} = g_{\text{uni}}\left(\left\{\widetilde{\boldsymbol{h}}\big((j-J)T_{\text{e}}\big)\right\}_{j=1}^{J}\right), \tag{5}$$

wherein T_{e} represents the period of channel estimation, J and P are the lengths of estimated and predicted channel sequences, respectively, while $\widehat{\boldsymbol{h}}(t) \in \mathbb{C}^{M\times 1}$ denotes the predicted channel vector at time slot t. Since the specific time interval is not modeled in the uniform time-scale prediction, only the channel sequence with period T_{e} can be accurately predicted. However, in high-mobility scenarios, channel aging becomes severe. To tackle this problem, the estimation period T_{e} needs to be shortened in the uniform time-scale prediction, which can cause enormous pilot overhead.

To handle this issue, recent works [28]–[31] investigated continuous-time channel prediction, which reconstructs the underlying continuous-time channels from the historical channel sequence, thereby realizing arbitrary time-scale prediction. Specifically, considering the estimation time slots $\{t_j\}_{j=1}^{J}$ with $t_j \le 0$ and prediction time slots $\{t_p\}_{p=1}^{P}$ with $t_p > 0$, the continuous-time prediction $g_{\text{con}}(\cdot)$ can be formulated as

$$\left\{\widehat{\boldsymbol{h}}(t_p)\right\}_{p=1}^{P} = g_{\text{con}}\left(\left\{\widetilde{\boldsymbol{h}}(t_j), t_j\right\}_{j=1}^{J}, \{t_p\}_{p=1}^{P}\right), \tag{6}$$

where the time information is embedded into the network input for reconstructing continuous-time channels. For these existing continuous-time prediction works, the uniform pilot pattern is assumed in the estimated channel sequence, as shown in Fig. 1 (b), i.e., $t_j = (j-J)T_{\text{e}}$ for $j \in \{1, 2, ..., J\}$. However, continuous-time channel prediction in high-mobility scenarios may require to insert numerous uniform pilots with high density, with the reasons detailed in the next section.

III. Non-Uniform Pilot Pattern for Continuous-Time Channel Prediction

In this section, we first introduce the challenge of Doppler aliasing faced by continuous-time channel prediction in high-mobility scenarios. Next, we propose a non-uniform pilot pattern as shown in Fig. 1 (c), which offers a finer estimation resolution of Doppler phase to address the problem of Doppler aliasing. Then in the sense of maximum norm, we prove the optimality of our proposed non-uniform pilot pattern for the reconstruction of continuous-time channels. To improve the readability, the definitions of notations used in Section III are listed in Table II.

TABLE II
Definitions of Notations.

Notations	Definitions
$J/\widetilde{J}$	Estimation length of channel sequence before/after inserting additional pilots
T_{e}	Estimation period before inserting additional pilots
$T_{\text{e,min}}$	Minimum estimation period
$T_{\text{che,min}}/T_{\text{uni,min}}$	Minimum estimation period of non-uniform/uniform pilot patterns
N	Number of inserted additional pilots
$N_{\min}$	Minimum number of inserted additional pilots
$t_{\text{che}}^{(n)}/t_{\text{che},j}^{(n)}$	Time slot of n-th inserted pilot in time intervals $[-1,\ 1]$ and $[(j-J)T_{\text{e}}, (j+1-J)T_{\text{e}}]$
$t_{\text{che},j}$	Time slot of j-th pilot after inserting non-uniform pilot pattern

A. Doppler Aliasing in High-Mobility Scenarios

Reconstructing the underlying continuous-time channels from the estimated channel sequence is key to achieving accurate prediction across arbitrary time scales. However, this reconstruction imposes constraints on the minimum period of channel estimation $T_{\text{e,min}}$ according to the Nyquist criterion, which is elaborated in the following theorem.

Theorem 1. *To avoid Doppler aliasing, the minimum period of channel estimation $T_{\text{e,min}}$ needs to satisfy*

$$T_{\text{e,min}} < \frac{c}{f_{\text{c}} v}. \tag{7}$$

Proof: According to (1) and (3), the time-varying part of channels is an analytic signal $e^{\text{j}2\pi \frac{f_{\text{c}}(\boldsymbol{r}_{\text{rx},l_{\text{p}}})^{\text{T}}\boldsymbol{v}}{c}t}$, and its maximum rotation of Doppler phase is $2\pi\frac{f_{\text{c}}v}{c}t$ when the spherical unit vector, $\boldsymbol{r}_{\text{rx},l_{\text{p}}}$, and the UE speed vector in (4), $\boldsymbol{v}$, are aligned. Based on the Nyquist criterion for analytic signals [39], the minimum period of channel estimation $T_{\text{e,min}}$ needs to satisfy

$$2\pi\frac{f_{\text{c}}v}{c}T_{\text{e,min}} < 2\pi, \tag{8}$$

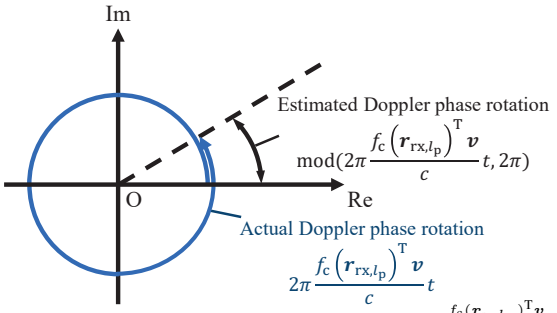


Fig. 2. Illustration of time-varying part of channels $e^{\mathrm{j}2\pi\frac{f_c(\boldsymbol{r}_{\mathrm{rx},l_p})^{\mathrm{T}}\boldsymbol{v}}{c}t}$ when Doppler phase rotates beyond one full cycle (2π).

which is equivalent to (7). ■

Remark 1: Fig. 2 illustrates the time-varying part of channels $e^{\mathrm{j}2\pi\frac{f_c(\boldsymbol{r}_{\mathrm{rx},l_p})^{\mathrm{T}}\boldsymbol{v}}{c}t}$ to better understand Doppler aliasing. Due to the periodicity, only the rotation of Doppler phase in one full cycle $\mathrm{mod}\left(2\pi\frac{f_c(\boldsymbol{r}_{\mathrm{rx},l_p})^{\mathrm{T}}\boldsymbol{v}}{c}t, 2\pi\right)$ can be estimated. When $T_{\mathrm{e,min}} \geq \frac{c}{f_c v}$, the maximum rotation of Doppler phase satisfies $2\pi\frac{f_c v}{c}T_{\mathrm{e,min}} \geq 2\pi$. In this case, the actual rotation of Doppler phase $2\pi\frac{f_c(\boldsymbol{r}_{\mathrm{rx},l_p})^{\mathrm{T}}\boldsymbol{v}}{c}t$ may not be recovered from the estimated rotation of Doppler phase $\mathrm{mod}\left(2\pi\frac{f_c(\boldsymbol{r}_{\mathrm{rx},l_p})^{\mathrm{T}}\boldsymbol{v}}{c}t, 2\pi\right)$.

Remark 2: Doppler aliasing is a commonly encountered issue in practical scenarios. For instance, considering the widely used carrier frequency $f_c = 3.5\,\mathrm{GHz}$ and the UE velocity $v = 60\,\mathrm{km/h}$, $T_{\mathrm{e,min}}$ needs to satisfy $T_{\mathrm{e,min}} < 5.14\,\mathrm{ms}$ to avoid Doppler aliasing. However, according to the 3rd generation partnership project (3GPP) specifications [40], the maximum period of channel estimation can be set to 160 ms, and thus Doppler aliasing may occur in this scenario.

B. Chebyshev Polynomial Roots Based Non-Uniform Pilots

To address Doppler aliasing in high-mobility scenarios, existing uniform pilot based continuous-time prediction methods need to shorten the estimation period to satisfy $T_{\mathrm{e,min}} \leq \frac{c}{f_c v}$ based on Theorem 1, which incurs considerable pilot overhead.

Fortunately, UE typically does not suddenly and drastically change its velocity in practical scenarios, and the corresponding Doppler shift remains relatively stable during several channel estimations according to (3). Consequently, we can insert a small number of non-uniform pilots within the original estimation period T_{e}, such that the minimum time interval between pilots satisfies $T_{\mathrm{e,min}} < \frac{c}{f_c v}$ to accurately estimate the Doppler phase rotation within $T_{\mathrm{e,min}}$. Note that in order to simplify notation, we slightly abuse the definition of T_{e} as the original estimation period prior to inserting additional pilots in the sequel. Then due to the stationary nature of Doppler shift, the Doppler phase rotation within the original estimation period T_{e} can be obtained by extending the accurately estimated Doppler phase rotation within $T_{\mathrm{e,min}}$, thereby effectively overcoming Doppler aliasing.

Furthermore, under the constraint of $T_{\mathrm{e,min}} < \frac{c}{f_c v}$, the time distribution of the pilot pattern should facilitate the accurate reconstruction of continuous-time channels. Inspired by the successful application of Chebyshev polynomial roots in the field of function interpolation [41], we propose a non-uniform pilot pattern in the following lemma and theorem.

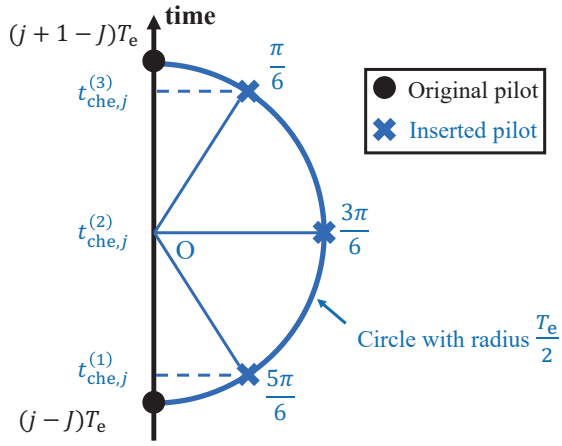


Fig. 3. Illustration of Chebyshev based non-uniform pilot pattern $\left\{t^{(n)}_{\mathrm{che},j}\right\}_{n=1}^{N}$, where number of inserted additional pilots is assumed to be $N = 3$, and $\left\{\frac{5\pi}{6}, \frac{3\pi}{6}, \frac{\pi}{6}\right\}$ correspond to the angles $\frac{2N-2n+1}{2N}\pi$ of $\left\{t^{(1)}_{\mathrm{che},j}, t^{(2)}_{\mathrm{che},j}, t^{(3)}_{\mathrm{che},j}\right\}$ in (12).

Lemma 1. *Among all choices of N distinct time slots $\mathcal{T} = \left\{t^{(n)}\right\}_{n=1}^{N} \subset [-1,\ 1]$, Chebyshev polynomial roots,*

$$t^{(n)}_{\mathrm{che}} = \cos\left(\frac{2N-2n+1}{2N}\pi\right),\ n \in \{1, 2, ..., N\}, \quad (9)$$

are optimal for reconstructing continuous-time functions in $C^N\left([-1,\ 1]\right)$ under the criterion of maximum norm.

Proof: Let $h(t), \widehat{h}(t) \in C^N\left([-1,\ 1]\right)$ denote the objective function and reconstructed function, respectively. Since $h(t)$ is unknown in practical scenarios, the specific $\widehat{h}(t)$ cannot be determined by theoretical analysis. In contrast, the time slots $\mathcal{T}$ can be preset, and the works [42], [43] have proved

$$\|h(t) - \widehat{h}(t)\|_\infty \propto \max_{t\in[-1,1]}\left|\prod_{n=1}^{N}(t - t^{(n)})\right|,\ t \in [-1,\ 1]. \quad (10)$$

Furthermore, it can be derived [41]

$$\left\{t^{(n)}_{\mathrm{che}}\right\}_{n=1}^{N} = \arg\min_{\left\{t^{(n)}\right\}_{n=1}^{N}} \max_{t\in[-1,1]}\left|\prod_{n=1}^{N}(t - t^{(n)})\right|, \quad (11)$$

which means that the choice of $\mathcal{T} = \left\{t^{(n)}_{\mathrm{che}}\right\}_{n=1}^{N}$ optimally facilitates the reconstruction of $h(t)$. ■

Theorem 2. *Within the time interval between the j-th and $(j+1)$-th channel estimations $\mathcal{T}_j = [(j-J)T_{\mathrm{e}}, (j+1-J)T_{\mathrm{e}}]$, the time slots of inserted pilots $\mathcal{T}_{\mathrm{che},j} = \left\{t^{(n)}_{\mathrm{che},j}\right\}_{n=1}^{N}$ with*

$$t^{(n)}_{\mathrm{che},j} = \frac{(2j-2J+1)T_{\mathrm{e}}}{2} + \frac{T_{\mathrm{e}}}{2}\cos\left(\frac{2N-2n+1}{2N}\pi\right), \quad (12)$$

are optimal for the continuous-time channel reconstruction under Doppler-dominated channel variations with relatively stable UE velocity, where the minimum number of inserted pilots N_{min} to avoid Doppler aliasing needs to satisfy

$$N \geq N_{\mathrm{min}} = \left\lceil \frac{\pi}{2\arccos\left(1 - \frac{2c}{f_c v T_{\mathrm{e}}}\right)} \right\rceil. \quad (13)$$

Proof: (12) is the affine transformation of (9) from $[-1,\ 1]$ to $\mathcal{T}_j$, thereby preserving the optimality in reconstructing continuous-time functions. Due to the curve shape of the cosine function used in (12), the pilots are densely configured at both ends but relatively sparsely deployed in the middle of $\mathcal{T}_j$. An illustration of $t_{\mathrm{che},j}^{(n)}$ is presented in Fig. 3. The specific minimum time interval is

$$T_{\mathrm{che,min}} = \frac{T_{\mathrm{e}}}{2} - \frac{T_{\mathrm{e}}}{2}\cos\left(\frac{\pi}{2N}\right), \tag{14}$$

which is the time interval from $(j-J)T_{\mathrm{e}}$ to $t_{\mathrm{che},j}^{(1)}$ or from $t_{\mathrm{che},j}^{(N)}$ to $(j+1-J)T_{\mathrm{e}}$. Based on Theorem 1, the minimum number of inserted pilots $N_{\min}$ needs to satisfy (13) to avoid Doppler aliasing. ∎

Remark 3: The optimality of the Chebyshev based non-uniform pilot pattern relies on the assumption of Doppler-dominated channel variations with relatively stable UE velocity, which is generally realistic for the short estimation interval T_{e}. Specifically, when considering a typical estimation interval in 3GPP specifications, e.g., $T_{\mathrm{e}} = 40\,\mathrm{ms}$ [40], the UE velocity of $v = 60\,\mathrm{km/h}$, and a high acceleration of $a = 4\,\mathrm{m/s^2}$ [44], [45], the maximum variation of UE velocity is around $aT_{\mathrm{e}} = 0.16\,\mathrm{m/s}$, which is negligible, demonstrating that the UE velocity is relatively stable within a short estimation interval T_{e}. Besides, the maximum UE displacement is around $vT_{\mathrm{e}} = 0.67\,\mathrm{m}$, which is very small compared to the typical distance between base station (BS) and UE. Therefore, the birth-death process of multiple paths and the changes in scattering environments can be ignored, and the dominant factor for channel variations is the Doppler shift [7], [12], [14], [46]. In brief, when considering a short estimation interval T_{e}, the assumption of Doppler-dominated channel variations with relatively stable UE velocity is usually satisfied, and our optimality derivation in Theorem 2 holds. Nevertheless, when the estimation interval T_{e} is long, the scattering environment may experience rapid changes, or the UE velocity may vary abruptly. In these scenarios, the optimality of the Chebyshev based non-uniform pilot pattern may not strictly hold, and the pilot pattern should instead be configured according to the sensed scattering environment complexity and the UE velocity.

Remark 4: The works [47], [48] proposed the deep learning based joint design of pilot signal and channel estimation, where the channel is compressed into the low-dimensional received signal by a fully-connected (FC) layer, and its learnable weights are regarded as the pilot signal. Furthermore, the least significant pilots are removed over different orthogonal frequency division multiplexing (OFDM) symbols and subcarriers by neural network pruning in [48], which can reduce the pilot overhead at the cost of only marginal estimation performance degradation. However, the above pilot design and pruning aim at estimating the channel at a given time slot, and cannot be directly generalized to our work, as our focus is pilot allocation over different time slots to accurately capture the temporal correlations within channel sequences. On the other hand, considering the capability of deep learning to sense temporal channel variations in a data-driven manner, the integration of adaptive pilot allocation over different time slots is worthy of investigation in future works. The proposed Chebyshev based non-uniform pilot pattern can serve as an effective initialization for the adaptive pilot allocation, since it mitigates Doppler aliasing and facilitates accurate channel prediction.

According to Theorem 2, the finally designed non-uniform pilot pattern is $\mathcal{T}_{\mathrm{che}} = \left\{(1-J)T_{\mathrm{e}}, t_{\mathrm{che},1}^{(1)}, t_{\mathrm{che},1}^{(2)}, \cdots, t_{\mathrm{che},1}^{(N)}, (2-J)T_{\mathrm{e}}, \cdots, 0\right\}$ with length $\widetilde{J} = J + (J-1)N$. For convenience, the elements of $\mathcal{T}_{\mathrm{che}}$ are renumbered as $\mathcal{T}_{\mathrm{che}} = \left\{t_{\mathrm{che},1}, t_{\mathrm{che},2}, \cdots, t_{\mathrm{che},\widetilde{J}}\right\}$, and we present the following corollary to show that $\mathcal{T}_{\mathrm{che}}$ can provide the finer estimation resolution of Doppler phase than the uniform pilot pattern with the same overhead as $\mathcal{T}_{\mathrm{uni}} = \left\{(1-J)T_{\mathrm{e}}, (\frac{N+2}{N+1}-J)T_{\mathrm{e}}, (\frac{N+3}{N+1}-J)T_{\mathrm{e}}, \cdots, 0\right\}$.

Corollary 1. *Given the same length of channel estimations $\widetilde{J}$, the minimum time intervals of $\mathcal{T}_{\mathrm{che}}$ and $\mathcal{T}_{\mathrm{uni}}$ satisfy*

$$T_{\mathrm{che,min}} \le T_{\mathrm{uni,min}},\ \forall N \in \mathbb{N}_+, \tag{15}$$

where the equality holds if and only if $N = 1$.

Proof: See Appendix A. ∎

In summary, the choice of $\mathcal{T}_{\mathrm{che}}$ is optimal for the reconstruction of continuous-time channels based on Theorem 2, and enjoys stronger capability to mitigate Doppler aliasing than $\mathcal{T}_{\mathrm{uni}}$ of the same overhead according to Corollary 1.

IV. Element-Wise Continuous-Time Transformer

Utilizing the historical channel sequence estimated at $\mathcal{T}_{\mathrm{che}}$, the continuous-time channel prediction in (6) can be further reformulated as

$$\left\{\widehat{\boldsymbol{h}}(t_p)\right\}_{p=1}^{P} = g_{\mathrm{con}}\left(\left\{\widetilde{\boldsymbol{h}}(t_{\mathrm{che},j}), t_{\mathrm{che},j}\right\}_{j=1}^{\widetilde{J}}, \{t_p\}_{p=1}^{P}\right). \tag{16}$$

In this section, we focus on the design of $g_{\mathrm{con}}(\cdot)$. First, we introduce the continuous-time attention mechanism, which extends the standard attention mechanism to the continuous-time domain by incorporating neural ODE. Next, a high-frequency temporal encoding is proposed to accurately fit the rapidly time-varying channels. Then, the overall architecture of the proposed continuous-time transformer based on the element-wise prediction mechanism is presented. For clarification, the definitions of notations used in Section IV are listed in Table III.

A. Continuous-Time Attention Mechanism

As the fundamental component of transformer, the attention mechanism enables the efficient extraction of temporal correlations within the sequence data. Specifically, let the input sequence be $\mathbf{X} = \left[\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_L\right] \in \mathbb{R}^{d_{\mathrm{i}} \times L}$ with input feature dimension d_{i} and sequence length L. The corresponding sampling time slots are $\{\tau_1, \tau_2, \cdots, \tau_L\}$. Given the output feature dimension d_{o}, we further define the queries, keys, and values observed at $\{\tau_1, \tau_2, \cdots, \tau_L\}$ as $\{\mathbf{q}_l\}_{l=1}^{L}$, $\{\mathbf{k}_l\}_{l=1}^{L}$, and $\{\mathbf{v}_l\}_{l=1}^{L}$, where $\mathbf{q}_l, \mathbf{k}_l, \mathbf{v}_l \in \mathbb{R}^{d_{\mathrm{o}} \times 1}$ represent the query, key, and value observations at the time slot τ_l, respectively. They can be obtained by $\mathbf{q}_l = \mathbf{W}^{\mathrm{q}}\mathbf{x}_l$, $\mathbf{k}_l = \mathbf{W}^{\mathrm{k}}\mathbf{x}_l$, and $\mathbf{v}_l = \mathbf{W}^{\mathrm{v}}\mathbf{x}_l$ with the corresponding trainable matrices

TABLE III
DEFINITIONS OF NOTATIONS.

Notations	Definitions
$\{t_{\mathrm{che},j}\}_{j=1}^{\bar{J}}/\{t_p\}_{p=1}^{P}$	Estimation/Prediction time slots
$\mathbf{X}/\mathbf{X}_{\mathrm{tok}}/\mathbf{X}_{\mathrm{pos}}$	Input/Token embedding/Position embedding
$\{\tau_l\}_{l=1}^{L}$	Sampling time slots for query/key/value
$\mathbf{q}_l/\mathbf{k}_l/\mathbf{v}_l$	Query/Key/Value observations at τ_l
$\mathbf{q}_l(t)/\mathbf{k}_l(t)/\mathbf{v}_l(t)$	Query/Key/Value trajectories based on $\mathbf{q}_l/\mathbf{k}_l/\mathbf{v}_l$
c_{l_1,l_2}	Standard attention weight at key time slot τ_{l_1} and query time slot τ_{l_2}
$\mathrm{c}(\tau_{l_1},\tau_{l_2})$	Continuous-time attention weight at key time slot τ_{l_1} and query time slot τ_{l_2}
$\overline{\mathbf{v}}_{l_1}(\tau_{l_2})$	Average temporal influence exerted by $\mathbf{v}_{l_1}(t)$ from value time slot τ_{l_1} to query time slot τ_{l_2}
$\mathbf{o}_{l_2}$	Output of standard attention at query time slot τ_{l_2}
$\mathbf{o}(\tau_{l_2})$	Output of continuous-time attention at query time slot τ_{l_2}

$\mathbf{W}^{\mathrm{q}},\mathbf{W}^{\mathrm{k}},\mathbf{W}^{\mathrm{v}}\in\mathbb{R}^{d_{\mathrm{o}}\times d_{\mathrm{i}}}$. Then, the standard attention weights $\{\mathrm{c}_{l_1,l_2}\}_{l_1=1}^{L}$, which denote the temporal correlations between all key time slots $\{\tau_{l_1}\}_{l_1=1}^{L}$ to a given query time slot τ_{l_2}, $l_2\in\{1,2,\cdots,L\}$, can be calculated by [12]

$$\{\mathrm{c}_{l_1,l_2}\}_{l_1=1}^{L}=\sigma\left(\left\{\frac{(\mathbf{k}_{l_1})^{\mathrm{T}}\mathbf{q}_{l_2}}{\sqrt{d_{\mathrm{o}}}}\right\}_{l_1=1}^{L}\right),\tag{17}$$

where $\sigma(\cdot)$ denotes the softmax operation, and $\frac{1}{\sqrt{d_{\mathrm{o}}}}$ is the scaling factor for normalization. Finally, the output at the given query time slot τ_{l_2}, $\mathbf{o}_{l_2}\in\mathbb{R}^{d_{\mathrm{o}}\times 1}$, can be formulated as

$$\mathbf{o}_{l_2}=\sum_{l_1=1}^{L}\mathbf{v}_{l_1}\mathrm{c}_{l_1,l_2},\tag{18}$$

which is the linear combination of the values $\{\mathbf{v}_{l_1}\}_{l_1=1}^{L}$ based on the corresponding attention weights $\{\mathrm{c}_{l_1,l_2}\}_{l_1=1}^{L}$. According to (17) and (18), as the alignment between $\mathbf{q}_{l_2}$ and $\mathbf{k}_{l_1}$ becomes closer, the attention weight c_{l_1,l_2} increases, indicating that output $\mathbf{o}_{l_2}$ has stronger correlation with value $\mathbf{v}_{l_1}$.

However, since the standard attention mechanism does not model the specific time interval, it is restricted to the prediction with uniform time scales. To overcome this limitation, a continuous-time attention mechanism is designed by introducing neural ODE [49]–[52]. Specifically, the discrete-time observations in the standard attention mechanism $\{\mathbf{q}_l\}_{l=1}^{L}$, $\{\mathbf{k}_l\}_{l=1}^{L}$, and $\{\mathbf{v}_l\}_{l=1}^{L}$ are extended to their corresponding continuous-time trajectories $\{\mathbf{q}_l(t)\}_{l=1}^{L}$, $\{\mathbf{k}_l(t)\}_{l=1}^{L}$, and $\{\mathbf{v}_l(t)\}_{l=1}^{L}$, where $\mathbf{q}_l(t)\in\mathbb{R}^{d_{\mathrm{o}}\times 1}$ represents a query trajectory at time slot t based on the observation $\mathbf{q}_l$, while $\mathbf{k}_l(t)$ and $\mathbf{v}_l(t)$ follow the similar formulation. Taking $\mathbf{q}_l(t)$ as an example, its continuous-time modeling can be implemented using two distinct schemes. One scheme is to adopt the closed-form interpolation function $f_{\mathrm{interp}}(\cdot)$, which can be expressed as

$$\mathbf{q}_l(t)=f_{\mathrm{interp}}\Big(\{\mathbf{q}_l,\tau_l\}_{l=1}^{L}\Big),\tag{19}$$

where $f_{\mathrm{interp}}(\cdot)$ can be realized by linear interpolation, spline interpolation, or other interpolation methods. The other scheme is based on neural ODE, which can be written as

$$\mathbf{q}_l(t)=\int_{\tau_l}^{t}f_{\mathrm{ODE}}\big(\mathbf{q}_l(z),z\big)\mathrm{d}z+\mathbf{q}_l,\tag{20}$$

where $f_{\mathrm{ODE}}(\cdot)$ is the neural network that fits the derivative of $\mathbf{q}_l(t)$, i.e., $f_{\mathrm{ODE}}\big(\mathbf{q}_l(z),z\big)=\frac{\mathrm{d}\mathbf{q}_l(z)}{\mathrm{d}z}$, and the integration can be approximated by numerical methods, such as Euler and Runge-Kutta methods. On the one hand, neural ODE can adaptively approximate the dynamics of $\mathbf{q}_l(t)$, offering superior capability in continuous-time modeling. On the other hand, neural ODE suffers from high computational complexity due to the integral operations [51], whereas the closed-form interpolation function based modeling allows the rapid calculation of $\mathbf{q}_l(t)$ at arbitrary time slots. In the attention mechanism, queries and keys only need to capture the temporal correlations, while values are responsible for directly modeling the complex relationships between the input and output according to (17) and (18). Consequently, we use the interpolation function $f_{\mathrm{interp}}(\cdot)$ in (19) to efficiently construct $\mathbf{q}_l(t)$ and $\mathbf{k}_l(t)$, and utilize neural ODE in (20) to accurately fit $\mathbf{v}_l(t)$.

To sense the variations of time scales, we extend the standard attention in (17) to the continuous-time domain by explicitly embedding time information, which is given by

$$\{\mathrm{c}(\tau_{l_1},\tau_{l_2})\}_{l_1=1}^{L}=\begin{cases}\sigma\left(\left\{\frac{\int_0^{\tau_{l_2}-\tau_{l_1}}\big(\mathbf{k}_{l_1}(t+\tau_{l_1})\big)^{\mathrm{T}}\mathbf{q}_{l_2}(-t+\tau_{l_2})\mathrm{d}t}{(\tau_{l_2}-\tau_{l_1})\sqrt{d_{\mathrm{o}}}}\right\}_{l_1=1}^{L}\right), & \text{if } \tau_{l_2}\neq\tau_{l_1},\\ \sigma\left(\left\{\frac{(\mathbf{k}_{l_1})^{\mathrm{T}}\mathbf{q}_{l_2}}{\sqrt{d_{\mathrm{o}}}}\right\}_{l_1=1}^{L}\right), & \text{otherwise,}\end{cases}\tag{21}$$

where $\{\mathrm{c}(\tau_{l_1},\tau_{l_2})\}_{l_1=1}^{L}$ denote the continuous-time attention weights between all key time slots $\{\tau_{l_1}\}_{l_1=1}^{L}$ to a given query time slot τ_{l_2}, $l_2\in\{1,2,\cdots,L\}$. When $\tau_{l_2}\neq\tau_{l_1}$, $\mathrm{c}(\tau_{l_1},\tau_{l_2})$ quantifies the degree of alignment between the query trajectory $\mathbf{q}_{l_2}(t)$ and the key trajectory $\mathbf{k}_{l_1}(t)$ within the time interval between 0 and $(\tau_{l_2}-\tau_{l_1})$. The normalized factor $\frac{1}{\tau_{l_2}-\tau_{l_1}}$ is utilized to ensure that the continuous-time attention concentrates on modeling the alignment between the query and key trajectories rather than the time interval. When $\tau_{l_2}=\tau_{l_1}$, $\mathrm{c}(\tau_{l_1},\tau_{l_2})$ is equal to c_{l_1,l_1} in (17) to maintain its continuity. The integration in $\mathrm{c}(\tau_{l_1},\tau_{l_2})$ can be solved by classical numerical approximation schemes, such as trapezoidal integration or Gauss–Legendre quadrature, which can be written as

$$\begin{aligned}\int_0^{\tau_{l_2}-\tau_{l_1}}&\big(\mathbf{k}_{l_1}(t+\tau_{l_1})\big)^{\mathrm{T}}\mathbf{q}_{l_2}(-t+\tau_{l_2})\mathrm{d}t\approx\\ &\sum_{e=1}^{E}\gamma_e\big(\mathbf{k}_{l_1}(\tau_{l_e}+\tau_{l_1})\big)^{\mathrm{T}}\mathbf{q}_{l_2}(-\tau_{l_e}+\tau_{l_2}),\end{aligned}\tag{22}$$

where E is the number of integration points, γ_e is the integration coefficient, and τ_{l_e} is the integration point within the time interval between 0 and $(\tau_{l_2}-\tau_{l_1})$. The specific γ_e and τ_{l_e} depend on the selection of numerical approximation schemes [53], while $\mathbf{q}_{l_2}(-\tau_{l_e}+\tau_{l_2})$ and $\mathbf{k}_{l_1}(\tau_{l_e}+\tau_{l_1})$ can be

obtained by respectively setting $t=-\tau_{l_e}+\tau_{l_2}$ and $t=\tau_{l_e}+\tau_{l_1}$ in (19).

Similarly, the continuous-time values can be calculated by

$$\overline{\mathbf{v}}_{l_1}(\tau_{l_2})=\begin{cases}\frac{\int_{\tau_{l_1}}^{\tau_{l_2}}\mathbf{v}_{l_1}(t)\mathrm{d}t}{\tau_{l_2}-\tau_{l_1}}, & \text{if } \tau_{l_2}\neq\tau_{l_1},\\ \mathbf{v}_{l_1}, & \text{otherwise,}\end{cases} \tag{23}$$

where $\overline{\mathbf{v}}_{l_1}(\tau_{l_2})$ denotes the average temporal influence exerted by the value trajectory $\mathbf{v}_{l_1}(t)$ from its sampling time slot τ_{l_1} to the query time slot τ_{l_2}. Utilizing $\{\overline{\mathbf{v}}_{l_1}(\tau_{l_2})\}_{l_1=1}^{L}$ and the corresponding attention weights $\{\mathrm{c}(\tau_{l_1},\tau_{l_2})\}_{l_1=1}^{L}$, the final output of continuous-time attention mechanism at the query time slot τ_{l_2}, $\mathbf{o}(\tau_{l_2})\in\mathbb{R}^{d_\mathrm{o}\times 1}$, can be expressed as

$$\mathbf{o}(\tau_{l_2})=\sum_{l_1=1}^{L}\overline{\mathbf{v}}_{l_1}(\tau_{l_2})\mathrm{c}(\tau_{l_1},\tau_{l_2}). \tag{24}$$

It should be noticed that since the calculation of the proposed continuous-time attention at distinct query time slots are mutually independent, we can adopt parallel processing similar to the standard attention for efficient computation [12].

Remark 5: According to (22), the calculation of continuous-time attention in (21) contains two steps. Firstly, both $\mathbf{q}_{l_2}(-\tau_{l_e}+\tau_{l_2})$ and $\mathbf{k}_{l_1}(\tau_{l_e}+\tau_{l_1})$ are modeled via closed-form interpolation functions (e.g., linear interpolation), and thus they can be rapidly obtained in $\mathcal{O}(d)$ computational complexity from the observations $\{\mathbf{q}_{l_2}\}_{l_2=1}^{L}$ and $\{\mathbf{k}_{l_1}\}_{l_1=1}^{L}$, respectively [42]. Furthermore, since τ_{l_e} is a function of both τ_{l_1} and τ_{l_2} [53], and the attention mechanism is required to obtain $\{\mathbf{q}_{l_2}(-\tau_{l_e}+\tau_{l_2})\}_{e=1}^{E}$ and $\{\mathbf{k}_{l_1}(\tau_{l_e}+\tau_{l_1})\}_{e=1}^{E}$ for $\forall l_1,l_2\in[1,L]$, the corresponding computational complexity is $\mathcal{O}(L^2dE)+\mathcal{O}(L^2dE)=\mathcal{O}(L^2dE)$. Secondly, by utilizing the calculated $\{\mathbf{q}_{l_2}(-\tau_{l_e}+\tau_{l_2})\}_{e=1}^{E}$ and $\{\mathbf{k}_{l_1}(\tau_{l_e}+\tau_{l_1})\}_{e=1}^{E}$ for $\forall l_1,l_2\in[1,L]$, the computational complexity of the dot-product formulation in (22) is $\mathcal{O}(L^2dE)$. In summary, the overall complexity of these two steps for calculating the continuous-time attention in (21) is $\mathcal{O}(L^2dE)+\mathcal{O}(L^2dE)=\mathcal{O}(L^2dE)$, which is larger than $\mathcal{O}(L^2d)$ of the standard attention in (17) due to incorporating integral operators.

Remark 6: When the sequence length L and the feature dimension d are large, the computational complexity of continuous-time attention $\mathcal{O}(L^2dE)$ becomes high. To solve this issue, some potential techniques, such as low-rank kernel approximation [54], [55], sparse attention [18], and patching [34], can be used to alleviate the computational burden.

B. High-Frequency Temporal Encoding

Since the time slots are incorporated into the computation of queries, keys, and values, the continuous-time attention has the potential capability to sense the variations of time scales. However, this capability is usually insufficient to achieve accurate continuous-time prediction for two reasons. Firstly, similar to the standard attention, the continuous-time attention remains permutation-invariant [19], which implies that the order information of sequences is not utilized by attention mechanisms, thereby limiting prediction performance. Secondly, wireless channels exhibit the features of rapidly periodic variations in high-mobility scenarios, which are difficult for attention mechanisms to directly learn [16], [56], [57].

To solve the first problem, existing transformer models adopt the positional encoding to retain the order information of sequences. Specifically, the input of attention mechanism is given by $\mathbf{X}=\mathbf{X}_{\mathrm{tok}}+\mathbf{X}_{\mathrm{pos}}$, where $\mathbf{X}_{\mathrm{tok}}\in\mathbb{R}^{d_\mathrm{i}\times L}$ is the token embedding from the sequence data and $\mathbf{X}_{\mathrm{pos}}\in\mathbb{R}^{d_\mathrm{i}\times L}$ is the position embedding from the positional encoding. The standard positional encoding utilizes the sine and cosine functions to represent the order information, which can be expressed as [12]

$$[\mathbf{X}_{\mathrm{pos}}]_{i,l}=\begin{cases}\sin\left(\frac{(l-1)}{10000^{i/d_\mathrm{i}}}\right), & \text{if } i \text{ is even number,}\\ \cos\left(\frac{(l-1)}{10000^{i/d_\mathrm{i}}}\right), & \text{otherwise,}\end{cases} \tag{25}$$

with $i\in\{0,1,\cdots,d_\mathrm{i}-1\}$. It can be seen that the fixed order information $l\in\{1,2,\cdots,L\}$ is encoded into $\mathbf{X}_{\mathrm{pos}}$ to break the permutation-invariant property of attention mechanisms. However, the standard positional encoding cannot sense the variations of time scales in continuous-time prediction.

In contrast, we propose the temporal encoding to explicitly embed the time slot τ_l, $\tau_l\in\{\tau_1,\tau_2,\cdots,\tau_L\}$, instead of the order l, thereby effectively modeling the variation of time scales. Besides, in order to efficiently learn the features of rapidly periodic channel dynamics in high-mobility scenarios, we further extend the temporal encoding to the high frequency, and the corresponding temporal embedding $\mathbf{X}_{\mathrm{tem}}\in\mathbb{R}^{d_\mathrm{i}\times L}$ can be written as

$$[\mathbf{X}_{\mathrm{tem}}]_{i,l}=\begin{cases}\sin\left(\frac{\omega_0\tau_l}{10000^{i/d_\mathrm{i}}}\right), & \text{if } i \text{ is even number,}\\ \cos\left(\frac{\omega_0\tau_l}{10000^{i/d_\mathrm{i}}}\right), & \text{otherwise,}\end{cases} \tag{26}$$

where ω_0 denotes the frequency hyperparameter, and is set to a relatively large value. In this manner, $[\mathbf{X}_{\mathrm{tem}}]_{i,l}$ varies rapidly with τ_l, and can be regarded as a predefined, rapidly periodic temporal component that drives the attention mechanism to effectively capture the corresponding channel dynamics [58], [59].

C. Continuous-Time Transformer Architecture with Element-Wise Prediction

In this subsection, we present how to utilize the aforementioned continuous-time attention mechanism with the high-frequency temporal encoding to construct the continuous-time transformer based on the element-wise prediction.

First, in order to fully exploit the powerful feature extraction capability of deep learning, the estimated channel vector $\widetilde{\boldsymbol{h}}(t_{\mathrm{che},j})$ needs to be transformed to the angle domain via discrete Fourier transform (DFT), which can be expressed as

$$\widetilde{\boldsymbol{h}}_{\mathrm{A}}(t_{\mathrm{che},j})=\mathbf{F}_{\mathrm{A}}\widetilde{\boldsymbol{h}}(t_{\mathrm{che},j}), \tag{27}$$

where $\widetilde{\boldsymbol{h}}_{\mathrm{A}}(t_{\mathrm{che},j})\in\mathbb{C}^{M\times 1}$ denotes the corresponding angle-domain channel vector, and $\mathbf{F}_{\mathrm{A}}=\mathrm{diag}(\mathbf{F}_{\mathrm{DFT}},\mathbf{F}_{\mathrm{DFT}})\in\mathbb{C}^{M\times M}$ represents the angle-domain unitary matrix for dual-polarized directions with the DFT matrix $\mathbf{F}_{\mathrm{DFT}}\in\mathbb{C}^{\frac{M}{2}\times\frac{M}{2}}$.

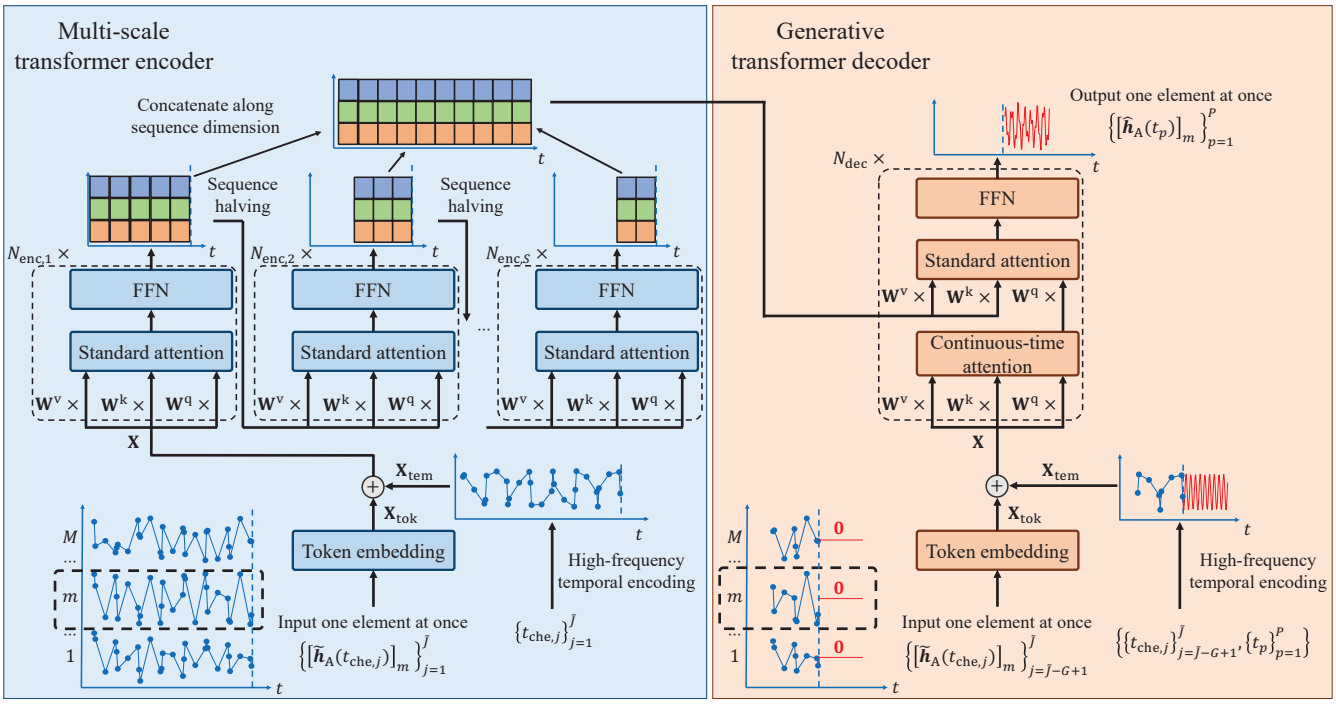


Fig. 4. Overall architecture of proposed continuous-time transformer with element-wise prediction mechanism, including multi-scale transformer encoder and generative transformer decoder.

The existing channel prediction methods typically feed the entire channel vector $\widetilde{\boldsymbol{h}}_{\mathrm{A}}(t_{\mathrm{che},j})$ into the network at once and jointly predict all of its elements by leveraging spatial and temporal correlations, which is called vector-wise prediction mechanism [28]–[31]. Under this mechanism, the prediction of different elements in $\widetilde{\boldsymbol{h}}_{\mathrm{A}}(t_{\mathrm{che},j})$ is forced to share the same attention weights. However, due to the varying Doppler shifts across paths caused by their different azimuth and elevation arrival angles in (3), the temporal variations for the elements of the angle-domain channel vector $\widetilde{\boldsymbol{h}}_{\mathrm{A}}(t_{\mathrm{che},j})$ corresponding to different paths may differ significantly. This mismatch between the shared attention weights and the element-specific dynamics may severely impair the extraction of temporal correlations. Besides, with the growth of antenna number in massive MIMO systems, the spatial correlations among different elements in $\widetilde{\boldsymbol{h}}_{\mathrm{A}}(t_{\mathrm{che},j})$ become weaker [7]. Therefore, utilizing the spatial correlations may inadvertently introduce excessive irrelevant information from other elements into the target element and lead to overfitting.

Motivated by the effectiveness of the channel-independent framework in time-series modeling [19], [34], we propose an element-wise prediction mechanism to resolve the above issues. Specifically, instead of the whole vector $\widetilde{\boldsymbol{h}}_{\mathrm{A}}(t_{\mathrm{che},j})$, each element $[\widetilde{\boldsymbol{h}}_{\mathrm{A}}(t_{\mathrm{che},j})]_m$ for $m \in \{1, 2, \cdots, M\}$ is fed into the neural network separately. In this way, each element can generate its own attention weights to accurately capture temporal correlations, while the risk of overfitting can be mitigated. Moreover, the prediction of $[\widetilde{\boldsymbol{h}}_{\mathrm{A}}(t_{\mathrm{che},j})]_m$ for $m \in \{1, 2, \cdots, M\}$ in our element-wise mechanism shares the same network parameters, which has two advantages. Firstly, the model storage overhead can be significantly reduced. Secondly, the computation of M elements can be parallelized by extending the batch dimension from B to BM, thereby improving computational efficiency.

To effectively process $\left[\widetilde{\boldsymbol{h}}_{\mathrm{A}}(t_{\mathrm{che},j})\right]_m$, the network architecture of our continuous-time transformer is shown in Fig. 4, which consists of two parts: the multi-scale transformer encoder and the generative transformer decoder. The motivation behind our network architecture design lies in the characteristics of estimation and prediction time slots. Specifically, the estimation time slots $\{t_{\mathrm{che},j}\}_{j=1}^{\widetilde{J}}$ remain fixed in the proposed non-uniform pilot pattern, while the prediction time slots $\{t_p\}_{p=1}^{P}$ are arbitrary for continuous-time prediction. Therefore, instead of the encoder-only structure [14], the encoder–decoder structure is adopted to explicitly separate the extraction of temporal correlations at fixed estimation time slots $\{t_{\mathrm{che},j}\}_{j=1}^{\widetilde{J}}$ and arbitrary prediction time slots $\{t_p\}_{p=1}^{P}$ into the encoder and decoder, respectively. In this way, the encoder can pay attention to consistently and efficiently learning temporal dependencies from the fixed estimation time slots $\{t_{\mathrm{che},j}\}_{j=1}^{\widetilde{J}}$. The specific details are elaborated in the following.

1) Multi-Scale Transformer Encoder: $\left[\widetilde{\boldsymbol{h}}_{\mathrm{A}}(t_{\mathrm{che},j})\right]_m$ is first processed by token embedding, which is composed of linear layers. As described in Subsection IV-B, the output of token embedding is added to the result of high-frequency temporal encoding at the estimation time slots $\{t_{\mathrm{che},j}\}_{j=1}^{\widetilde{J}}$, serving as the input to the standard attention. After extracting the temporal correlations via (17) and (18), a feed-forward network (FFN), which consists of multi-layer perceptrons (MLPs), residual connections, and layer normalization, is subsequently employed to refine and aggregate the features. This cascaded structure of standard attention and FFN is repeated $N_{\mathrm{enc},1}$ times to fully capture features. However, it should be noticed that although $\{t_{\mathrm{che},j}\}_{j=1}^{\widetilde{J}}$ is fixed, the nature of its non-uniform distribution necessitates the extraction of temporal features across multiple scales. To extract these multi-scale features, we

do not employ the continuous-time attention. This is because the estimation length $\widetilde{J}$ is usually long to ensure the acquisition of comprehensive historical information, which results in substantial computational overhead for continuous-time attention as discussed in Remark 6. In contrast, considering that the estimation time slots $\{t_{\mathrm{che},j}\}_{j=1}^{\widetilde{J}}$ are fixed, the multi-scale encoder stacking strategy with the standard attention [18] is sufficient to capture multi-scale temporal features with low computational cost. Specifically, we further stack the above cascaded structure with stack number S, where the s-th stack uses the latter half slice of the output sequence from the $(s-1)$-th stack as the input. This design facilitates a shift in the extraction of temporal correlations from the full sequence to the finer-scale subsequence. Finally, the output sequences of S stacks are concatenated along the sequence dimension, which contain multi-scale time information, enabling the effective representation of temporal features at the non-uniform estimation time slots $\{t_{\mathrm{che},j}\}_{j=1}^{\widetilde{J}}$.

2) Generative Transformer Decoder: The input of generative transformer decoder is the combination of historical channel sequence with length G and zero padding with length P, $\left\{\left[\widetilde{\boldsymbol{h}}_{\mathrm{A}}(t_{\mathrm{che},\widetilde{J}-G+1})\right]_m,\cdots,\left[\widetilde{\boldsymbol{h}}_{\mathrm{A}}(t_{\mathrm{che},\widetilde{J}})\right]_m,0,\cdots,0\right\}$, where the objective is to generate the prediction output at the time slots of these zero values in parallel. Specifically, similar to the encoder, the decoder input is processed by token embedding, and its output is summed with the output of high-frequency temporal encoding. Furthermore, since the decoder needs to extract the temporal features at arbitrary time scales for continuous-time prediction, the continuous-time attention as detailed in Subsection IV-A is utilized. The corresponding output is used to query the correlations between the future prediction and the historical information captured from the multi-scale transformer encoder, while a FFN is subsequently cascaded to integrate features. The above cascaded structure is repeated N_{dec} times, and the final output is the prediction result in the angle domain $\left\{\left[\widehat{\boldsymbol{h}}_{\mathrm{A}}(t_p)\right]_m\right\}_{p=1}^{P}$.

After predicting all M elements, our objective $\left\{\widehat{\boldsymbol{h}}(t_p)\right\}_{p=1}^{P}$ can be obtained by applying the inverse transform in (27).

D. Model Training and Performance Evaluation

The normalized mean squared error (NMSE) is widely used to optimize channel prediction models and evaluate their performance [12], [30], which can be formulated as

$$\mathrm{NMSE}(t_p)=\frac{\|\boldsymbol{h}(t_p)-\widehat{\boldsymbol{h}}(t_p)\|_2^2}{\|\boldsymbol{h}(t_p)\|_2^2},\ 1\le p\le P,$$
$$\mathrm{NMSE}=\frac{1}{P}\sum_{p=1}^{P}\mathrm{NMSE}(t_p). \tag{28}$$

To facilitate performance comparison in the model test stage, given the prediction period T_{p}, the prediction time slots $\{t_p\}_{p=1}^{P}$ are set to $t_p=pT_{\mathrm{p}}$ for $p\in\{1,2,\cdots,P\}$. Conversely, in the model training stage, the prediction time slots $\{t_p\}_{p=1}^{P}$ are uniformly sampled from $(0,PT_{\mathrm{p}}]$, enabling the prediction capability across arbitrary time scales [29], [30].

TABLE IV
CHANNEL PARAMETERS OF 3GPP TR 38.901 DATASET.

Parameters	Values
Cell radius r_{c}	300 m
BS height h_{BS}	25 m
Minimum/maximum UE heights $h_{\mathrm{UE,min}}/h_{\mathrm{UE,max}}$	1.5/22.5 m
Center frequency f_{c}	3.5 GHz
Number of paths L_{p}	12
Delay spread τ_{s}	98.3 ns
Number of BS horizontal/vertical antennas $M_{\mathrm{h}}/M_{\mathrm{v}}$	4/4
Number of BS antennas M	32

TABLE V
STRUCTURES OF PROPOSED CONTINUOUS-TIME TRANSFORMER.

Models	Layers	Structures
Multi-scale transformer encoder	Token embedding	$d_{\mathrm{i}}=2, d_{\mathrm{o}}=64$
	Standard attention	$N_{\mathrm{head}}=8, S=2$, $N_{\mathrm{enc},1}=2, N_{\mathrm{enc},2}=1$, $d_{\mathrm{i}}=64, d_{\mathrm{o}}=64$, FFN, LayerNorm
Generative transformer decoder	Token embedding	$d_{\mathrm{i}}=2, d_{\mathrm{o}}=64$
	Continuous-time attention	$N_{\mathrm{head}}=8, N_{\mathrm{dec}}=2$, $d_{\mathrm{i}}=64, d_{\mathrm{o}}=64$, FFN, LayerNorm
	Neural ODE	$d_{\mathrm{i,t}}=1, d_{\mathrm{i,f}}=64$, $d_{\mathrm{o}}=64$, LayerNorm, Tanh

V. SIMULATION STUDY

A. Simulation Setup

The channel model of 3GPP TR 38.901 for urban macro-cell scenarios is adopted to generate the data set in our simulations [36]. This channel model contains both line-of-sight (LOS) and non-line-of-sight (NLOS) scenarios, where the LOS probability is determined by the UE height h_{UE} and the 2D distance d_{2D} between the BS and UE. The specific channel parameters are listed in Table IV. Unless otherwise stated, the UE velocity is set to $v=60$ km/h, and the signal-to-noise ratio (SNR) of channel estimation is 10 dB. Before inserting additional pilots to address Doppler aliasing, the original length and period of historical channel estimations are set to $J=8$ and $T_{\mathrm{e}}=40$ ms. The number of additionally inserted pilots within T_{e} is $N=3$. Thus, the final length of estimated channel sequence is $\widetilde{J}=J+(J-1)N=29$, where the corresponding estimation time slots of non-uniform and uniform pilot patterns are $\mathcal{T}_{\mathrm{che}}=\{-280\,\mathrm{ms},-278\,\mathrm{ms},-260\,\mathrm{ms},-242\,\mathrm{ms},-240\,\mathrm{ms},\cdots,0\,\mathrm{ms}\}$ and $\mathcal{T}_{\mathrm{uni}}=\{-280\,\mathrm{ms},-270\,\mathrm{ms},-260\,\mathrm{ms},-250\,\mathrm{ms},-240\,\mathrm{ms},\cdots,0\,\mathrm{ms}\}$, respectively. The prediction length P is set to 8, and the prediction period in the stage of model test is $T_{\mathrm{p}}=5$ ms.

The structures of the proposed continuous-time transformer are presented in Table V. Here, d_{i} and d_{o} represent the dimensions of input and output features, respectively. The well-known multi-head attention is adopted to concurrently focus on distinct input aspects [60], where the number of heads is denoted as N_{head}. We utilize the layer normalization to accelerate convergence, and exploit the dropout strategy to mitigate the risk of overfitting. As outlined in Subsection IV-A, the continuous-time modeling for queries and keys adopts linear interpolation, while that for values is conducted via neural ODE with Runge-Kutta method. To construct the derivative in neural ODE, MLP based on hyperbolic tangent

TABLE VI
NMSEs OF DIFFERENT ATTENTION MECHANISMS, ω_0, AND ENCODING CONFIGURATIONS.

Methods	NMSE (dB)
Standard attention [12], $\omega_0 = 30$	-5.45
Continuous-time attention, $\omega_0 = 1$	-5.52
Continuous-time attention, $\omega_0 = 10$	-6.60
Continuous-time attention, $\boldsymbol{\omega_0 = 30}$	$\mathbf{-6.77}$
Continuous-time attention, $\omega_0 = 50$	-6.71
Continuous-time attention w/o any encoding	-4.76
Continuous-time attention w/ positional encoding	-5.32

TABLE VII
NMSE AND COMPUTATIONAL COMPLEXITY OF DIFFERENT VARIANTS OF CONTINUOUS-TIME TRANSFORMER.

Architectural variants	NMSE (dB)	Computational complexity (MFLOPs)
All-interpolation	-6.12	103.28
Hybrid (Original)	$\mathbf{-6.77}$	**395.59**
All-ODE	-6.80	986.72
Sparse attention [18]	-6.57	179.82
Fixed step size $\Delta t = 0.5$	-6.59	307.50
Fixed step size $\Delta t = 1$	-6.41	269.88
Doppler-adaptive step size Δt	-6.69	279.27
Encoder-only [14]	-4.85	1866.82
Continuous-time encoder	-6.78	1486.50

(tanh) activation is employed, wherein $d_{\mathrm{i,t}}$ and $d_{\mathrm{i,f}}$ denote the input dimensions for embedding time slots and features, respectively. Trapezoidal integration with the number of integration points $E = 2$ is used to approximate the calculation of continuous-time attention in (22). The default frequency hyperparameter ω_0 in (26) is set to 30. The training data set and the test data set contain $12,800$ and $2,560$ samples, respectively. The batch size is set to $B = 128$, and the entire training process contains 300 epochs using the Adam optimizer with the learning rate $\mathrm{lr} = 5 \times 10^{-4}$. The source codes of our simulations have been released at [35].

B. Simulation Results

1) Validation of Continuous-Time Attention and High-Frequency Temporal Encoding: Table VI presents the NMSE performance of continuous-time channel prediction under different attention mechanisms, frequency hyperparameters ω_0, and encoding configurations, where the bold value indicates the best performance with its corresponding setting. It can be seen that the continuous-time attention significantly outperforms the standard attention, implying that our continuous-time attention can accurately model the correlations across arbitrary time scales. When ω_0 is set to 1, the NMSE performance of continuous-time prediction degrades significantly, because the corresponding temporal encoding has difficulty in tracking the rapidly time-varying channels in high-mobility scenarios. In contrast, the high-frequency temporal encoding with $\omega_0 = 10/30/50$ can effectively address this issue, where the setting of $\omega_0 = 30$ achieves the best NMSE performance. In addition, we can see that the performance with high-frequency temporal encoding is superior to that without any encoding or with positional encoding, confirming its strong capability in capturing rapidly periodic channel dynamics. Thus, the continuous-time attention and the high-frequency temporal encoding with $\omega_0 = 30$ are appropriate for our continuous-time transformer.

2) Comparison of Different Architectural Variants for Continuous-Time Transformer: Table VII compares different architectural variants of the proposed continuous-time transformer in terms of NMSE and floating-point operations (FLOPs), with the original architecture denoted in boldface for clarity. Specifically, we first compare interpolation and neural ODE for the continuous-time modeling of queries, keys, and values. It can be seen that the original hybrid version (i.e., neural ODE for values, interpolation for queries and keys) significantly outperforms the all-interpolation variant with respect to NMSE, which validates the necessity of neural ODE to adaptively model the continuous-time dynamics of values. The all-ODE variant offers marginal prediction performance gains but with significantly higher computational complexity compared to our original hybrid version. Consequently, considering its accurate prediction capability with moderate computational cost, we adopt the hybrid version in the proposed continuous-time transformer.

Next, we evaluate the sparse attention variant of our continuous-time transformer. Concretely, sparse attention employs a probabilistic sparsity scoring mechanism to select critical queries and construct the sparse attention matrix, which can reduce the computational complexity of continuous-time attention from $\mathcal{O}(L^2 dE)$ to $\mathcal{O}\big(L \log(L) dE\big)$ [18]. It can be observed that sparse attention leads to a slight performance drop compared to the original architecture, but significantly reduces the computational complexity.

Furthermore, the performance impact of the step size Δt in ODE solver is investigated. "Fixed step size $\Delta t = 0.5$" and "Fixed step size $\Delta t = 1$" correspond to the fixed step size of $\Delta t = 0.5$ and $\Delta t = 1$, respectively, while "Doppler-adaptive step size Δt" refers to the assignment of fixed step sizes Δt selected from the set $\{0.5, 0.75, 1\}$ based on Doppler spectrum characteristics. Specifically, smaller Δt is used for higher Doppler shifts to improve prediction accuracy, while larger Δt is applied to the scenarios with lower Doppler shifts to enhance computational efficiency. In contrast, our original architecture adopts the standard ODE solver [49] incorporating an error control mechanism, which compares the difference between high-order and low-order solutions, and enables the adaptive adjustment of Δt. We can see that "Doppler-adaptive step size Δt" achieves better prediction performance compared to "Fixed step size $\Delta t = 0.5$" and "Fixed step size $\Delta t = 1$", because it selects an effective Δt for ODE solving based on Doppler spectrum characteristics. The original architecture exhibits performance advantages compared to these fixed-step variants, indicating that ODE solving based on error-control adaptive Δt is more accurate. However, this error-control adaptive Δt causes higher computational complexity. Linking neural ODE with Doppler spectrum characteristics for accurate and efficient ODE solving is worthy of investigation in future works.

In addition, different network architectures are studied at the bottom of Table VII. "Encoder-only" indicates that we replace the decoder with a two-layer FFN [14], and substitute the

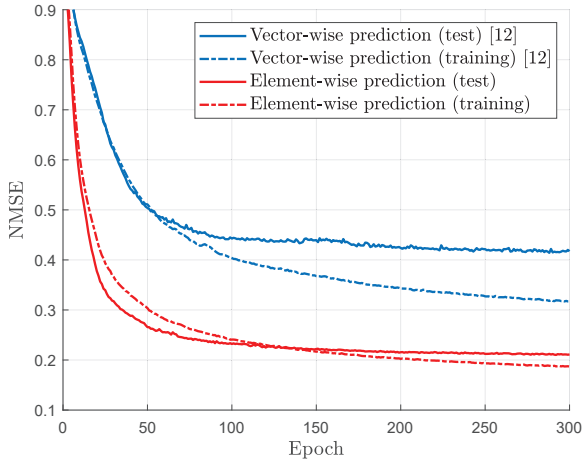


Fig. 5. Convergence performance comparison of continuous-time transformer with vector- and element-wise prediction mechanisms.

TABLE VIII
NMSE OF VECTOR- AND ELEMENT-WISE PREDICTION MECHANISMS UNDER DIFFERENT DEEP LEARNING MODELS.

Models	Vector-wise NMSE (dB)	Element-wise NMSE (dB)
GRU [11]	-0.50	-1.12
Vanilla transformer [12]	-0.77	-1.35
Neural ODE [30]	-0.74	-2.69
Latent ODE [50]	-1.07	-3.49
Continuous-time transformer	-3.83	-6.77

standard attention in the encoder with the continuous-time attention. "Continuous-time encoder" represents that the encoder adopts the continuous-time attention, while the remaining structure is identical to the original architecture. Firstly, we can see that "Encoder-only" suffers from a significant performance degradation compared to the original architecture. This is because the encoder of "Encoder-only" needs to concurrently model the temporal dependencies at fixed estimation time slots and arbitrary prediction time slots. Since the latter are randomly sampled during model training and differ across training samples, such variability can impede the single encoder of "Encoder-only" from adequately learning temporal correlations. In contrast, our original architecture explicitly separates the extraction of temporal correlations at fixed estimation time slots and arbitrary prediction time slots into the encoder and decoder, respectively, thereby ensuring effective learning of temporal dependencies. Secondly, "Continuous-time encoder" achieves nearly the same NMSE as the original architecture, yet with substantially increased computational complexity. This implies that the multi-scale encoder stack with the standard attention in our original architecture is already sufficient for the accurate extraction of multi-scale temporal features at fixed estimation time slots with relatively low computational cost.

3) Comparison of Vector- and Element-Wise Prediction Mechanisms: Fig. 5 investigates the convergence performance of the proposed continuous-time transformer with vector- and element-wise prediction mechanisms. It can be seen that the performance of element-wise prediction mechanism is superior to that of vector-wise prediction mechanism, indicating that the element-wise prediction mechanism is more effective in capturing temporal correlations. Furthermore, we can observe a substantial gap between the test and training performance with the vector-wise prediction mechanism, implying the presence of severe overfitting. However, this performance gap is significantly reduced under the element-wise prediction mechanism, which means that the overfitting is alleviated.

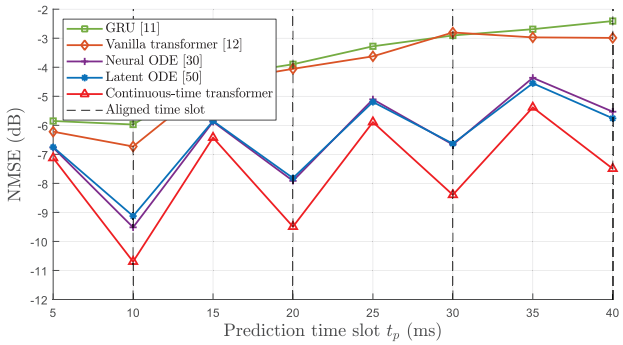
TABLE IX
PILOT NUMBER AND NMSE FOR DIFFERENT PILOT PATTERNS.

Pilot patterns	Pilot number	NMSE (dB)
Uniform [30]	29	-1.65
	36	-2.03
	43	-6.84
Random [61]	29	-1.99
	36	-2.17
	43	-2.26
Non-uniform	29	-6.86

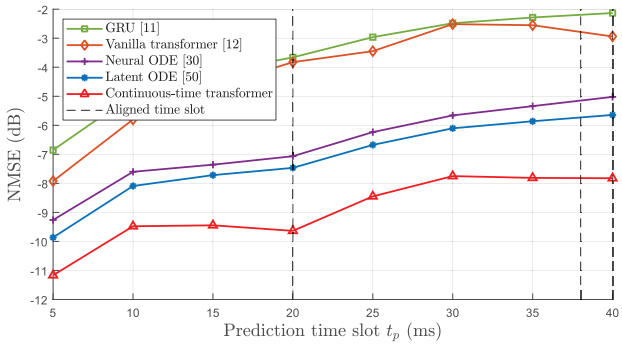
We further compare the NMSE performance of vector- and element-wise prediction mechanisms under various deep learning models in Table VIII. It can be seen that the superiority of element-wise prediction mechanism is consistently validated across diverse models. Thus all subsequent simulations are conducted under the element-wise prediction mechanism.

4) Investigation of Different Pilot Patterns in terms of Pilot Number and NMSE: In Table IX, we investigate the pilot number and NMSE performance for the uniform, random, and Chebyshev-deterministic non-uniform pilot patterns. The random pilot pattern [61] represents that the estimation time slots are randomly selected from the whole duration $[-280\,\text{ms}, 0\,\text{ms}]$, with different samples corresponding to distinct estimation time slots. For fairness, we uniformly sample the prediction time slots from $(0\,\text{ms}, 40\,\text{ms}]$. At first, it can be seen that compared to the non-uniform pilot pattern with 29 pilots, the uniform pilot patterns with 29 and 36 pilots yield significantly inferior performance due to the Doppler aliasing issue. When the pilot number increases from 36 to 43, the prediction performance of uniform pilot pattern realizes remarkable enhancement. This is because the estimation period $\frac{280\,\text{ms}}{43-1} = 6.67\,\text{ms}$ is almost sufficient to meet the theoretical minimum requirement for fully avoiding Doppler aliasing in Theorem 1. However, under comparable NMSE, the proposed non-uniform pilot pattern reduces pilot overhead by $\frac{43-29}{43} = 32.56\%$ compared to the uniform pilot pattern, indicating that the non-uniform pilot pattern achieves higher pilot efficiency. In addition, we can observe that the random pilot pattern exhibits poor prediction performance, which verifies that the channel prediction network relies on deterministic pilot patterns to efficiently extract temporal correlations. The adaptive pilot pattern learned jointly with the channel prediction network has the potential to achieve more accurate prediction than deterministic pilot patterns, as it can sense channel variations in a data-driven manner. The design of adaptive pilot pattern is worthy of investigation in future works.

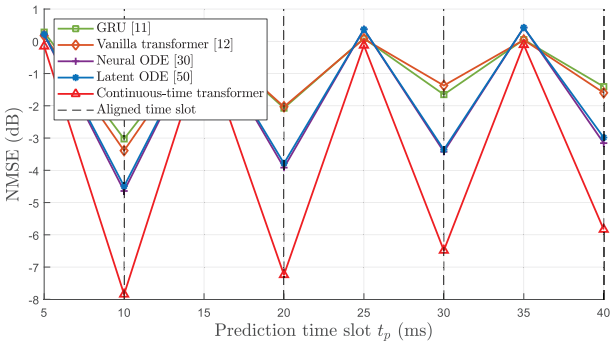
5) Investigation of Pilot Patterns and Deep Learning Models: Fig. 6 compares the NMSE performance of different pilot patterns and deep learning models at various prediction time slots t_p, given UE velocities $v = 20\,\text{km/h}$ and $80\,\text{km/h}$, respectively. Using the uniform and the proposed non-uniform pilot patterns $\{\mathcal{T}_{\text{uni}}, \mathcal{T}_{\text{che}}\}$ in Subsection V-A, the continuous-time channel prediction remains unaffected by Doppler aliasing at



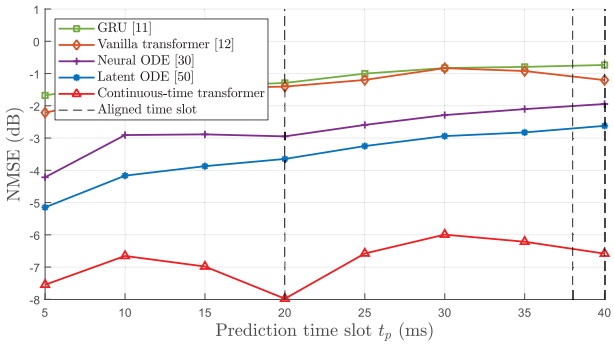
(a) Uniform pilot pattern [30] with $v = 20$ km/h, where Doppler aliasing is absent



(b) Proposed non-uniform pilot pattern with $v = 20$ km/h, where Doppler aliasing is absent



(c) Uniform pilot pattern [30] with $v = 80$ km/h, where Doppler aliasing occurs



(d) Proposed non-uniform pilot pattern with $v = 80$ km/h, where Doppler aliasing is absent

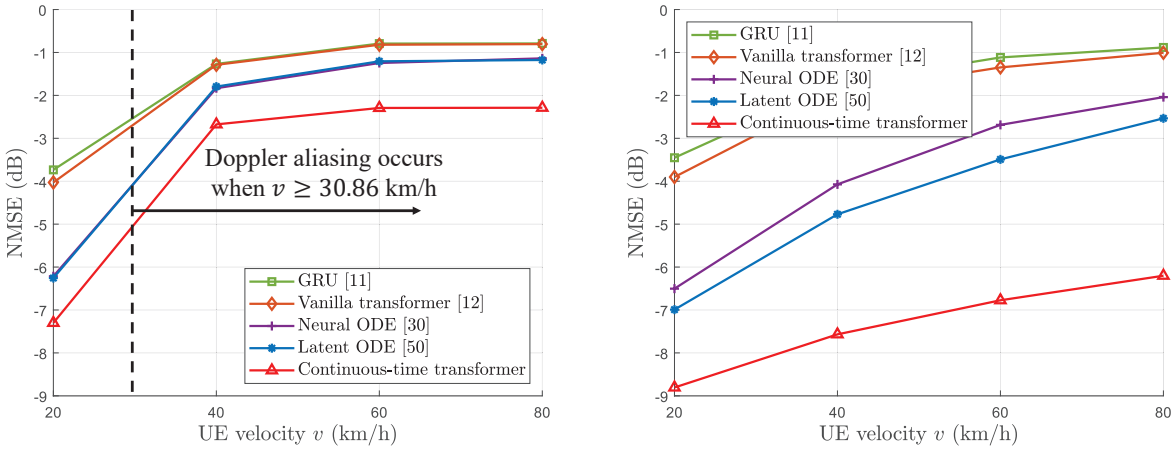
Fig. 6. NMSE performance under different pilot patterns and deep learning models, given UE velocities $v = 20$ km/h and 80 km/h.

$v = 20$ km/h in Figs. 6 (a) and 6 (b) based on Theorem 1. In contrast, at $v = 80$ km/h, Doppler aliasing arises in the uniform pilot pattern of Fig. 6 (c), but is absent in the proposed non-uniform pilot pattern of Fig. 6 (d).

It can be observed that the continuous-time prediction models, including neural ODE [30], latent ODE [50] and our proposed continuous-time transformer, significantly outperform GRU [11] and vanilla transformer [12], since the continuous-time prediction models have the capability to model the variation of time scales. Furthermore, for the uniform pilot pattern $\mathcal{T}_{\text{uni}}$ with $v = 20$ km/h in Fig. 6 (a), it can be seen that the NMSE performance of continuous-time prediction models at the prediction time slots $\{10\,\text{ms}, 20\,\text{ms}, 30\,\text{ms}, 40\,\text{ms}\}$ surpasses that at the prediction time slots $\{5\,\text{ms}, 15\,\text{ms}, 25\,\text{ms}, 35\,\text{ms}\}$, since the time slots $\{10\,\text{ms}, 20\,\text{ms}, 30\,\text{ms}, 40\,\text{ms}\}$ uniformly align with $\mathcal{T}_{\text{uni}}$. This performance gap between different prediction time slots is considerably exacerbated with $v = 80$ km/h in Fig. 6 (c) when Doppler aliasing occurs, indicating that Doppler aliasing can severely impair the accuracy of continuous-time channel prediction. Conversely, in Fig. 6 (d), the proposed non-uniform pilot pattern enables the continuous-time prediction models to maintain accuracy for all prediction time slots even at $v = 80$ km/h, validating that the proposed non-uniform pilot pattern can enhance the resolution of Doppler phase estimation and effectively overcome Doppler aliasing. We can also observe that our continuous-time transformer consistently achieves the best performance, demonstrating that the combination of neural ODE and attention mechanism can efficiently capture temporal dependencies across arbitrary time scales.

Furthermore, we focus on comparing the NMSE performance of different pilot patterns and deep learning models under varying UE velocities v in Fig. 7. First, it can be seen that the proposed continuous-time transformer consistently outperforms other deep learning models across various UE velocities v, confirming its robust performance gains. Next, when both uniform and non-uniform pilot patterns avoid Doppler aliasing at $v = 20$ km/h, the non-uniform pilot pattern achieves better NMSE performance, which verifies that using Chebyshev polynomial roots to configure pilots can facilitate the reconstruction of continuous-time channels. Moreover, a substantial performance degradation can be seen as the UE velocity v increases from 20 km/h to 40 km/h in Fig. 7 (a), since Doppler aliasing occurs when $v \geq 30.86$ km/h for the uniform pilot pattern according to Theorem 1. Conversely, we can observe that the prediction performance of the proposed continuous-time transformer degrades slowly with increasing v in Fig. 7 (b), because the proposed non-uniform pilot pattern prevents Doppler aliasing even when $v = 80$ km/h. Overall, across all settings of v, the proposed non-uniform pilot pattern achieves more accurate prediction than its uniform counterpart, thereby confirming its robust performance advantages.

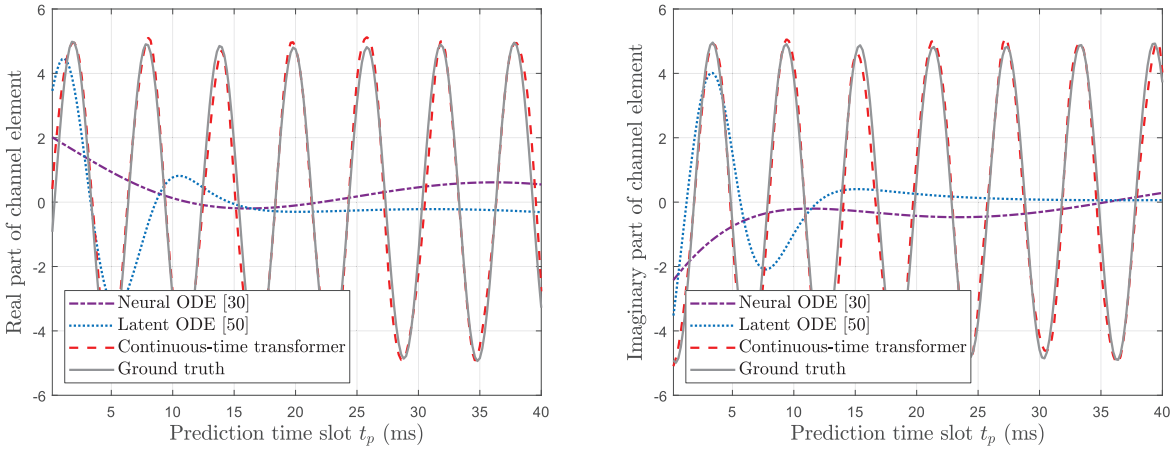
6) Explicit Illustration of Continuous-Time Prediction: Fig. 8 depicts the predicted continuous-time channels in the strongest angle index, generated by the continuous-time prediction models of neural ODE [30], latent ODE [50] and our continuous-time transformer under the proposed non-uniform pilot pattern. It can be seen that neural ODE and latent ODE fail to accurately fit the future continuous-time channels when the prediction time slot t_p becomes large, due to their



(a) Uniform pilot pattern [30]

(b) Proposed non-uniform pilot pattern

Fig. 7. NMSE performance with varying UE velocities v under different pilot patterns and deep learning models.



(a) Real part of continuous-time channels

(b) Imaginary part of continuous-time channels

Fig. 8. Explicit illustration of continuous-time prediction under different continuous-time prediction models with proposed non-uniform pilot pattern.

TABLE X
MODEL PARAMETERS AND COMPUTATIONAL COMPLEXITY OF DIFFERENT DEEP LEARNING MODELS.

Models	Model parameters (M)	Computational complexity (MFLOPs)
GRU [11]	50.24	58.84
Vanilla transformer [12]	173.51	100.02
Neural ODE [30]	66.88	64.09
Latent ODE [50]	83.52	122.81
ODE-Former [32]	169.84	305.66
Continuous-time transformer	176.13	395.59

reliance on RNN-like architectures that suffer from long-term dependencies and error accumulation issues. In contrast, the proposed continuous-time transformer demonstrates a precise tracking of future continuous-time channels across all t_p. It implies that the continuous-time attention mechanism can accurately extract the temporal dependencies across arbitrary time scales, and the parallel prediction framework can effectively solve error accumulation issues.

7) Comparison of Model Parameters and Computational Complexity: Table X compares the model parameters and computational complexity of different deep learning models. It can be seen that the proposed continuous-time transformer has a comparable number of model parameters to vanilla transformer. In contrast, due to the integration calculation of continuous-time attention, the computational complexity of continuous-time transformer is higher than that of other deep learning models. However, as demonstrated in previous simulations, our continuous-time transformer achieves much more accurate prediction than other deep learning models. In addition, when computational resources are limited, the architectural variants of continuous-time transformer such as the sparse attention and the all-interpolation versions in Table VII can be adopted. Taking the all-interpolation version as an example, its computational complexity is 103.28 MFLOPs, which is close to that of vanilla transformer, yet it remains a marked NMSE performance advantage over other deep learning models.

8) Investigation in Ray-Tracing Channels: To provide a more comprehensive and realistic evaluation, we further compare the prediction performance of different deep learning models in ray-tracing channels. Specifically, as a dataset widely adopted for channel prediction [17], [62], [63], DeepMIMO [64] based on Wireless InSite is used to generate ray-tracing channels. The channel parameters of DeepMIMO dataset are largely consistent with those in Table IV of 3GPP TR 38.901 dataset, while its additional channel parameters are listed in Table XI.

TABLE XI
CHANNEL PARAMETERS OF DEEPMIMO DATASET.

Parameters	Values
Scenario	Outdoor 1
BS index	3
UE location	Row 501 $\sim$ 1400

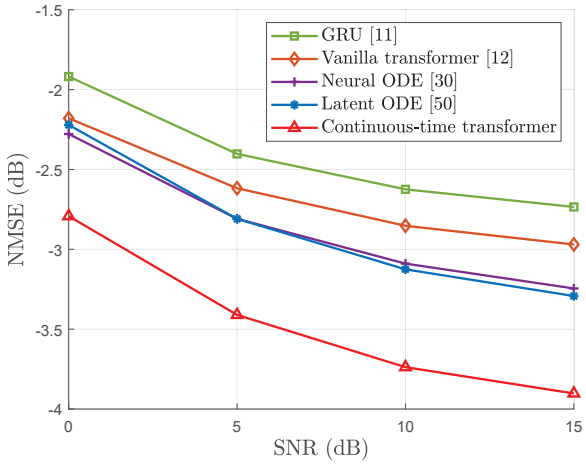


Fig. 9. NMSE performance under varying channel estimation SNRs for different deep learning models in DeepMIMO dataset, given proposed non-uniform pilot pattern and UE velocity $v = 60$ km/h.

Figure 9 depicts the NMSE performance of different deep learning models as the function of channel estimation SNR. It can be seen that the proposed continuous-time transformer markedly outperforms other deep learning models under all SNRs, which verifies its robust performance advantage to different SNR configurations and ray-tracing channel scenarios.

VI. CONCLUSIONS

We have studied continuous-time channel prediction in this paper, and our contribution has been threefold. First, we have analyzed the Doppler aliasing issue in continuous-time channel prediction from the perspective of Nyquist criterion. To handle this issue, a non-uniform pilot pattern based on Chebyshev polynomial roots has been proposed to provide a significantly finer estimation resolution of Doppler phase. Moreover, we have demonstrated that the proposed pilot pattern optimally facilitates the reconstruction of continuous-time channels in the sense of maximum norm. Second, we have proposed the continuous-time transformer, where the continuous-time attention mechanism has been derived to effectively capture the temporal correlations at arbitrary time scales, and a high-frequency temporal encoding has been designed to accurately track the rapidly time-varying channels in high-mobility scenarios. Third, an element-wise prediction mechanism has been proposed to focus on the efficient extraction of temporal dependencies for each element and mitigate overfitting. Simulation results have demonstrated that our proposed method can implement accurate continuous-time channel prediction in high-mobility scenarios and markedly outperforms existing channel prediction methods.

APPENDIX

A. Proof of Corollary 1

Proof: The minimum time intervals of $\mathcal{T}_{\text{che}}$ and $\mathcal{T}_{\text{uni}}$ are given by $T_{\text{che,min}} = \frac{T_{\text{e}}}{2} - \frac{T_{\text{e}}}{2}\cos\left(\frac{\pi}{2N}\right)$ and $T_{\text{uni,min}} = \frac{T_{\text{e}}}{N+1}$. Our objective is to prove

$$T_{\text{che,min}} - T_{\text{uni,min}} = \left(\frac{1}{2} - \frac{1}{2}\cos\left(\frac{\pi}{2N}\right) - \frac{1}{N+1}\right)T_{\text{e}} \leq 0, \ \forall N \in \mathbb{N}_+. \quad (29)$$

With $y = \frac{1}{N}$, the proof of (29) is equivalent to

$$\Delta_{\min}(y) = \frac{1}{2} - \frac{1}{2}\cos\left(\frac{\pi}{2}y\right) - \frac{y}{y+1} \leq 0, \ y \in (0, \ 1], \quad (30)$$

where we have $\Delta_{\min}(0) = 0$ and $\Delta_{\min}(1) = 0$.

The second derivatives of $\Delta_{\min}(y)$ can be calculated as

$$\Delta''_{\min}(y) = \frac{\pi^2}{8}\cos\left(\frac{\pi}{2}y\right) + \frac{2}{(y+1)^3}. \quad (31)$$

Due to $\cos\left(\frac{\pi}{2}y\right) \geq 0$ for $y \in (0, \ 1]$, $\Delta''_{\min}(y)$ satisfies

$$\Delta''_{\min}(y) > 0, \ y \in (0, \ 1]. \quad (32)$$

Therefore, $\Delta_{\min}(y)$ is a strictly convex function for $y \in (0, \ 1]$. Based on the strictly convex property and $\Delta_{\min}(0) = \Delta_{\min}(1) = 0$, (30) can be proved. ■

REFERENCES

[1] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, "Massive MIMO for next generation wireless systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 186–195, Feb. 2014.

[2] 5G PPP Architecture Working Group, "View on 5G architecture (Version 3.0)," 5G PPP Archit. Work. Group, White Paper, Jun. 2019.

[3] J. Yuan, H. Q. Ngo, and M. Matthaiou, "Machine learning-based channel prediction in massive MIMO with channel aging," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 2960–2973, May 2020.

[4] K. T. Truong and R. W. Heath, "Effects of channel aging in massive MIMO systems," *J. Commun. Netw.*, vol. 15, no. 4, pp. 338–351, Aug. 2013.

[5] X. Yu, D. Li, Z. Wang, and S. Sun, "An integrated new deep learning framework for reliable CSI acquisition in connected and autonomous vehicles," *IEEE Netw.*, vol. 37, no. 4, pp. 216–222, Jul./Aug. 2023.

[6] H. Kim, S. Kim, H. Lee, C. Jang, Y. Choi, and J. Choi, "Massive MIMO channel prediction: Kalman filtering vs. machine learning," *IEEE Trans. Commun.*, vol. 69, no. 1, pp. 518–528, Jan. 2021.

[7] C. Wu, *et al.*, "Channel prediction in high-mobility massive MIMO: From spatio-temporal autoregression to deep learning," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7, pp. 1915–1930, Jul. 2021.

[8] J. Vanderpypen and L. Schumacher, "MIMO channel prediction using ESPRIT based techniques," in *Proc. PIMRC* (Athens, Greece), Sep. 2007, pp. 1–5.

[9] H. Yin, H. Wang, Y. Liu, and D. Gesbert, "Addressing the curse of mobility in massive MIMO with Prony-based angular-delay domain channel predictions," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 12, pp. 2903–2917, Dec. 2020.

[10] W. Jiang and H. D. Schotten, "Neural network-based fading channel prediction: A comprehensive overview," *IEEE Access*, vol. 7, pp. 118112–118124, Sep. 2019.

[11] W. Jiang and H. D. Schotten, "Deep learning for fading channel prediction," *IEEE Open J. Commun. Soc.*, vol. 1, pp. 320–332, 2020.

[12] H. Jiang, M. Cui, D. W. K. Ng, and L. Dai, "Accurate channel prediction based on transformer: Making mobility negligible," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 9, pp. 2717–2732, Sep. 2022.

[13] S. Zhang, S. Zhang, Y. Mao, L. K. Yeung, B. Clerckx, and T. Q. S. Quek, "Transformer-based channel prediction for rate-splitting multiple access-enabled vehicle-to-everything communication," *IEEE Trans. Wireless Commun.*, vol. 23, no. 10, pp. 12717–12730, Oct. 2024.

[14] Y. Jin, Y. Wu, Y. Gao, S. Zhang, S. Xu, and C. -X. Wang, "LinFormer: A linear-based lightweight transformer architecture for time-aware MIMO channel prediction," *IEEE Trans. Wireless Commun.*, vol. 24, no. 9, pp. 7177–7190, Sep. 2025.

[15] O. Stenhammar, G. Fodor, and C. Fischione, "A comparison of neural networks for wireless channel prediction," *IEEE Wireless Commun.*, vol. 31, no. 3, pp. 235–241, Jun. 2024.

[16] Z. Xiao, Z. Zhang, C. Huang, X. Chen, C. Zhong, and M. Debbah, "C-GRBFnet: A physics-inspired generative deep neural network for channel representation and prediction," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 8, pp. 2282–2299, Aug. 2022.

[17] Z. Chen, Z. Zhang, Z. Yang, C. Huang, and M. Debbah, "Channel deduction: A new learning framework to acquire channel from outdated samples and coarse estimate," *IEEE J. Sel. Areas Commun.*, vol. 43, no. 3, pp. 944–958, Mar. 2025.

[18] H. Zhou, *et al.*, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proc. AAAI* (Vancouver, BC, Canada), Feb. 2021, pp. 11106–11115.

[19] A. Zeng, *et al.*, "Are transformers effective for time series forecasting?" in *Proc. AAAI* (Washington, DC, USA), Feb. 2023, pp. 11121–11128.

[20] Z. Chen, F. Gu, and R. Jiang, "Channel estimation method based on transformer in high dynamic environment," in *Proc. WCSP* (Nanjing, China), Oct. 2020, pp. 817–822.

[21] S. Singh, A. Trivedi, and D. Saxena, "Channel estimation for intelligent reflecting surface aided communication via graph transformer," *IEEE Trans. Green Commun. Netw.*, vol. 8, no. 2, pp. 756–766, Jun. 2024.

[22] H. Ju, S. Jeong, B. Lee, and B. Shim, "Transformer-based predictive channel estimation for mmWave massive MIMO systems," in *Proc. VTC-Fall* (Washington, DC, USA), Oct. 2024, pp. 1–5.

[23] J. Guo, G. Liu, Q. Wu, and P. Fan, "Parallel attention-based transformer for channel estimation in RIS-aided 6G wireless communications," *IEEE Trans. Veh. Technol.*, vol. 73, no. 11, pp. 15927–15940, Nov. 2024.

[24] Y. Cui, *et al.*, "Sensing-assisted high reliable communication: A transformer-based beamforming approach," *IEEE J. Select. Topics Signal Process.*, vol. 18, no. 5, pp. 782–795, May 2024.

[25] Y. Zhang, S. Li, D. Li, J. Zhu, and Q. Guan, "Transformer-based predictive beamforming for integrated sensing and communication in vehicular networks," *IEEE Internet Things J.*, vol. 11, no. 11, pp. 20690–20705, Jun. 2024.

[26] S. Zhang, S. Zhang, W. Yuan, and T. Q. S. Quek, "Transformer-empowered predictive beamforming for rate-splitting multiple access in non-terrestrial networks," *IEEE Trans. Wireless Commun.*, vol. 23, no. 12, pp. 19776–19788, Dec. 2024.

[27] J. Zhang, *et al.*, "Deep learning-empowered secure predictive beamforming design for integrated sensing and communications systems," *IEEE Trans. Wireless Commun.*, vol. 24, no. 10, pp. 8565–8580, Oct. 2025.

[28] Z. Xiao, Z. Zhang, Z. Chen, Z. Yang, and R. Jin, "Mobile MIMO channel prediction with ODE-RNN: A physics-inspired adaptive approach," in *Proc. PIMRC* (Kyoto, Japan), Sep. 2022, pp. 1301–1307.

[29] K. Ma, F. Zhang, W. Tian, and Z. Wang, "Continuous-time mmWave beam prediction with ODE-LSTM learning architecture," *IEEE Wireless Commun. Lett.*, vol. 12, no. 1, pp. 187–191, Jan. 2023.

[30] M. Cui, H. Jiang, Y. Chen, Y. Du, and L. Dai, "Continuous-time channel prediction based on tensor neural ordinary differential equation," *China Commun.*, vol. 21, no. 1, pp. 163–174, Jan. 2024.

[31] Z. Xiao, Z. Zhang, Z. Chen, Z. Yang, C. Huang, and X. Chen, "From data-driven learning to physics-inspired inferring: A novel mobile MIMO channel prediction scheme based on neural ODE," *IEEE Trans. Wireless Commun.*, vol. 23, no. 7, pp. 7186–7199, Jul. 2024.

[32] Z. Xiao, Y. Huang, Y. Xu, T. Jiao, and D. He, "ODE-Former for mobile channel prediction: A novel learning structure leveraging the physics continuity," *IEEE Wireless Commun. Lett.*, vol. 14, no. 7, pp. 2184–2188, Jul. 2025.

[33] Y. Sang, K. Ma, Z. Wang, and S. Chen, "Dual-band super-resolution channel prediction in high-mobility MIMO systems," *IEEE Trans. Commun.*, vol. 73, no. 6, pp. 4409–4424, Jun. 2025.

[34] Y. Nie, *et al.*, "A time series is worth 64 words: Long-term forecasting with transformers," in *Proc. ICLR* (Kigali, Rwanda), May 2023.

[35] *Continuous-time transformer for channel prediction*, Implementation Available in the Public Domain. Accessed: 2025. [Online]. Available: https://github.com/sangyl23/Continuous-Time-Transformer

[36] *Study on channel model for frequencies from 0.5 to 100 GHz (Release 17)*, document 3GPP, TR 38.901, 2022, version 17.0.0.

[37] G. Liu, Z. Hu, L. Wang, H. Zhang, J. Xue, and M. Matthaiou, "A hypernetwork based framework for non-stationary channel prediction," *IEEE Trans. Veh. Technol.*, vol. 73, no. 6, pp. 8338–8351, Jun. 2024.

[38] L. Wang, G. Liu, J. Xue, and K. -K. Wong, "Channel prediction using ordinary differential equations for MIMO systems," *IEEE Trans. Veh. Technol.*, vol. 72, no. 2, pp. 2111–2119, Feb. 2023.

[39] V. Lesnikov, T. Naumovich, and A. Chastikov, "Dealiasing technique for processing of sub-Nyquist sampled bandpass analytic signals," in *Proc. SIBCON* (Tomsk, Russia), May 2021, pp. 1–6.

[40] *Radio resource control (RRC) protocol specification (Release 15)*, document 3GPP, TS 38.331, 2019, version 15.6.0.

[41] T. J. Rivlin, *Chebyshev Polynomials*. Mineola, NY, Courier Dover, 2020.

[42] P. J. Davis, *Interpolation and Approximation*. New York, NY, Blaisdell, 1963.

[43] E. W. Cheney and W. A. Light, *A Course in Approximation Theory*. Providence, RI, American Mathematical Society, 2009.

[44] A. M. Dizqah, B. Lenzo, A. Sorniotti, P. Gruber, S. Fallah, and J. De Smet, "A fast and parametric torque distribution strategy for four-wheel-drive energy-efficient electric vehicles," *IEEE Trans. Ind. Electron.*, vol. 63, no. 7, pp. 4367–4376, Jul. 2016.

[45] R. Liu, X. Zhu, L. Liu, and B. Wu, "Personalized and common acceleration distribution characteristic of human driver," in *Proc. ITSC* (Maui, HI, USA), Nov. 2018, pp. 1820–1825.

[46] T. S. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed. Upper Saddle River, NJ, Prentice Hall, 2002.

[47] X. Ma and Z. Gao, "Data-driven deep learning to design pilot and channel estimator for massive MIMO," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5677–5682, May 2020.

[48] M. B. Mashhadi and D. Gündüz, "Pruning the pilots: Deep learning-based pilot design and channel estimation for MIMO-OFDM systems," *IEEE Trans. Wireless Commun.*, vol. 20, no. 10, pp. 6315–6328, Oct. 2021.

[49] R. T. Q. Chen, *et al.*, "Neural ordinary differential equations," in *Proc. NeurIPS* (Montréal, QC, Canada), Dec. 2018, pp. 6572–6583.

[50] Y. Rubanova, R. T. Q. Chen, and D. K. Duvenaud, "Latent ODEs for irregularly-sampled time series," in *Proc. NeurIPS* (Vancouver, BC, Canada), Dec. 2019, pp. 5320–5330.

[51] J. Morrill, *et al.*, "Neural rough differential equations for long time series," in *Proc. ICML* (Vienna, Austria), Jul. 2021, pp. 7829–7838.

[52] Y. Chen, *et al.*, "ContiFormer: Continuous-time transformer for irregular time series modeling," in *Proc. NeurIPS* (New Orleans, LA, USA), Dec. 2023, pp. 47143–47175.

[53] W. H. Press, *et al.*, *Numerical Recipes*. New York, NY, Cambridge Univ. Press, 1986.

[54] Y. Xiong, *et al.*, "Nyströmformer: A Nyström-based algorithm for approximating self-attention," in *Proc. AAAI* (Vancouver, Canada), Feb. 2021, pp. 14138-14148.

[55] K. Choromanski, *et al.*, "Rethinking attention with performers," in *Proc. ICLR*, May 2021.

[56] M. Raghu, *et al.*, "On the expressive power of deep neural networks," in *Proc. ICML* (Sydney, Australia), Aug. 2017, pp. 2847–2854.

[57] N. Rahaman, *et al.*, "On the spectral bias of neural networks," in *Proc. ICML* (Long Beach, CA, USA), Jun. 2019, pp. 5301–5310.

[58] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing scenes as neural radiance fields for view synthesis," in *Proc. ECCV* (Glasgow, UK), Aug. 2020, pp. 405–421.

[59] M. Tancik, *et al.*, "Fourier features let networks learn high frequency functions in low dimensional domains," in *Proc. NeurIPS* (Vancouver, Canada), Dec. 2020, pp. 7537–7547.

[60] P. Wang, K. Ma, Y. Bai, C. Sun, and Z. Wang, "Deep learning assisted mmWave beam prediction with flexible network architecture," *IEEE Trans. Wireless Commun.*, vol. 24, no. 11, pp. 9435–9448, Nov. 2025.

[61] A. N. Uwaechia, and N. M. Mahyuddin, "A review on sparse channel estimation in OFDM system using compressed sensing," *IETE Tech. Rev.*, vol. 34, no. 5, pp. 514–531, 2017.

[62] H. Kang, Q. Hu, H. Chen, Q. Huang, Q. Zhang, and M. Cheng, "Cross-shaped separated spatial-temporal UNet transformer for accurate channel prediction," in *Proc. IEEE INFOCOM* (Vancouver, Canada), May 2024, pp. 2079–2088.

[63] S. Fan, H. Li, X. Liang, Z. Liu, X. Gu, and L. Zhang, "E2ENet: An end-to-end channel prediction neural network based on uplink pilot for FDD systems," *IEEE Wireless Commun. Lett.*, vol. 13, no. 5, pp. 1285–1289, May 2024.

[64] A. Alkhateeb, "DeepMIMO: A generic deep learning dataset for millimeter wave and massive MIMO applications," in *Proc. ITA* (San Diego, CA, USA), Feb. 2019, pp. 1–8.