IEEE Transactions on
Wireless Communications

Digital Twin of Channel: Diffusion Model for Sensing-Assisted Statistical Channel State Information Generation

Journal:	*IEEE Transactions on Wireless Communications*
Manuscript ID	Draft
Manuscript Type:	Original Transactions Paper
Date Submitted by the Author:	n/a
Complete List of Authors:	Gong, Xinrui; Purple Mountain Laboratories; National Mobile Communications Research Laboratory Liu, Xiaofeng; Purple Mountain Laboratories; National Mobile Communications Research Laboratory Lu, An-An; Purple Mountain Laboratories; National Mobile Communications Research Laboratory Gao, Xiqi; Purple Mountain Laboratories; National Mobile Communications Research Laboratory Xia, Xiang-Gen; University of Delaware Wang, Cheng-Xiang; Purple Mountain Laboratories; National Mobile Communications Research Laboratory You, Xiaohu; Purple Mountain Laboratories; National Mobile Communications Research Laboratory
Keyword:	

Digital Twin of Channel: Diffusion Model for Sensing-Assisted Statistical Channel State Information Generation

Xinrui Gong, *Member, IEEE,* Xiaofeng Liu, *Member, IEEE,* An-an Lu, *Member, IEEE,* Xiqi Gao, *Fellow, IEEE,* Xiang-Gen Xia, *Fellow, IEEE,* Cheng-Xiang Wang, *Fellow, IEEE* and Xiaohu You, *Fellow, IEEE*

Abstract—**With the advancement of communication technology and the improvement of localization accuracy, cellular networks are gradually evolving from communication to perception-integrated networks. Addressing the research challenges of sensing-assisted communication, we propose, for the first time, the concept of Digital Twin of Channel (DToC). Specifically, we regard user terminal (UT) positions as physical objects, and statistical channel state information (CSI) as virtual digital objects. Observing the change trend of UTs' statistical CSI caused by the changes of UT's physical position enables predictive analytics for subsequent communication tasks. Then, we establish the relationship between physical and virtual digital objects using a Diffusion Model (DM) to achieve the DToC. Indeed, the DM can generate the desired objects by gradually denoising from noisy data using neural networks. Furthermore, we propose a conditional DM utilizing UTs' positions, which completes the task of generating the corresponding statistical CSI under known user-specific position conditions, thus mapping UT positions to statistical CSI. Simulation results demonstrate that our DToC framework outperforms previous statistical CSI estimation methods. Without the need of pilots, our method can simultaneously generate statistical CSIs from a large number of UTs' positions, achieving satisfactory results.**

Index Terms—**Digital twin, integrated sensing and communication, deep generative model, diffusion model, statistical channel information generation.**

I. Introduction

A. Background and Motivation

The integration of sensing and communication (ISAC) not only enables high-quality communication services but also provides high-precision sensing capabilities, making it a key research area for future wireless systems [1]. Advancements in wireless technologies, such as beyond fifth generation (B5G) and sixth generation (6G) systems, are expected to support various sensing-based applications like indoor positioning, Wi-Fi sensing, and radar sensing [1]. Traditionally, sensing and communication (S&C) operate independently in different frequency bands. However, with base stations (BSs) utilizing massive Multiple Input Multiple Output (MIMO) and Orthogonal Frequency Division Multiplexing (OFDM) technologies, they can transmit high-resolution radio signals, enabling high-precision sensing [2]. Jointly designing S&C systems to share the same frequency band and hardware not only improves efficiency but also reduces costs, making ISAC a highly promising research field [3].

Device-based ISAC, an essential component of ISAC, empowers the BS to achieve wireless positioning functionality while enhancing user terminals' (UTs') services [4]. Handling radio signals that carry both transmission data and location-related information, seamlessly integrating sensing and communication becomes a natural choice [5]. The application of the latest technologies, such as millimeter wave and massive MIMO, significantly improves positioning accuracy, advancing from hundreds of meters in the 2G era to meter-level precision in the 5G era [6]. This progress pushes future cellular mobile networks towards perceptive networks.

In this context, it becomes crucial to explore how to leverage the location and sensing data, obtained through sensors or BSs [7]. A promising and novel approach is the concept of Digital Twin (DT), widely acknowledged as a modeling technique for physical processes through digital technology. By analyzing UTs' historical location and sensing data, researchers can observe *trends in the DT, identify issues, and optimize, thereby providing predictive analytics for accurate subsequent decision-making*. Typically, a DT comprises three components: *a physical object, a virtual twin object, and their connections.* This concept has garnered significant attention from both industry and academia [8].

DT will play a crucial role in the operation and infrastructure management of 6G ISAC networks, providing significant assistance in shaping the future of communication networks. Supported by DT technology, communication networks will become more intelligent, flexible and will not only serve as infrastructure but also as a platform, fostering the development of other application areas such as Industrial Internet of Things (IoT), smart cities, Extended Reality (XR), and the Metaverse [9]. By integrating DT with 6G networks, a seamless connection between the physical and digital worlds can be fully established. This paper attempts to propose a kind of DT model tailored for the future 6G system, where the DT's *physical objects* are UTs' locations and other sensing data.

In the 6G systems, acquiring statistical Channel State

X. Gong, X. Liu, An-an Lu, X. Q. Gao, C.-X. Wang and X. You are with Purple Mountain Laboratories, Nanjing 211111, China and also with the National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China, (e-mail: {xinruigong, xf_liu, aalu, xqgao, chxwang, xhyu}@seu.edu.cn).

Xiang-Gen Xia is with the Department of Electrical and Computer Engineering, University of Delaware, Newark, DE 19716, USA (e-mail: xxia@ee.udel.edu).

Information (CSI) for UTs is crucial for the subsequent performance of communication. Statistical CSI enables the establishment of a *posterior* channel model, characterizing defects arising from channel aging and other factors, and facilitating the design of robust linear precoders [10]. Moreover, excellent performance in instantaneous CSI estimation has been achieved through method such as Minimum Mean Square Error (MMSE) channel estimation. However, prior to conducting instantaneous channel estimation, knowledge of the statistical CSI or statistical parameters is necessary.

Viewing the construction of the above communication systems from the perspective of DT, when BS detects the changes in UT's sensing data (such as location information), BS can map this *physical object to the DT's virtual twin object*. By observing the trend of the virtual twin object, BS can make advanced optimization decisions for subsequent communication services (such as instantaneous CSI estimation and precoding). Therefore, the *virtual twin object* in DT should correspond to certain channel characteristic parameters that reflect the physical location and serve the subsequent communication processes. It is natural to choose statistical CSI as the *virtual twin object* of the DT. In this paper, our objective is to explore the relationship between the physical object and the virtual twin object, establishing a mapping from the UT's location to the statistical CSI. We refer to this concept as Digital Twin of Channel (DToC).

The integration of artificial intelligence (AI) is expected to be a fundamental aspect of 6G systems. To establish DToC, leveraging AI to address the aforementioned challenges is a feasible approach. By combining data-driven and model-driven advantages, AI helps design an efficient and robust 6G system. Additionally, the powerful sensing and communication capabilities of 6G system can provide large training data for AI model learning. The aim is to combine the strengths of both AI and 6G system to model the complex relationships between the physical and virtual twin objects of DToC, including the propagation channels, surrounding scattering environment, and UT's locations. Therefore, utilizing AI to advance DToC research and explore the interaction between communication and sensing in AI-assisted systems is an interesting and promising research area.

Deep generative models have got significant attention within the AI field in recent years, such as variational autoencoders (VAEs) [11], generative adversarial networks (GANs) [12], and normalizing flows (NFs) [13]. By learning data-driven implicit priors, these models exhibit exceptional latent representational power, effectively capturing complex and rich structures found in natural signals. Diffusion models (DMs), a prominent subset of deep generative models, have gained popularity due to their remarkable performance in multi-modal learning applications [14]. With the emergence of DALL-E2 [15] and Stable Diffusion [16], DMs have revolutionized artificially intelligent-generated content (AIGC). Unlike other generative models, DMs do not require aligning posterior distributions (VAEs) [11], training extra discriminators (GANs) [12], or enforcing network constraints (NFs) [13]. The record-breaking performance of DMs in many areas have made them a compelling choice for AI applications.

The 'diffusion' model achieves the process of transforming structured signals into noise through gradual iterations. This process is a pre-processing step that generates noise matrices based on existing training data, commonly referred to as the 'forward process'. Then, the generated noise matrices are denoised by training a neural network (NN). Essentially, the NN learns how to reverse the diffusion process, effectively recovering the training data from the noise. This process is commonly referred to as the 'reverse process'. The novel approach of DM lies in their ability to learn from noisy samples and subsequently generate high-quality data (the desired output) by iteratively denoising pure noise matrices.

In this study, we propose an innovative sensing-assisted statistical CSI generation method to accomplish the DToC. Specifically, we delve into the construction of a conditional denoising diffusion model, including the derivation of the loss function, the implementation of the NN, and the iterative sampling technique. Leveraging this conditional DM, our trained NN combines UT location information to generate statistical CSI under specific UTs' locations. This establishes a connection between the physical object and the virtual digital object in DToC, achieving a mapping from UT location to statistical CSI. Our research adopts a data-driven approach aimed at enabling the BS to obtain location-conditioned statistical priors of the propagation channel.

B. Related Works

In the area of 6G ISAC, researchers have identified communication-assisted sensing as a pivotal research direction, aiming to explore high-precision BS positioning methods to achieve new 6G performance metrics. Among these methods, fingerprint localization is a feasible approach for achieving accurate positioning. For instance, in [2], the authors present a fingerprint-based single-cell localization method for massive MIMO-OFDM systems in rich scattering environments. The BS first extracts location marker features from the collected data, known as fingerprints. Then, by matching the fingerprint of the UT with a pre-recorded database, the UT can be accurately located. In [2], the location fingerprint corresponds to the Angle-Delay Channel Amplitude Matrix (ADCAM) of the channel. In [17], the authors extend single-cell fingerprint localization to a multi-cell cooperative scenario. Also deep learning methods are utilized to obtain location information faster and smarter. The authors use a refined beam domain channel energy coupling matrix as the location fingerprint, which better reflects the multipath characteristics of the channel. However, in the aforementioned literature, the location fingerprints, whether they are the Angle-Delay Channel Amplitude Matrix or the refined beam domain channel energy coupling matrix, correspond to the UT's statistical CSI. Moreover, in the localization problems discussed in these studies, it is assumed that the statistical CSI is known.

In the study [18], a direct localization method based on received pilot signals is proposed, which does not require the assumption of known statistical CSI. Through unsupervised pre-training of an NN with extensive received pilot signals, followed by fine-tuning the NN with a small set of signals labeled with location information, the NN can regress location coordinates based on received signals. All of the works can be considered as an exploration into communication-assisted sensing.

Given that the BS can obtain the UT's location directly from received signals, the next step lies in addressing the research gap of sensing-assisted communication. Exploration of DToC precisely fills this gap, as it can enable the BS to establish a mapping from location to statistical information, gaining prior knowledge about the surrounding scatterer environment. It can also be regarded as the *inverse process* of fingerprint localization. The DToC may enable the seamless integration of localization with communication functionalities, reaching in a comprehensive cellular system that uniformly combines S&C capabilities. Moreover, DToC may provide an essential *prior* information for subsequent instantaneous channel estimation and robust precoding, enhancing its potential impact.

C. Contribution and Organization

We summarize the contributions of this paper as follows:

1) For 6G massive MIMO ISAC systems, we propose, for the first time, the concept of DToC in sensing-assisted communication. We regard UTs' positions as the physical objects of DToC, and the statistical CSI of UTs as the virtual twin objects of the DToC. We attempt to explore a mapping from physical objects to virtual twin objects and then achieving DToC.
2) To achieve DToC, our work first introduces the Diffusion Model to address the statistical CSI generation challenge. Within this model, two distinct Markov chains—the forward and reverse processes—are deployed. The forward process applies controlled Gaussian noise gradually to training data through a predefined variance schedule. In contrast, the reverse process aims to remove the introduced noise step by step. By incorporating location coordinates as conditional supplementary inputs, we explore on the construction of conditional denoising diffusion models. This includes deriving the loss function, implementing neural networks, and iterative sampling techniques to generate statistical CSI at arbitrary positions within a target area.
3) Capitalizing on the robust parallel computing capabilities of the diffusion model, simultaneous generation of statistical CSI for multiple UTs becomes feasible once their location coordinates are provided. Our simulations demonstrate the simultaneous generation of statistical CSI for up to 100 UTs. Compared to traditional estimation methods and interpolation techniques, the proposed method achieves superior results without the need of pilot signals. These results demonstrate the excellent performance of DToC.

The rest of this paper is organized as follows. System model and channel model are introduced in Section II. Section III introduces the methodology design. In Section IV, we give the corresponding simulation results. Finally, in Section V, we conclude this paper. Some of the symbols used in this paper are listed below: Bold upper and lower cases denote matrix and column vectors, respectively. $\mathbb{C}^{M\times N}(\mathbb{R}^{M\times N})$ denotes the set of $M\times N$-dimensional complex (real) matrices, and $(\cdot)^H, (\cdot)^T, (\cdot)^*$ denote the conjugate transpose, transpose, and complex conjugate operations, respectively. $\mathbb{E}\{\cdot\}$ indicates the expected operation. $\odot$ denotes the Hadamard product, i.e., point-wise product, of two matrices. $\mathcal{CN}(\mathbf{a}, \mathbf{B})$ denotes the circular symmetric complex Gaussian distribution with mean $\mathbf{a}$ and covariance matrix $\mathbf{B}$. $[\cdot]_i$ and $[\cdot]_{ij}$ denote the i-th element of a vector and the (i, j)-th element of a matrix, respectively. $\mathrm{vec}(\cdot)$ transforms a matrix into a column vector by stacking its columns on top of each other. The symbol $|\cdot|$ represents the magnitude of a complex number.

II. System Model and Problem Formulation

In this section, we begin with introducing the channel model of the massive MIMO system. Subsequently, we will provide a overview of the digital twins of channel (DToC) and outline the challenges associated with the research problem.

A. System and Channel Model

In this paper, we focus on 6G cellular BS communication and sensing systems, along with its corresponding enabling technologies, to achieve the design of DToC and empower the communication technology and network design. This concept is represented in Fig. 1.

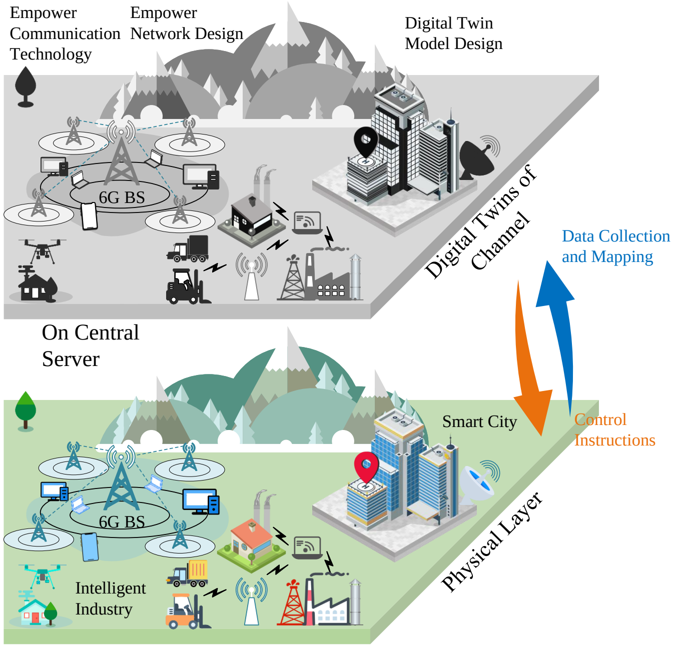

Fig. 1. Schematic diagram of 6G cellular BS scenario and the concept of the DToC.

Consider a 6G massive MIMO ISAC system with frequency selective fading channels. The system comprises a BS with uniform linear array (ULA) and K UTs, each equipped with a single antenna. The inter-antenna spacing amounts to half a wavelength. The BS has a total of N_r antennas. The steering vector in the spatial domain is

$$\mathbf{a}(\Theta) = \begin{bmatrix} 1 & e^{-j\pi\Theta} & \cdots & e^{-j\pi(N_r-1)\Theta} \end{bmatrix}^T. \quad (1)$$

Let $\cos(\theta) = \Theta$ denote the directional cosine, corresponding to the angle of arrival (AoA) θ. The integration of orthogonal frequency division multiplexing (OFDM) modulation is employed to convert the frequency-selective fading channel into multiple parallel channels. In the context of the system, the count of subcarriers is denoted as N_c, with N_p subcarriers reserved for uplink pilot signal transmission. We denote the cyclic prefix length and the sampling interval as N_g and T_s, respectively. Additionally, the subcarrier spacing is set as $\delta = \frac{1}{N_c T_s}$. This formulation facilitates the frequency domain steering vector, corresponding to the delay of arrival (DoA) τ, to be expressed as

$$\mathbf{b}(\tau) = \begin{bmatrix} 1 & e^{-j2\pi\delta\tau} & \cdots & e^{-j2\pi(N_p-1)\delta\tau} \end{bmatrix}^T. \tag{2}$$

Let's consider that the signal traverses through P_k Non-Line-of-Sight (NLoS) paths connecting the k-th UT and the BS. Notably, we denote $\Theta_{k,p}$, $\tau_{k,p}$, and $\beta_{k,t,p} \sim \mathcal{CN}(0, \gamma_{k,p}^2)$ as the directional cosine, DoA, and fading coefficient respectively, associated with the p-th path during the t-th OFDM symbol, connecting the k-th UT and the BS. Importantly, we make this assumption that the maximum delay spread $\tau_{k,max}$ remains within the duration of the cyclic prefix, i.e., $\tau_{k,max} \leq N_g T_s$. Under these premises, the spatial frequency domain channel matrix over the t-th symbol is given by [10], [19]–[21]

$$\mathbf{H}_{k,t} = \sum_{p=1}^{P_k} \beta_{k,t,p} e^{-j2\pi f_c \tau_{k,p}} \mathbf{a}(\Theta_{k,p}) \mathbf{b}^T(\tau_{k,p}). \tag{3}$$

Here, f_c signifies the carrier frequency. And we introduce a corresponding definition of the spatial frequency domain channel coefficients

$$h_{k,t}(\Theta, \tau) \triangleq \beta_{k,t,p} e^{-j2\pi f_c \tau_{k,p}} \delta(\Theta - \Theta_{k,p}) \delta(\tau - \tau_{k,p}). \tag{4}$$

Then, define sampled sets $\mathcal{B}_{a,i}$ and $\mathcal{B}_{d,j}$, respectively,

$$\mathcal{B}_{a,i} = \left\{ \Theta \mid \arg\min_{\Theta} |\Theta - \Theta_i| \right\}, \tag{5}$$

$$\mathcal{B}_{d,j} = \left\{ \tau \mid \arg\min_{\tau} |\tau - \tau_j| \right\}, \tag{6}$$

where the terms Θ_i and τ_j denote the sampled directional cosine and delay, respectively. We use N_a and N_d to represent the numbers of samples, and $\mathbf{a}(\Theta_i)$ and $\mathbf{b}(\tau_j)$ serve as the sampled steering vectors in the spatial and frequency domains, respectively. To ensure accurate quantization, we establish that N_a is greater than or equal to N_r, and N_d is greater than or equal to N_g. Additionally, we set Θ_i as $\frac{2i}{N_a} - 1$ for $i = 1, 2, \cdots, N_a$, which uniformly spans the interval $(-1, 1]$. Similarly, τ_j is uniformly distributed within the range $(0, N_g T_s]$ with $j = 1, 2, \cdots, N_d$.

When the values of N_a and N_d are sufficiently large, equation (3) can be represented using sampled steering vectors $\mathbf{a}(\Theta_i)$ and $\mathbf{b}(\tau_j)$, given as follows

$$\mathbf{H}_{k,t} = \sum_{i=1}^{N_a} \sum_{j=1}^{N_d} g_{k,t}(\Theta_i, \tau_j) \mathbf{a}(\Theta_i) \mathbf{b}^T(\tau_j), \tag{7}$$

where

$$g_{k,t}(\Theta_i, \tau_j) = \sum_{\Theta_{k,p} \in \mathcal{B}_{a,i}, \tau_{k,p} \in \mathcal{B}_{d,j}} h_{k,t}(\Theta, \tau). \tag{8}$$

Further, we define matrices $\mathbf{A}$ and $\mathbf{B}$ as

$$\mathbf{A} = [\mathbf{a}(\Theta_1), \mathbf{a}(\Theta_2), \cdots, \mathbf{a}(\Theta_{N_a})] \in \mathbb{C}^{N_r \times N_a}, \tag{9}$$

$$\mathbf{B} = [\mathbf{b}(\tau_1), \mathbf{b}(\tau_2), \cdots, \mathbf{b}(\tau_{N_d})] \in \mathbb{C}^{N_p \times N_d}, \tag{10}$$

then (7) can be expressed by matrix multiplication as

$$\mathbf{H}_{k,t} = \mathbf{A} \mathbf{G}_{k,t} \mathbf{B}^T. \tag{11}$$

Equation (11) can be referred to as a refined double beam-based channel model, with $\mathbf{G}_{k,t} \in \mathbb{C}^{N_a \times N_d}$ being the refined beam domain channel matrix. The (i,j)-th element of $\mathbf{G}_{k,t}$ is represented as $[\mathbf{G}_{k,t}]_{i,j} = g_{k,t}(\Theta_i, \tau_j)$, signifying the fading coefficients of the spatial and frequency beams. Each sampled space/frequency steering vector corresponds to a distinct physical beam in the space/frequency domain. When $N_a = N_r$ and $N_d = N_g$, $\mathbf{A}$ takes the form of a discrete Fourier transform (DFT) matrix, while $\mathbf{B}$ represents the submatrix consisting of the first N_g columns of a DFT matrix. In this configuration, equation (11) coincides with the DFT-based channel model [22].

Define $\Gamma_k(\Theta_{k,p}, \tau_{k,p}) = \mathbb{E}\{h_{k,t}(\Theta_{k,p}, \tau_{k,p}) \cdot h_{k,t}^*(\Theta_{k,p}, \tau_{k,p})\}$. Subsequently, introduce the beam domain channel power matrix $\mathbf{\Omega}_k \in \mathbb{R}^{N_a \times N_d}$, where the (i,j)-th element is expressed as

$$[\mathbf{\Omega}_k]_{i,j} \triangleq \sum_{\Theta_{k,p} \in \mathcal{B}_{a,i}, \tau_{k,p} \in \mathcal{B}_{d,j}} \Gamma_k(\Theta_{k,p}, \tau_{k,p}). \tag{12}$$

The elements of $\mathbf{G}_{k,t}$ satisfy

$$\mathbb{E}\left\{ [\mathbf{G}_{k,t}]_{i,j} \cdot [\mathbf{G}_{k,t}]_{i',j'}^* \right\} = [\mathbf{\Omega}_k]_{i,j} \delta(i - i') \delta(j - j'). \tag{13}$$

The equations (12) and (13) explicitly highlight the crucial role of the refined beam domain channel power matrix, denoted as $\mathbf{\Omega}_k$, in capturing the long-term characteristics of the refined beam domain channel matrix $\mathbf{G}_{k,t}$ during temporal variations. And $\mathbf{\Omega}_k$ reflects the channel information regarding the power, DoA and AoA of each path associated with the scatterers between the UT and the BS. This unique matrix is appropriately referred to as the 'beam domain statistical CSI' denoting its ability to capture the invariant characteristics of channel over time.

The strength of $\mathbf{\Omega}_k$ lies in its ability to distribute channel power across different distinguishable spatial directions and time delays. Within the matrix $\mathbf{\Omega}_k$, one can find information regarding the power, AoA, and DoA associated with each path connecting the k-th UT to the BS. And the equation (13) reveals that the elements within the refined beam domain channel matrix $\mathbf{G}_{k,t}$ are independent, while each component of $\mathbf{\Omega}_k$ is fundamentally linked with the propagation environment and the scatterers found between the UT and the BS. This interrelation is further confirmed by the principle of channel sparsity, where the majority of elements in $\mathbf{\Omega}_k$ tend to zero, forming clusters that correspond to distinct physical scatterers [23].

Compared to the conventional DFT-based channel model, the refined double beam domain channel power matrix $\mathbf{\Omega}_k$ provides enhanced resolution for both angle and delay of the paths connecting UTs and the BS. This enhanced precision of $\mathbf{\Omega}_k$ allows for a more accurate representation of the local propagation environment. Serving as a comprehensive representation, $\mathbf{\Omega}_k$ represents the information involving the BS, UT, and environmental scatterers. Consequently, the acquisition of UT's $\mathbf{\Omega}_k$ plays a pivotal role in 6G massive MIMO ISAC systems, serving as a cornerstone for comprehending surroundings of systems and offering communication services to UTs. And this unique characteristic lays the groundwork for a complex interaction between communication and sensing.

B. Problem Formulation

Based on the description of refined double beam-based channel matrix $\mathbf{G}_{k,t}$ and power matrix $\mathbf{\Omega}_k$, they represent the environmental characteristics of the main scatterers between the BS and the UTs. We observe a high correlation between the channel and the environment. As illustrated in Fig. 1, we can design the refined double beam-based channel power matrix $\mathbf{\Omega}_k$ as a *virtual twin object* of the physical transmission channel, and we take the UT's location as the *physical object* in DToC. In this paper, our main goal is to find the mapping between physical and the virtual twin based on the proposed method.

Building upon our previous work [18], this paper assumes the existence of a location-aware method within the system, facilitating the acquisition of UTs' location coordinates. To simplify the issue in this paper, the UTs' location information represents the system perception of the environment. Our focus here is to skip beyond the UT positioning step and directly generate the UT's statistical CSI, $\mathbf{\Omega}_k$, from the specific location coordinates. In essence, our goal is to develop a novel location-assisted statistical CSI generation method. The generation method based on other perceptual information is left for future research.

If we look at this problem from the perspective of probability and statistics, we assume that the statistical CSI $\mathbf{\Omega}$ at any location within the target area should satisfy the conditional distribution $q(\mathbf{\Omega}|\mathbf{p})$, where q represents the probability density function, $\mathbf{\Omega} \in \mathbb{R}^{N_a \times N_d}$, and $\mathbf{p} \in \mathbb{R}^2$ represents the location coordinate vector. We aim to find this conditional distribution $q(\mathbf{\Omega}|\mathbf{p})$ so that, given a UT's location coordinate vector $\mathbf{p}_k = [x_k, y_k]^T$, we can sample the statistical CSI $\mathbf{\Omega}_k^{UT}$ for the UT from the distribution. In our problem, we assume that the location coordinates $\mathbf{p}_i^{RP}$ at some 'reference points' (RPs) and their corresponding statistical CSI $\mathbf{\Omega}_i^{RP}$ are known. We hope to learn the underlying probability distribution of the data and then generate new data from this distribution, which aligns with the goals of generative AI. Once we obtain this probabilistic representation q, we establish the connection between the location coordinates and the statistical CSI, thereby constructing a DToC.

The problem discussed in this paper can be formulated as

$$\hat{\mathbf{\Omega}} \sim \hat{q}(\mathbf{\Omega}|\mathbf{p}), \tag{14}$$

where $\hat{q}$ represents the desired probability distribution of the statistical CSI $\mathbf{\Omega}$ obtained from the known RPs, and the symbol $\sim$ indicates obeying a certain distribution.

Through DToC, we no longer solely rely on obtaining channel information within the communication system but leverage location data to acquire partial channel information in advance. Consequently, this will provide new opportunities for the physical layer design of communication. We observe that the problem represents a mapping from low-dimensional to high-dimensional space $\mathbb{R}^2 \rightarrow \mathbb{R}^{N_a \times N_d}$. Traditional supervised AI methods to find this mapping is difficult to achieve the desired results. Therefore, we have adopted a novel generative AI approach, namely the diffusion model (DM).

III. Methodology

In this section, we start with introducing the overarching strategy of our system. Subsequently, we delve into the detailed of the DM approach. This includes an in-depth exploration of 'noise scheduling', the acquisition of the loss function, and the details of the network configuration.

A. General Strategy of Generating Statistical CSI

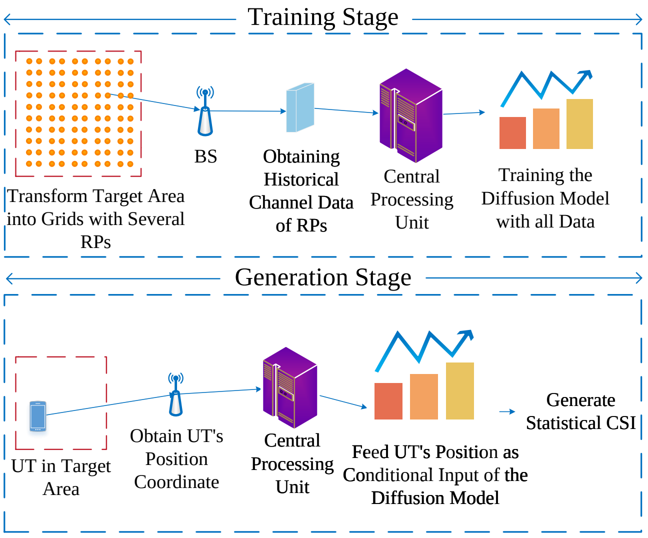
Training Stage
Transform Target Area into Grids with Several RPs
BS
Obtaining Historical Channel Data of RPs
Central Processing Unit
Training the Diffusion Model with all Data
Generation Stage
UT in Target Area
Obtain UT's Position Coordinate
Central Processing Unit
Feed UT's Position as Conditional Input of the Diffusion Model
Generate Statistical CSI

Fig. 2. General strategy of the DM-based statistical CSI generation methods.

Similar to numerous end-to-end AI-based communication methodologies [17], [24], [25], the proposed statistical CSI generation method based on DM is split into two principal phases: an offline training phase and an online generation phase, as shown in Fig. 2. In the training phase, a self-guided vehicle traverses the target area, which could include locales such as a university campus or a shopping mall. At specific intervals, this vehicle establishes RPs along its path, noting the spatial coordinates of these RPs' locations, referred to as $\mathbf{p}_i^{RP} \in \mathbb{R}^2$. It then transmits detection signals to the BS, which subsequently captures and archives the statistical CSI $\mathbf{\Omega}_i^{RP}$ associated with these RPs' locations using the method in the related works [26]. Eventually, the BS accumulates the data to form a training dataset $\mathcal{D} = \{\mathbf{p}_i^{RP}, \mathbf{\Omega}_i^{RP}\}_{i=1}^{N_{RP}}$, the resource utilized for training the Diffusion Model.

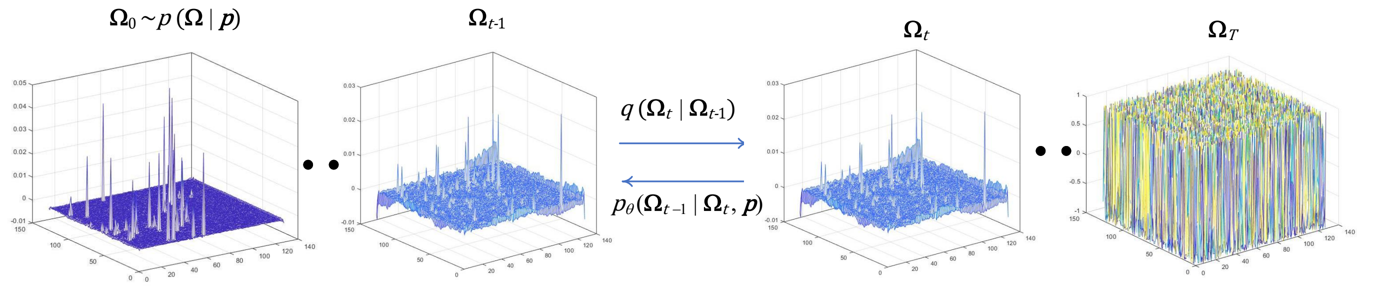

Fig. 3. The forward diffusion process q (left to right) gradually adds Gaussian noise to the target CSI $\mathbf{\Omega}_0$. The reverse inference process p_θ (right to left) iteratively denoises the noisy matrix conditioned on a position $\mathbf{p}$.

During the generation phase, when a UT shows communication requisites, we suppose that the BS can promptly determine its geographical coordinates (positioning issues are not the focus of our work). Subsequently, employing the proposed conditional DM, the UT's statistical CSI, denoted as $\hat{\mathbf{\Omega}}^{UT} \in \mathbb{R}^{N_a \times N_d}$, is generated conditioned upon the UT's positional coordinates $\mathbf{p}^{UT} \in \mathbb{R}^2$.

B. Overview of Diffusion Model

The denoising diffusion probabilistic model (DDPM), a successful practice of DM despite not considering conditional information, fundamentally aims to establish an easily samplable distribution, denoted as $p_\theta(\mathbf{x})$ where θ representing NN parameters, to effectively approximate a training target data distribution, represented by $q(\mathbf{x})$ for $\mathbf{x} \in \mathbb{R}^G$. DDPM includes both forward diffusion and reverse inference processes [14]. In essence, DDPM is designed to create an NN that allows us to efficiently generate samples from the easily-sampled distribution $p_\theta(\mathbf{x})$.

Furthermore, the distribution $q(\mathbf{\Omega})$ that characterizes the statistical CSI $\mathbf{\Omega}$ within the target area can be likened to $q(\mathbf{x})$ in DDPM, which is transitioning from vector to matrix problem. Consequently, the statistical CSI observed at the RP locations, denoted as $\mathbf{\Omega}_i^{RP}$ within training dataset, serves as a series of data points sampled from the distribution $q(\mathbf{\Omega})$. *The core objective of employing the DDPM for CSI generation is to uncover an easily-sampled distribution, denoted as $p_\theta(\mathbf{\Omega})$, enabling the BS to comprehend the surrounding scatterer environment and sampling the statistical CSI of UTs from $p_\theta(\mathbf{\Omega})$.* In the context of statistical CSI generation, we can relate forward diffusion and reverse process to the training stage, where the NN is learned, and the generation stage corresponding to using the trained NN to sample from the easily-sampled distribution $p_\theta(\mathbf{x})$ in Fig. 2.

However, compared with the DDPM [14], our scenario differs in that we have a training dataset $\mathcal{D}$ including pairs of RP's statistical CSI $\mathbf{\Omega}_i^{RP}$ and their corresponding position coordinates $\mathbf{p}_i^{RP}$. This dataset $\mathcal{D}$ actually comprises samples drawn from an unknown conditional distribution $q(\mathbf{\Omega}|\mathbf{p})$. So our primary objective further revolves around developing a parametric approximation distribution $p_\theta(\mathbf{\Omega}|\mathbf{p})$ for $q(\mathbf{\Omega}|\mathbf{p})$, utilizing a conditional DM to establish the relationship between position coordinates $\mathbf{p}_i^{RP}$ and the corresponding targets $\mathbf{\Omega}_i^{RP}$ in $\mathcal{D}$, achieving the DToC.

To address this challenge, we propose an adaptation of the DDPM introduced in [14] to suit the context of conditional generation. This modified model is tailored to learn the underlying conditional distribution $q(\mathbf{\Omega}|\mathbf{p})$ in (14), the task that aligns with our aim of generating statistical CSI based on provided position coordinates.

The conditional DM generates the target $\mathbf{\Omega}_i^{RP}$ through a series of T refinement steps. For introducing the DM conveniently, we omit the ordinal subscript i of the RP in subsequent text for clarity, while the subscript t of $\mathbf{\Omega}$ denote different time steps for consistency with [14]. The reverse inference process initiates with a purely noise matrix $\mathbf{\Omega}_T \sim \mathcal{N}(0, \mathbf{I})$ and progressively refines this matrix over subsequent iterations $(\mathbf{\Omega}_{T-1}, \mathbf{\Omega}_{T-2}, \ldots, \mathbf{\Omega}_0)$ using learnable conditional transition distributions $p_\theta(\mathbf{\Omega}_{t-1}|\mathbf{\Omega}_t, \mathbf{p})$, ultimately leading to $\mathbf{\Omega}_0 \sim p_\theta(\mathbf{\Omega}|\mathbf{p})$ (refer to Fig. 3 right to left).

In the reverse process, the state of the intermediate matrix is controlled by the forward diffusion process. The forward diffusion process is implemented by a predefined deterministic Markov chain, denoted as $q(\mathbf{\Omega}_t|\mathbf{\Omega}_{t-1})$, which physically means gradually introducing Gaussian noise into the training data (refer to Fig. 3 left to right). The key of conditional DM lies in reversing this Gaussian forward diffusion process, which is achieved through a reverse Markov chain (namely, reverse process) that's conditioned on the positional information $\mathbf{p}$. The reverse process utilizes a denoising deep NN $\boldsymbol{\epsilon}_\theta$ to implement the learnable conditional transition distribution $p_\theta(\mathbf{\Omega}_{t-1}|\mathbf{\Omega}_t, \mathbf{p})$. $\boldsymbol{\epsilon}_\theta$ takes the position $\mathbf{p}$ and intermediate noisy CSI matrix $\mathbf{\Omega}_t$ as inputs and effectively estimates the added noise component.

In the subsequent sections, we present a comprehensive overview of the forward diffusion process, followed by details of how denoising model $\boldsymbol{\epsilon}_\theta$ is trained and subsequently using $\boldsymbol{\epsilon}_\theta$ for the sample.

C. Gaussian Diffusion Process

Within the forward process, the objective involves the gradual transformation of all the RPs' statistical CSI training data in dataset $\mathcal{D}$ into Gaussian noise by introducing minor noise perturbations. In contrast, the reverse process needs the stepwise denoising of Gaussian noise to restore the original target by NN. To facilitate this, a sequence of latent variables $\mathbf{\Omega}_{1:T} = \{\mathbf{\Omega}_t\}_{t=1}^T$ is introduced, maintaining the same dimensionality as the original training $\mathbf{\Omega}_0 \in \mathbb{R}^{N_a \times N_d}$ (128×128 in simulation). Referred to as the diffusion process, the forward process is formally defined as a T Markov chain

$$q(\boldsymbol{\Omega}_{1:T}|\boldsymbol{\Omega}_0) = \prod_{t=1}^{T} q(\boldsymbol{\Omega}_t|\boldsymbol{\Omega}_{t-1}), \tag{15}$$

with the transition probability density function $q(\boldsymbol{\Omega}_t|\boldsymbol{\Omega}_{t-1})$ is a pre-designed Gaussian distribution aimed at transforming the original prior $q(\boldsymbol{\Omega}_0)$ into a simple distribution

$$q(\boldsymbol{\Omega}_t|\boldsymbol{\Omega}_{t-1}) = \mathcal{N}\left(\boldsymbol{\Omega}_t; \sqrt{1-\beta_t}\boldsymbol{\Omega}_{t-1}, \beta_t \mathbf{I}_G\right). \tag{16}$$

Here, $\beta_{1:T} = \{\beta_t\}_{t=1}^{T}$ represents a predefined variance schedule that regulates the added noise at each step, $0 < \beta_1 < \beta_2 < \cdots < \beta_T \ll 1$. $\mathbf{I}_G$ is the $G = 128$-dimensional identity matrix. The attribute of the forward process lies in it can generate $\boldsymbol{\Omega}_t$ from $\boldsymbol{\Omega}_0$ using a closed-form sampling

$$q(\boldsymbol{\Omega}_t|\boldsymbol{\Omega}_0) = \mathcal{N}\left(\boldsymbol{\Omega}_t; \sqrt{\alpha_t}\boldsymbol{\Omega}_0, (1-\alpha_t)\mathbf{I}_G\right). \tag{17}$$

Based on (17), we can use reparameterization method to sample $\boldsymbol{\Omega}_t$ from the original data $\boldsymbol{\Omega}_0$ in a closed-form manner through 'noise scheduling value' $\alpha_t = \prod_{s=1}^{t}(1-\beta_t)$.

$$\boldsymbol{\Omega}_t = \sqrt{\alpha_t}\boldsymbol{\Omega}_0 + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_t. \tag{18}$$

This process involves $\boldsymbol{\epsilon}_t \sim \mathcal{N}(0, \mathbf{I}_G)$. When T is sufficiently large, $\alpha_T \approx 0$ and $q(\boldsymbol{\Omega}_T|\boldsymbol{\Omega}_0) \approx \mathcal{N}(0, \mathbf{I}_G)$, effectively making $\boldsymbol{\Omega}_T$ resemble standard Gaussian noise matrix. By applying Bayes' rule

$$q(\boldsymbol{\Omega}_t|\boldsymbol{\Omega}_{t-1}) = q(\boldsymbol{\Omega}_{t-1}|\boldsymbol{\Omega}_t, \boldsymbol{\Omega}_0)\frac{q(\boldsymbol{\Omega}_t|\boldsymbol{\Omega}_0)}{q(\boldsymbol{\Omega}_{t-1}|\boldsymbol{\Omega}_0)}, \tag{19}$$

the forward process (15) can be reformulated as

$$q(\boldsymbol{\Omega}_{1:T}|\boldsymbol{\Omega}_0) = q(\boldsymbol{\Omega}_T|\boldsymbol{\Omega}_0)\prod_{t=2}^{T} q(\boldsymbol{\Omega}_{t-1}|\boldsymbol{\Omega}_t, \boldsymbol{\Omega}_0), \tag{20}$$

according to (16) and (17),

$$q(\boldsymbol{\Omega}_{t-1}|\boldsymbol{\Omega}_t, \boldsymbol{\Omega}_0) = \mathcal{N}\left(\boldsymbol{\Omega}_{t-1}; \widetilde{\boldsymbol{\mu}}_t(\boldsymbol{\Omega}_t, \boldsymbol{\Omega}_0), \widetilde{\beta}_t \mathbf{I}_G\right), \tag{21}$$

where

$$\widetilde{\boldsymbol{\mu}}_t(\boldsymbol{\Omega}_t, \boldsymbol{\Omega}_0) = \frac{\sqrt{\alpha_{t-1}}\beta_t}{1-\alpha_t}\boldsymbol{\Omega}_0 + \frac{\sqrt{1-\beta_t}(1-\alpha_{t-1})}{1-\alpha_t}\boldsymbol{\Omega}_t, \tag{22}$$

$$\widetilde{\beta}_t = \frac{\beta_t\,(1-\alpha_{t-1})}{1-\alpha_t}. \tag{23}$$

Furthermore, we establish $\alpha_0 = 1$ to ensure consistency with the situation when $t = 1$.

This (21) posterior distribution plays a crucial role in shaping the parameterization of the reverse Markov chain and constructing a variational lower bound on the log-likelihood of the reverse Markov chain. In the subsequent sections, we delve into the training an NN to effectively reverse this Gaussian diffusion process. We provide the derivation process about (21) in (24), where $C\left(\boldsymbol{\Omega}_t, \boldsymbol{\Omega}_0\right)$ is a constant term about $\boldsymbol{\Omega}_{t-1}$, calculated solely by a combination of $\boldsymbol{\Omega}_t, \boldsymbol{\Omega}_0$, which can be implicitly returned in the final equation to satisfy the perfect squared formula.

D. Loss Function of Conditional Denoising Model

Subsequently, the reverse process, is formulated as a reverse Markov chain beginning with Gaussian noise $\boldsymbol{\Omega}_T$, $p(\boldsymbol{\Omega}_T) \sim \mathcal{N}(\boldsymbol{\Omega}_T; 0, \mathbf{I}_G)$. *Noticing that in order to complete DToC, we utilize location coordinate vector* $\mathbf{p}$ *as the conditional information for the Markov chain during the design of the reverse process to learn an easily-sampled distribution* $p_\theta(\boldsymbol{\Omega}_{0:T}|\mathbf{p})$ *correlated with UT's location*:

$$p_\theta(\boldsymbol{\Omega}_{0:T}|\mathbf{p}) = p(\boldsymbol{\Omega}_T)\prod_{t=1}^{T} p_\theta(\boldsymbol{\Omega}_{t-1}|\boldsymbol{\Omega}_t, \mathbf{p}) \tag{25}$$

We formulate the reverse process using isotropic Gaussian conditional distribution, denoted as $p_\theta(\boldsymbol{\Omega}_{t-1}|\boldsymbol{\Omega}_t, \mathbf{p})$, which is learned from the training data. Designing a Gaussian transition probability density function that can be learned by NN

$$p_\theta(\boldsymbol{\Omega}_{t-1}|\boldsymbol{\Omega}_t, \mathbf{p}) = \mathcal{N}\left(\boldsymbol{\Omega}_{t-1}; \boldsymbol{\mu}_\theta(\boldsymbol{\Omega}_t, t, \mathbf{p}), \boldsymbol{\Sigma}_\theta(\boldsymbol{\Omega}_t, t, \mathbf{p})\right). \tag{26}$$

Here, θ represents the learnable parameters of an NN, and $\boldsymbol{\mu}_\theta(\boldsymbol{\Omega}_t, t, \mathbf{p})$ and $\boldsymbol{\Sigma}_\theta(\boldsymbol{\Omega}_t, t, \mathbf{p})$ represent the mean and the covariance function, respectively, with the independent variables being $\boldsymbol{\Omega}_t$, t, and $\mathbf{p}$, and the parameters being θ. The key to the reverse process lies in training the parameters θ to make the reverse Markov chain (25) match the inversion form (20) of the forward Markov chain as closely as possible.

The parameters θ in (26) are trained through the minimization of the variational upper bound on the negative log-likelihood $\mathbb{E}_{q(\boldsymbol{\Omega})}[-\ln p_\theta(\boldsymbol{\Omega}|\mathbf{p})]$, as detailed in [14].

$$\begin{aligned}\mathcal{L} &= \mathbb{E}_{q(\boldsymbol{\Omega}_{0:T})}\left[\ln q(\boldsymbol{\Omega}_{1:T}|\boldsymbol{\Omega}_0) - \ln p_\theta(\boldsymbol{\Omega}_{0:T}|\mathbf{p})\right] \\ &= \mathbb{E}_{q(\boldsymbol{\Omega}_{0:T})}\left[D\left(q(\boldsymbol{\Omega}_T|\boldsymbol{\Omega}_0)||p(\boldsymbol{\Omega}_T)\right)\right. \\ &\quad + \sum_{t=2}^{T} D\left(q(\boldsymbol{\Omega}_{t-1}|\boldsymbol{\Omega}_t, \boldsymbol{\Omega}_0)||p_\theta(\boldsymbol{\Omega}_{t-1}|\boldsymbol{\Omega}_t, \mathbf{p})\right) \\ &\quad \left. - \ln p_\theta(\boldsymbol{\Omega}_0|\boldsymbol{\Omega}_1, \mathbf{p})\right].\end{aligned} \tag{27}$$

Here, $D\left[\cdot||\cdot\right]$ represents the KL divergence. In order to effectively train the parameters θ, according to (27), it is necessary to design the corresponding reverse transition probability densities $p_\theta(\boldsymbol{\Omega}_{t-1}|\boldsymbol{\Omega}_t, \mathbf{p})$ at different time steps t. The $q(\boldsymbol{\Omega}_{t-1}|\boldsymbol{\Omega}_t, \boldsymbol{\Omega}_0)$ defined in (21) provides guidance for designing $p_\theta(\boldsymbol{\Omega}_{t-1}|\boldsymbol{\Omega}_t, \mathbf{p})$ in (26).

According to (26), the reverse transition probability density $p_\theta(\boldsymbol{\Omega}_{t-1}|\boldsymbol{\Omega}_t, \mathbf{p})$ only has information about the previous step $\boldsymbol{\Omega}_t$. Therefore, based on the expression of the mean of $q(\boldsymbol{\Omega}_{t-1}|\boldsymbol{\Omega}_t, \boldsymbol{\Omega}_0)$ in (22), the mean of $p_\theta(\boldsymbol{\Omega}_{t-1}|\boldsymbol{\Omega}_t, \mathbf{p})$ can be modeled as

$$\boldsymbol{\mu}_\theta(\boldsymbol{\Omega}_t, t, \mathbf{p}) = \frac{\sqrt{\alpha_{t-1}}\beta_t}{1-\alpha_t}\boldsymbol{\Omega}_\theta(\boldsymbol{\Omega}_t, t, \mathbf{p}) + \frac{\sqrt{1-\beta_t}(1-\alpha_{t-1})}{1-\alpha_t}\boldsymbol{\Omega}_t, \tag{28}$$

where $\boldsymbol{\Omega}_\theta(\boldsymbol{\Omega}_t, t, \mathbf{p})$ represents the prediction function for $\boldsymbol{\Omega}_0$ given the observation $\boldsymbol{\Omega}_t$. This function is formulated in accordance with (18) as

$$\boldsymbol{\Omega}_\theta(\boldsymbol{\Omega}_t, t, \mathbf{p}) = \frac{1}{\sqrt{\alpha_t}}\boldsymbol{\Omega}_t - \sqrt{\frac{1-\alpha_t}{\alpha_t}}\boldsymbol{\epsilon}_\theta(\boldsymbol{\Omega}_t, t, \mathbf{p}), \tag{29}$$

where $\boldsymbol{\epsilon}_\theta(\cdot)$ is a 'denoising' function mapping from $\mathbb{R}^{N_a \times N_d} \times \mathbb{R}^{(1+2)}$ to $\mathbb{R}^{N_a \times N_d}$, realized using an NN parameterized by θ

$$\begin{aligned}
&q(\mathbf{\Omega}_{t-1}|\mathbf{\Omega}_t,\mathbf{\Omega}_0) = q(\mathbf{\Omega}_t|\mathbf{\Omega}_{t-1})\frac{q(\mathbf{\Omega}_{t-1}|\mathbf{\Omega}_0)}{q(\mathbf{\Omega}_t|\mathbf{\Omega}_0)}\\
&= \frac{\mathcal{N}\left(\mathbf{\Omega}_t;\sqrt{1-\beta_t}\mathbf{\Omega}_{t-1},\beta_t\mathbf{I}_G\right)\mathcal{N}\left(\mathbf{\Omega}_{t-1};\sqrt{\alpha_{t-1}}\mathbf{\Omega}_0,(1-\alpha_{t-1})\mathbf{I}_G\right)}{\mathcal{N}\left(\mathbf{\Omega}_t;\sqrt{\alpha_t}\mathbf{\Omega}_0,(1-\alpha_t)\mathbf{I}_G\right)}\\
&\propto \exp\left\{-\left[\frac{(\mathbf{\Omega}_t-\sqrt{1-\beta_t}\mathbf{\Omega}_{t-1})^2}{2\beta_t}+\frac{(\mathbf{\Omega}_{t-1}-\sqrt{\alpha_{t-1}}\mathbf{\Omega}_0)^2}{2(1-\alpha_{t-1})}-\frac{(\mathbf{\Omega}_t-\sqrt{\alpha_t}\mathbf{\Omega}_0)^2}{2(1-\alpha_t)}\right]\right\}\\
&= \exp\left\{-\frac{1}{2}\left[\frac{(\mathbf{\Omega}_t-\sqrt{1-\beta_t}\mathbf{\Omega}_{t-1})^2}{\beta_t}+\frac{(\mathbf{\Omega}_{t-1}-\sqrt{\alpha_{t-1}}\mathbf{\Omega}_0)^2}{1-\alpha_{t-1}}-\frac{(\mathbf{\Omega}_t-\sqrt{\alpha_t}\mathbf{\Omega}_0)^2}{1-\alpha_t}\right]\right\}\\
&= \exp\left\{-\frac{1}{2}\left[\frac{-2\sqrt{1-\beta_t}\mathbf{\Omega}_{t-1}\mathbf{\Omega}_t+(1-\beta_t)\mathbf{\Omega}_{t-1}^2}{\beta_t}+\frac{\mathbf{\Omega}_{t-1}^2-2\sqrt{\alpha_{t-1}}\mathbf{\Omega}_0\mathbf{\Omega}_{t-1}}{1-\alpha_{t-1}}+C\left(\mathbf{\Omega}_t,\mathbf{\Omega}_0\right)\right]\right\}\\
&\propto \exp\left\{-\frac{1}{2}\left[\frac{-2\sqrt{1-\beta_t}\mathbf{\Omega}_{t-1}\mathbf{\Omega}_t+(1-\beta_t)\mathbf{\Omega}_{t-1}^2}{\beta_t}+\frac{\mathbf{\Omega}_{t-1}^2-2\sqrt{\alpha_{t-1}}\mathbf{\Omega}_0\mathbf{\Omega}_{t-1}}{1-\alpha_{t-1}}\right]\right\}\\
&= \exp\left\{-\frac{1}{2}\left[\left(\frac{1-\beta_t}{\beta_t}+\frac{1}{1-\alpha_{t-1}}\right)\mathbf{\Omega}_{t-1}^2-2\left(\frac{\sqrt{1-\beta_t}\mathbf{\Omega}_t}{\beta_t}+\frac{\sqrt{\alpha_{t-1}}\mathbf{\Omega}_0}{1-\alpha_{t-1}}\right)\mathbf{\Omega}_{t-1}\right]\right\}\\
&\propto \mathcal{N}\left(\mathbf{\Omega}_{t-1};\widetilde{\boldsymbol{\mu}}_t(\mathbf{\Omega}_t,\mathbf{\Omega}_0),\widetilde{\beta}_t\mathbf{I}_G\right),
\end{aligned} \tag{24}$$

to predict the added noise. And the structure of the network will be introduced in the next subsection. By substituting (29) into (28), the final mean is obtained as

$$\boldsymbol{\mu}_\theta(\mathbf{\Omega}_t,t,\mathbf{p}) = \frac{1}{\sqrt{1-\beta_t}}\left(\mathbf{\Omega}_t-\frac{\beta_t}{\sqrt{1-\alpha_t}}\boldsymbol{\epsilon}_\theta(\mathbf{\Omega}_t,t,\mathbf{p})\right). \tag{30}$$

Furthermore, the covariance of (26) can be determined in accordance with (23), a default covariance given by the forward process resulting in

$$\mathbf{\Sigma}_\theta(\mathbf{\Omega}_t,t,\mathbf{p}) = \widetilde{\beta}_t\mathbf{I}_G. \tag{31}$$

Given that the KL divergence between the d-dimensional multivariate Gaussian distribution has

$$\begin{aligned}
&\mathrm{D}\left[\mathcal{N}\left(\mathbf{z}_1;\boldsymbol{\mu}_1,\mathbf{\Sigma}_1\right)\|\mathcal{N}\left(\mathbf{z}_2;\boldsymbol{\mu}_2,\mathbf{\Sigma}_2\right)\right] = \frac{1}{2}\ln\frac{\det\left(\mathbf{\Sigma}_1\right)}{\det\left(\mathbf{\Sigma}_2\right)}-d\\
&+\operatorname{tr}\left(\mathbf{\Sigma}_2^{-1}\mathbf{\Sigma}_1\right)+\left(\boldsymbol{\mu}_2-\boldsymbol{\mu}_1\right)^{\mathsf{T}}\mathbf{\Sigma}_2^{-1}\left(\boldsymbol{\mu}_2-\boldsymbol{\mu}_1\right)
\end{aligned} \tag{32}$$

Substituting (18), (30), and (31) into (27) and disregarding constant terms leads to a reduction of the training loss (27) to

$$\begin{aligned}
\mathcal{L}-C &= \sum_{t=1}^{T}\mathbb{E}_{\mathbf{\Omega}_0,\boldsymbol{\epsilon}_t}\left[\gamma_t\left\|\boldsymbol{\epsilon}_t-\boldsymbol{\epsilon}_\theta(\mathbf{\Omega}_t,t,\mathbf{p})\right\|_2^2\right]\\
&= \sum_{t=1}^{T}\mathbb{E}_{\mathbf{\Omega}_0,\boldsymbol{\epsilon}_t}\left[\gamma_t\left\|\boldsymbol{\epsilon}_t-\boldsymbol{\epsilon}_\theta(\sqrt{\alpha_t}\mathbf{\Omega}_0+\sqrt{1-\alpha_t}\boldsymbol{\epsilon}_t,t,\mathbf{p})\right\|_2^2\right],
\end{aligned} \tag{33}$$

where $\gamma_t = \frac{\beta_t}{2(1-\alpha_{t-1})(1-\beta_t)}, \boldsymbol{\epsilon}_t \sim \mathcal{N}(0,\mathbf{I}_G)$. Notably, experiments conducted in [14] have illustrated that a simplified variant of equation (33) tends to enhance the generation quality while also being easier to implement. In this simplified variant, the weighting factor γ_t is omitted, and t is uniformly sampled from the range of 1 to T. For each time step

$$\mathcal{L} = \mathbb{E}_{\mathbf{\Omega}_0,\boldsymbol{\epsilon}_t}\left\|\boldsymbol{\epsilon}_t-\boldsymbol{\epsilon}_\theta(\sqrt{\alpha_t}\mathbf{\Omega}_0+\sqrt{1-\alpha_t}\boldsymbol{\epsilon}_t,t,\mathbf{p})\right\|_2^2. \tag{34}$$

E. Network Structure and Specific Training the Conditional Diffusion Model

The architecture of the denoising NN $\boldsymbol{\epsilon}_\theta$ proposed is based on the well-known U-Net network, which has been adopted in DDPM [14]. However, we have adjusted this architecture according to the requirements of our problem. First and foremost, the difference lies in the input layer. We integrate the independent variables $\mathbf{\Omega}_t$ and $\mathbf{p}$ from equation (34) into a single input tensor through a series of carefully crafted TensorFlow layers, which is then used for subsequent denoising processing, as shown in Fig. 4. Our approach to this integration unfolds in several key steps:

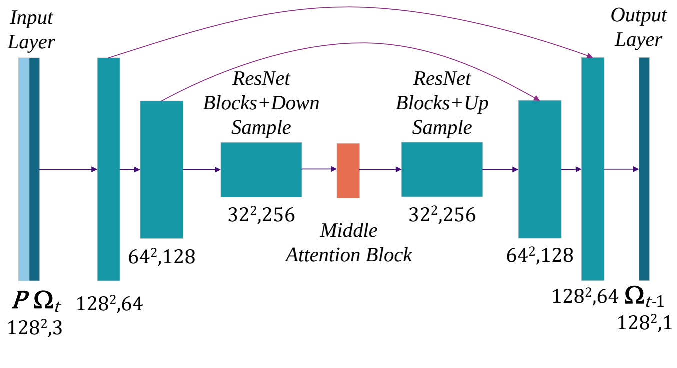

Fig. 4. Description of the U-Net architecture with skip connections. The position information $\mathbf{p}$ is replicated to the target dimensions, and concatenated with the noisy $\mathbf{\Omega}_t$ along channel dimension.

- **Input Layer Creation**: We first need to establish an input layer with the purpose of receiving the noisy training data Ω_t. This input layer needs to explicitly specify the width as $N_a = 128$, the height as $N_d = 128$, and the number of channel count as 1. It serves as the foundational gateway into the entire NN architecture.
- **Reshaping the Condition Information**: The next step involves reshaping the positional coordinate vector $\mathbf{p}$.

While $\mathbf{p}$ exists in the mathematical space $\mathbb{R}^2$, it needs to be expanded into a more expressive form $\mathbb{R}^{1\times1\times2}$. By introducing two single dimensions, $\mathbf{p}$ is adjusted into a tensor form. This seemingly minor change enables the model to handle and utilize the positional information more effectively.

- **Expansion positional information with 'Tile' layer**: To fullly utilize the potential of $\mathbf{p}$, the NN adopts a 'Tile' function. This function effectively amplifies the positional information by replicating the $\mathbf{p}$ tensor along defined dimensions. Given the specific dimensions of the training data, $\mathbf{p}$ is replicated N_a times along the 'width' dimension and N_d times along the 'height' dimension. This ultimately expands the conditional information into a tensor of dimensions $N_a \times N_d \times 2$, enabling 'pixel-level' conditional control of the training data $\mathbf{\Omega}_t$.
- **Fusion through 'Concatenate' layer**: This layer plays a crucial role in fusing the information of two distinct tensors - the noisy statistical CSI training data $\mathbf{\Omega}_t$ and the enriched conditional positional information derived from the modified $\mathbf{p}$. These tensors, sharing the same width and height dimensions but unique information, are seamlessly concatenated along the channel dimension. The result is an input tensor of dimensions $N_a \times N_d \times 3$, combining statistical information with positional conditions.

Next we will introduce the widely used U-Net network and our improvements to it as the main structure of the DM (see Fig. 4 for details). U-Net is a convolutional neural network (CNN) architecture firstly designed for image segmentation tasks, which has found successful applications in various fields, including computer vision, remote sensing, and more. The U-Net architecture is characterized by its input and output data that have the same 'width' and 'height' dimensions, which resembles an U-shaped structure. It consists of two main parts: the contracting path and the expansive path.

- **Contracting Path**: The contraction path is used with residual convolutional neural networks (i.e., ResNet blocks). A ResNet block consists of a series of convolutional layers, activation function layers and 'skip connections'. Subsequently, the output of the feature extraction through the ResNet block is connected to a downsampling layer that reduces the width and height of the feature representation while increasing the depth. The purpose of the contracting path is to capture the main feature information in the input and create intermediate feature representations.
- **Expansive Path**: The expansive path aims to generate the pixel-wise statistical channel information. It consists of many ResNet blocks and a series of up-sampling layers. Each up-sampling step gradually increases the width and height of the intermediate feature representation while reducing its depth.
- **Skip Connections**: In U-Net, these skip connections directly link the intermediate feature representations from the contraction path to the corresponding layers in the expansion path, as shown by the purple lines in Fig. 4. The skip connections help the network recover spatial information lost during down-sampling in the contracting path. This facilitates gradient flow and avoids introducing representation bottlenecks.

TABLE I
SPECIFIC U-NET ARCHITECTURE PARAMETERS

Parameter	Value
Kernel size	3
Channel dim[1]	64
Depth multipliers[2]	$\{1, 2, 4\}$
ResNet blocks[3]	2
#Params	150M

[1] Channel dim is the depth dimension of the first conv-layer.
[2] Depth multipliers apply to the subsequent conv-layer.
[3] ResNet blocks apply to the number before down-sample layer and up-sample layer.

- **Final Output Layer**: The last layer of the U-Net architecture utilizes a 1×1 convolutional kernel to map the feature maps from the contracting and expanding paths to the desired output depth (1 in the specific simulation), corresponding to the channel of $\mathbf{\Omega}_t$.

In summary, Fig. 4 depicts the main architecture of the U-Net network. This denoising U-Net takes a noisy statistical CSI $\mathbf{\Omega}_t$ and conditioning $\mathbf{p}$ that has been up-sampled to the target dimension as input. Task dependent parameters are summarized in Table I. The network architecture includes over 150 million parameters for generation and denoising. The design parameters in Table I are appropriately chosen based on the excellent results in the existing field of computer vision and the specific conditions of the experimental equipment. The model performance is proportional to the parameter capacity, so the number of layers can be increased further if conditions permit. In the next subsection, more training hyper-parameters will be introduced.

In addition to the noisy $\mathbf{\Omega}_t$ and the positional conditional information $\mathbf{p}$, the NN denoising model $\boldsymbol{\epsilon}_\theta$ needs to take the timestep t as input. The timestep t is required since denoising a signal requires different operations at different levels of noise. We transform the timestep t using 'sinusoidal' embeddings, similarly with the Transformers network. For a given timestep t, the 'Time-embedding' layer transforms it into a $d = 256$-dimensional vector denoted as 'SE', where each element is

$$\begin{aligned} \mathrm{SE}_{2i}(t) &= \sin(t/10000^{2i/d}) \\ \mathrm{SE}_{2i+1}(t) &= \cos(t/10000^{2i/d}). \end{aligned} \tag{35}$$

This embeds the timestep t into the ResNet block and helps the network to be highly sensitive to the added noise level, which is crucial for good performance.

The ResNet blocks serve as essential components within the U-Net framework, corresponding to the 'light green main module' in Fig. 4, playing a pivotal role in facilitating the training of very deep networks (e.g. our model has more than 50 layers). Their primary function is to address the issue of vanishing gradients that occurs as the network layers increase. Key characteristics of the ResNet Block used in our network (seen in Fig. 5) are shown below:

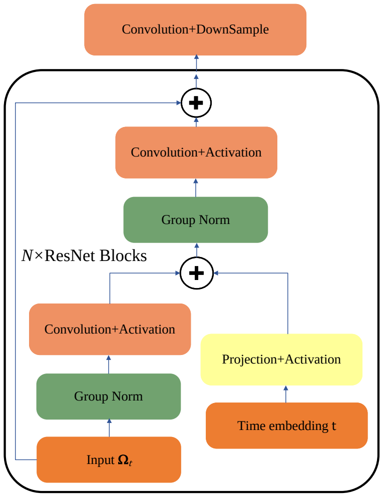

Fig. 5. Description of the ResNet block architecture with skip connections, which forms the basic building block of the overall U-Net structure in Fig. 4.

- **Adaptive Depth**: A characteristic of the ResNet block is its ability to adapt to convolution operations with various channel numbers (e.g. $\{64, 128, 256\}$ for each light green module in Fig. 4). It achieves this by dynamically adjusting depth 'multipliers' hyper-parameter in Table I, ensuring compatibility with various network architectures. When integrating ResNet blocks into the entire U-Net network, this adaptability allows for flexible hyper-parameter configuration based on the specific prediction accuracy and the experimental equipment.
- **Temporal t Integration**: In diffusion process, timestep t holds paramount importance. To address this need, the ResNet block incorporates an additional input denoted as 'Time Embedding' of t in (35) and processes it using a specified projection and activation. This embedding significantly enhances the ResNet's capability to capture time-sensitive features. When employed within U-Net networks, this temporal integration becomes essential for involving sequential data.
- **Normalization and Activation**: The ResNet block leverages group normalization that different from [27] and 'Swish' activation function to ensure a stable and efficient flow of information, thereby augmenting feature learning. This combination enhances the reliability of the ResNet's performance, making it a reliable choice when enhancing the whole U-Net models.
- **Residual Connection**: The core of the ResNet block is the use of residual connections. When the input and output channel dimensions align, a direct residual connection is established. If this alignment is not met, a 1×1 convolutional layer adjusts the channel dimension. This unique design plays a crucial role in mitigating the vanishing gradient problem, a challenge often encountered in deep networks. In the U-Net framework, these residual connections enhance gradient flow during both contracting and expansive path, enabling the training of deeper and more effective models.

Incorporating ResNet blocks into the U-Net architecture empowers the model with adaptability, enhances temporal understanding, improves the feature learning and robust gradient flow. These advantages collectively contribute to more effective and accurate deep learning model, particularly in the CSI generation where complex spatial and temporal information processing is required.

In summary, we have introduced several key enhancements to the specific U-Net network [14]. Firstly, $\boldsymbol{\epsilon}_\theta$ now includes three inputs: the noisy $\boldsymbol{\Omega}_t$, positional conditional information $\mathbf{p}$, and the time step t. This three-input scheme enables the network to denoise data while considering their temporal characteristics. Furthermore, once the feature map dimensions reach the bottleneck (e.g., 32×32 dimensions as shown in Fig. 4), a self-attention mechanism is introduced between ResNet blocks. This self-attention mechanism enhances the network's ability to focus on relevant information, which is particularly valuable when processing complex data. Moreover, we've replaced 'Weight Normalization' in [14] with 'Group Normalization', promoting more stable and efficient information flow within the network, ultimately improving its feature learning capabilities. Notably, in our specific applications, we found that configuring the 'groups' parameter in the 'Group Normalization layer' to 8 yields superior results. The whole proposed algorithm is given in Algorithms 1 and 2.

Algorithm 1 Training a diffusion model $\boldsymbol{\epsilon}_\theta$

```
repeat
Give batch of training data pairs $(\boldsymbol{\Omega}_0^{RP}, \mathbf{p}^{RP}) \sim q(\boldsymbol{\Omega}_0^{RP}|\mathbf{p}^{RP})$
  in the dataset $\mathcal{D}$.
Give time step $t$ that satisfies a uniform distribution $t$ $\sim$Uni
  $(\{1, \cdots, T\})$.
Randomly generate a pure Gaussian noise matrix $\boldsymbol{\epsilon}_t \sim$
  $\mathcal{N}(\boldsymbol{\epsilon}_t; \mathbf{0}, \mathbf{I}_G)$.
According equation (18) noise the training data $\boldsymbol{\Omega}_0^{RP}$.
Feed the noisy data, time step and the position
  condition through the model. Update model parameters
  $\theta$ with the mean squared error as our loss function
  $\nabla_\theta \left\| \boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\theta \left( \sqrt{\alpha_t} \boldsymbol{\Omega}_0^{RP} + \sqrt{1-\alpha_t} \boldsymbol{\epsilon}_t, t, \mathbf{p}^{RP} \right) \right\|_2^2$
Get the gradients and update the weights of the network. Then
  update the weight values for the network with EMA weights
until converged and save the NN parameters $\theta$.
```

Algorithm 2 Generation by T iterative steps in reverse process

```
Load the trained NN $\theta$ and give a sampling loop.
repeat
Give a pure Gaussian noise matrix $\boldsymbol{\Omega}_T \sim \mathcal{N}(\boldsymbol{\Omega}_T; \mathbf{0}, \mathbf{I}_G)$ and the
  UT's position coordinate $\mathbf{p}_k^{UT}$
for $t = T, \cdots, 1$ do
    $\mathbf{z} \sim \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I}_G)$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
    Reverse the diffusion process by the trained NN $\theta$
    and the equation (30) and (31) $\boldsymbol{\Omega}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \boldsymbol{\Omega}_t -$
    $\sqrt{\frac{1-\alpha_t}{\alpha_t}} \boldsymbol{\epsilon}_\theta(\boldsymbol{\Omega}_t, t, \mathbf{p}_k^{UT}) + \widetilde{\beta}_t \mathbf{z}$
end for
return $\boldsymbol{\Omega}_0^{UT}$ as the $k$-th UT's generation statistical CSI $\hat{\boldsymbol{\Omega}}_k^{UT}$
  conditioned on the given position $\mathbf{p}_k^{UT}$
```

IV. Simulation Results

In this section, we present simulation results to show the effectiveness of the proposed methods. We use a geometry-based model to simulate the massive MIMO wireless transmission environment, and consider a 2D propagation scenario, similarly to [2]. The BS is equipped with a ULA. In Fig. 6, we provide a 2D planar schematic diagram to demonstrate the configuration of our simulation. This plane's coordinates (X, Y) correspond to the X- and Y-axes. Assume that the BS is at the coordinate origin $(0, 0)$ m, and its equipped ULA is parallel to the Y-axis. The targeted area is a square area with its center at $(200, 0)$ m and sides of each 50 m long. There are 50 scatterers per km^2 of space [2], [17]. Each scatterer is represented as a collection of reflection points on its surface, and a path is any transmission line connecting the UT and BS that is not blocked by other scatterers. We consider the propagation of geometry-based single bounces (GBSB) to simplify the model, but without loss of generality [28], [29]. For simulation, we employ a typical set of wireless and system parameters, as indicated in Table II. We generate the channels according to the refined beam-based channel in (11) and (13). In this case, the accurate statistical CSIs are known, and thus can be used to evaluate the accuracy of the generated CSIs.

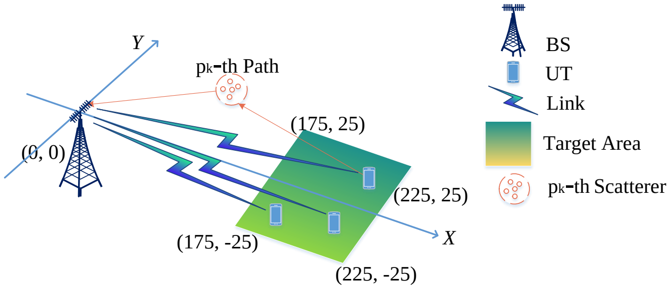

Fig. 6. The schematic diagram of the massive MIMO systems.

During the offline training phase, the target area is discretized along its geometric space dimensions, that is, along the X-axis and Y-axis, respectively. Taking Δ_x and Δ_y as the minimum interval units, and meshes the target area into a 2D matrix graph with N_x rows and N_y columns. The position coordinates of each grid point are recorded as the true positions $\mathbf{p}_i^{RP}$ of the RPs, and the statistical CSI of the RPs are obtained by channel model. We can get the training dataset of the RPs, including $\{\mathbf{\Omega}_i^{RP}, \mathbf{p}_i^{RP}\}_{i=1}^{N_x N_y}$. Similarly, during the online mode, generate 500 randomly distributed UTs in the whole target area for the testing UT set $\{\mathbf{\Omega}_i^{UT}, \mathbf{p}_i^{UT}\}_{i=1}^{500}$ to evaluate the NN generation $\hat{\mathbf{\Omega}}_i^{UT}$ performance. Since the accurate $\mathbf{\Omega}_i^{UT}$ is known, we can use the normalized mean squared error (NMSE) between the generated $\hat{\mathbf{\Omega}}_i^{UT}$ and the accurate $\mathbf{\Omega}_i^{UT}$ to evaluate the accuracy. The NMSE in dB is defined as

$$\mathrm{NMSE(dB)} = 10\log_{10}\left(\frac{1}{500}\sum_{i=1}^{500}\frac{\left\|\hat{\mathbf{\Omega}}_i - \mathbf{\Omega}_i\right\|_{\mathrm{F}}^2}{\|\mathbf{\Omega}_i\|_{\mathrm{F}}^2}\right). \quad (36)$$

The batch size is 32 and the Adam optimizer is used in this paper, with the learning rate 0.0002 for good performance. We utilize MATLAB 2020a to calculate and save the statistical CSI and UTs' and RP's coordinates for the methods described above. TensorFlow 2.9 is used to train and test the DNN. Our simulation was run on a Servers equipped with an Intel Xeon 5318Y CPU and a Geforce GTX 4090 24GB GPU.

TABLE II
SYSTEM AND MODEL PARAMETERS

Parameter	Value
Minimum interval units	$\Delta_x = 1$ m
Minimum interval units	$\Delta_y = 1$ m
Number of rows of the grid	$N_x = 50$
Number of colums of the grid	$N_y = 50$
Number of UTs	500
Carrier frequency	$f = 4.8$ GHZ
Bandwidth	$B = 20$ MHZ
Subcarrier spacing	$\Delta f = 15$ kHz
Subcarrier numbers	$N_c = 2048$
Cyclic prefix length	$N_g = 144$
Sampling interval	$T_s = 30.4$ ns

A. Training Stage

First, we evaluate the systems under $N = 64, N_a = 128$, and $BW = 20$ MHz, $N_d = 128$ wireless channel parameters. Thus the size of input tensors is 128×128 $(N_a \times N_d)$. The channel gains $\beta_{k,p}$ follow the standard complex Gaussian distribution. Our NN architecture has been described in detail in the last Section III. The hyper-parameters dropout and Exponential Moving Average (EMA) rate are set to 0.1 and 0.999, respectively (using the EMA technique to update NN parameters, the model weights are updated during training by combining the product of the current gradient and the EMA value with the product of the previous gradient weight and (1-EMA) value). The diffusion step number T is set to 1000. The diffusion noise variance follows an linear schedule from $\beta_1 = 0.0001$ to $\beta_{1000} = 0.02$. The training and testing datasets consist of 10,000 and 5,000 randomly generated channels based on the presented channel model.

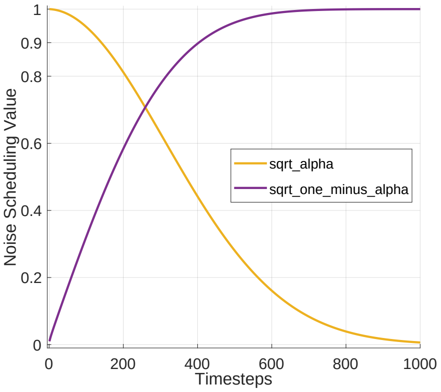

Fig. 7. Noise scheduling value $\sqrt{\alpha_t}$ and $\sqrt{1-\alpha_t}$.

To demonstrate the training process of Algorithm 1, we first need to obtain the noisy data $\mathbf{\Omega}_t^{RP}$ corresponding to any time step t by using a pre-designed noise scheduler α based on equations (17) and (18). We then use this $\mathbf{\Omega}_t^{RP}$ as input to train the U-Net to predict the noise added at time step t using the mean squared error (MSE) loss function (33). We begin by presenting the noise addition strategy used in simulation. The mathematical notation of the diffusion process can appear challenging in Section III-C and D, but fortunately, the scheduler α handles all of that. We plot $\sqrt{\alpha_t}$ (labeled as sqrt_alpha) and $\sqrt{1-\alpha_t}$ (labeled as sqrt_one_minus_alpha) in Fig. 7 to observe how the training data $\mathbf{\Omega}_0^{RP}$ and the pure noise matrix $\boldsymbol{\epsilon}_t$ are scaled and mixed across different time steps t.

Next, we construct the U-Net model as described in Sections III-E, consisting of various modules, including the input layer (comprising noisy data $\mathbf{\Omega}_t^{RP}$ at time step t and positional conditional information $\mathbf{p}^{RP}$), down-sampling modules (comprising two ResNet blocks and a down-sampling layer), middle modules (comprising two ResNet blocks and a self-attention block), up-sampling modules (comprising two ResNet blocks and an up-sampling layer), and finally connecting to an output layer. Our U-Net network accounts for the influence of time step t, enabling the prediction of noise levels present in noisy data $\mathbf{\Omega}_t^{RP}$ at any given time step t. Additionally, we consider the impact of physical spatial position information $\mathbf{p}^{RP}$, allowing the model to establish the mapping between RP's CSI and corresponding physical position of RP. In Fig. 8, we observe a decreasing trend in noise prediction error with increasing training epochs, indicating the increasingly accurate noise prediction of the model. The reduction in training loss corroborates the effectiveness of our method, demonstrating that through training, the model can predict the noise levels added to statistical CSI data corresponding to specific RP's positions $\mathbf{p}^{RP}$ at any time step t, and after T time steps, the statistical CSI data will transform into a pure noise matrix. The variation in training loss function aided in adjusting network parameters to achieve optimal performance, and we preserve the final trained model for using in Algorithm 2 during the generation stage.

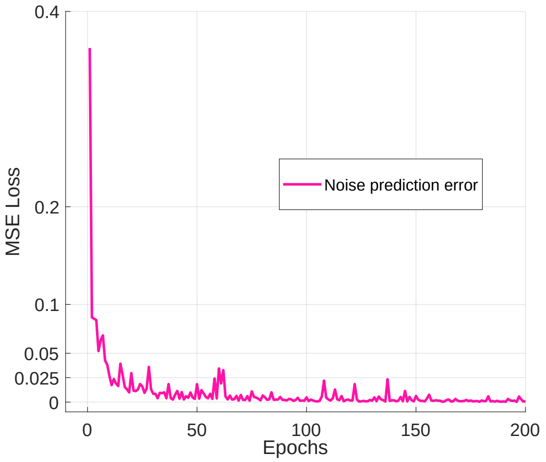

Fig. 8. The noise prediction MSE loss of the training stage by U-Net.

B. Generation Stage

Firstly, we introduce the testing environment setup. During the generation phase, 100 UTs are uniformly randomly distributed within the target area, assuming that the BS can obtain their positional coordinates $\mathbf{p}_k^{UT}$. To demonstrate the performance of our Algorithm 2 generation method, we first introduce two statistical CSI estimation methods for comparison. One is referred to as the 'interpolation' method, where the BS is assumed to have obtained the positional coordinate of the UT $\mathbf{p}_k^{UT}$. By calculating the physical distance with the RPs' coordinates, the four nearest RPs to the UTs are found, and the physical distances are used as weights $w_i = \frac{\left\|\mathbf{p}_k^{UT}-\mathbf{p}_i^{RP}\right\|_2}{\sum_{i=1}^{4}\left\|\mathbf{p}_k^{UT}-\mathbf{p}_i^{RP}\right\|_2}$ to compute the weighted average of the statistical CSI $\mathbf{\Omega}_i^{RP}$ of the four RPs as the UTs' statistical CSI estimation result:

$$\hat{\mathbf{\Omega}}_k = \sum_{i=1}^{4} w_i \mathbf{\Omega}_i^{RP}. \tag{37}$$

The other method is the statistical CSI estimation from reference [26], where the BS can obtain T_p receiving pilot signals from the UTs. By computing the KL divergence, an optimization problem is solved to obtain the UT' statistical CSI estimation value. We compare the NMSE performance of the method for $T_p = 3$ and $T_p = 10$ as a baseline with our method.

For our diffusion model-based generation method, our simulation approach is as follows: after every 1000 epochs of training, the trained model parameters are saved. Then, utilizing the model's powerful parallel computing capability and known positional coordinates of 100 test UTs, the statistical CSIs of the test UTs are simultaneously generated as the UT' statistical CSI estimation $\hat{\mathbf{\Omega}}_k$, and the NMSE is calculated and saved. Since the performance of the comparative methods is not based on AI, their performance does not improve with the progress of training. Our final simulation results are shown in Fig. 9.

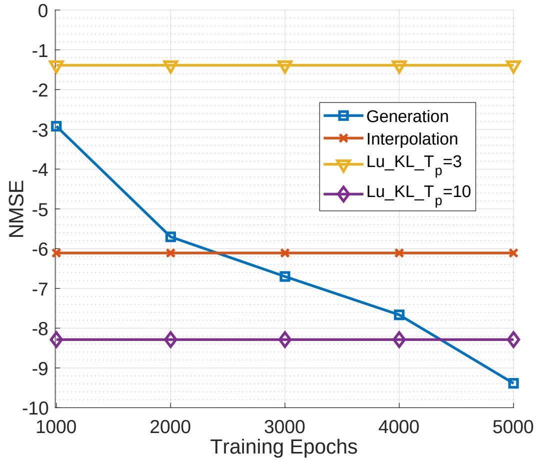

Fig. 9. The comparison of NMSE v.s. training epochs for different estimation methods of statistical CSI.

By comparing the performance lines in yellow, blue, and purple, it can be observed that increasing the number of pilots can improve the performance of the method from reference [26]. For example, when the number of pilots is $T_p = 10$, it can estimate 24 UTs simultaneously and achieve an NMSE of -8.28. In comparison, the proposed generation method does not require pilot information and can simultaneously generate the statistical CSI of 100 UTs with only known positional coordinates. Moreover, as training progresses, the performance gradually improves and surpasses the baseline method, and the NMSE gradually increases from -2.92 to -9.387. The simulation results demonstrate the accuracy and effectiveness of our method. Note that all training processes above are conducted offline, and the generation phase can select the saved network parameters as needed, without additional computational complexity in the generation phase. By comparing the performance lines in red and blue, it can be seen that when the training epochs of our method exceed 3000, an NMSE of -6.97 can be achieved, surpassing the ‘interpolation’ method’s NMSE. Although the ‘interpolation’ method is natural and convenient, for higher accuracy, the proposed generation method can achieve more satisfactory results.

V. Conclusion

This paper introduces the concept of the DToC. By treating UT positions as physical objects and statistical CSI as virtual digital object, DToC enables predictive analytics for subsequent communication tasks by observing trends in UT’s statistical CSI resulting from the changes in UT physical position. Additionally, the proposal of a conditional DM, incorporating UT position information, facilitates the generation of statistical CSI under specific UT position, thus enabling precise mapping between UT positions and statistical CSI. Simulation results show the superiority of our DToC framework over previous statistical CSI estimation methods, presenting its ability to generate statistical CSI for numerous UTs simultaneously based solely on their positions, without the need for pilot signals. In essence, the DToC framework presents a promising road for enhancing communication efficiency and accuracy in sensing-integrated networks.

References

[1] F. Liu, Y. Cui, C. Masouros, J. Xu, T. X. Han, Y. C. Eldar, and S. Buzzi, “Integrated sensing and communications: toward dual-functional wireless networks for 6G and beyond,” *IEEE J. Select. Areas Commun.*, vol. 40, no. 6, pp. 1728–1767, Jun. 2022.

[2] X. Sun, C. Wu, X. Q. Gao, and G. Y. Li, “Fingerprint-based localization for massive MIMO-OFDM system with deep convolutional neural networks,” *IEEE Trans. Veh. Technol.*, vol. 68, no. 11, pp. 10 846–10 857, Nov. 2019.

[3] W. Shi, W. Xu, X. You, C. Zhao, and K. Wei, “Intelligent reflection enabling technologies for integrated and green internet-of-everything beyond 5G: communication, sensing, and security,” *IEEE Wireless Commun.*, vol. 30, no. 2, pp. 147–154, Apr. 2023.

[4] A. Liu, Z. Huang, M. Li, Y. Wan, W. Li, T. X. Han, C. Liu, R. Du, D. K. P. Tan, J. Lu, Y. Shen, F. Colone, and K. Chetty, “A survey on fundamental limits of integrated sensing and communication,” *IEEE Commun. Surv. Tutorials*, vol. 24, no. 2, pp. 994–1034, 2022.

[5] X. Gong, X. Yu, X. Liu, and X. Q. Gao, “Machine learning-based fingerprint positioning for massive MIMO systems,” *IEEE Access*, vol. 10, pp. 89 320–89 330, 2022.

[6] J. A. del Peral-Rosado, R. Raulefs, J. A. Lopez-Salcedo, and G. Seco-Granados, “Survey of cellular mobile radio localization methods: from 1G to 5G,” *IEEE Commun. Surv. Tutorials*, vol. 20, no. 2, pp. 1124–1148, 2018.

[7] A. Zhang, M. L. Rahman, X. Huang, Y. J. Guo, S. Chen, and R. W. Heath, “Perceptive mobile networks: cellular networks with radio vision via joint communication and radar sensing,” *IEEE Veh. Technol. Mag.*, vol. 16, no. 2, pp. 20–30, Jun. 2021.

[8] Y. Cui, W. Yuan, Z. Zhang, J. Mu, and X. Li, “On the physical layer of digital twin: An integrated sensing and communications perspective,” *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 11, pp. 3474–3490, 2023.

[9] X. Shen, J. Gao, W. Wu, M. Li, C. Zhou, and W. Zhuang, “Holistic network virtualization and pervasive network intelligence for 6g,” *IEEE Communications Surveys Tutorials*, vol. 24, no. 1, pp. 1–30, 2022.

[10] A.-A. Lu, Y. Zhang, and X. Q. Gao, “Iterative generalized eigenvector precoder with deterministic equivalents for 3D massive MIMO,” *IEEE Trans. Veh. Technol.*, vol. 70, no. 11, pp. 11 784–11 795, Nov. 2021.

[11] Y. Pu, Z. Gan, R. Henao, X. Yuan, C. Li, A. Stevens, and L. Carin, “Variational autoencoder for deep learning of images, labels and captions,” in *Advances in Neural Information Processing Systems (NIPS)*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29. Curran Associates, Inc., 2016.

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in Neural Information Processing Systems (NIPS)*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds., vol. 27. Curran Associates, Inc., 2014.

[13] H. Cao, C. Tan, Z. Gao, G. Chen, P.-A. Heng, and S. Z. Li, “A survey on generative diffusion model,” *arXiv preprint arXiv:2209.02646*, 2022.

[14] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” in *Advances in Neural Information Processing Systems (NeurIPS)*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 6840–6851.

[15] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, “Hierarchical text-conditional image generation with clip latents,” 2022.

[16] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, 2022, pp. 10 684–10 695.

[17] X. Gong, A. Lu, X. Liu, X. Fu, X. Q. Gao, and X.-G. Xia, “Deep learning based fingerprint positioning for multi-cell massive MIMO-OFDM systems,” *IEEE Transactions on Vehicular Technology*, pp. 1–18, 2023.

[18] X. Gong, A.-a. Lu, X. Fu, X. Liu, X. Q. Gao, and X.-G. Xia, “Semi-supervised representation contrastive learning for massive mimo fingerprint positioning,” *IEEE Internet of Things Journal*, pp. 1–1, 2023.

[19] A.-A. Lu, X. Q. Gao, and C. Xiao, “Robust linear precoder design for 3D massive MIMO downlink with a posteriori channel model,” *IEEE Trans. Veh. Technol.*, vol. 71, no. 7, pp. 7274–7286, Jul. 2022.

[20] X. Yu, A.-A. Lu, X. Q. Gao, G. Y. Li, G. Ding, and C.-X. Wang, “HF skywave massive MIMO communication,” *IEEE Trans. Wireless Commun.*, vol. 21, no. 4, pp. 2769–2785, Apr. 2022.

[21] C. Wang, A.-A. Lu, X. Q. Gao, and Z. Ding, “Robust precoding for 3D massive MIMO configuration with matrix manifold optimization,” *IEEE Trans. Wireless Commun.*, vol. 21, no. 5, pp. 3423–3437, May 2022.

[22] C. Sun, X. Q. Gao, S. Jin, M. Matthaiou, Z. Ding, and C. Xiao, “Beam division multiple access transmission for massive MIMO communications,” *IEEE Trans. Commun.*, vol. 63, no. 6, pp. 2170–2184, Jun. 2015.

[23] X. Kuai, X. Yuan, W. Yan, H. Liu, and Y. J. Zhang, “Double-sparsity learning-based channel-and-signal estimation in massive MIMO with generalized spatial modulation,” *IEEE Trans. Commun.*, vol. 68, no. 5, pp. 2863–2877, May 2020.

[24] T. Wu, J. Liu, J. Liu, Z. Huang, H. Wu, C. Zhang, B. Bai, and G. Zhang, “A novel AI-based framework for AoI-optimal trajectory planning in uav-assisted wireless sensor networks,” *IEEE Transactions on Wireless Communications*, vol. 21, no. 4, pp. 2462–2475, 2022.

[25] A. Jagannath, J. Jagannath, and T. Melodia, “Redefining wireless communication for 6G: signal processing meets deep learning with deep unfolding,” *IEEE Transactions on Artificial Intelligence*, vol. 2, no. 6, pp. 528–536, 2021.

[26] A.-A. Lu, Y. Chen, and X. Q. Gao, “2D beam domain statistical csi estimation for massive mimo uplink,” *IEEE Transactions on Wireless Communications*, pp. 1–1, 2023.

[27] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[28] A. Zajic, G. Stuber, T. Pratt, and S. Nguyen, “Wideband MIMO mobile-to-mobile channels: Geometry-based statistical modeling with experimental verification,” *IEEE Trans. Veh. Technol.*, vol. 58, no. 2, pp. 517–534, Feb. 2009.

[29] C.-X. Wang, Z. Lv, X. Gao, X. You, Y. Hao, and H. Haas, “Pervasive wireless channel modeling theory and applications to 6g gbsms for all frequency bands and all scenarios,” *IEEE Transactions on Vehicular Technology*, vol. 71, no. 9, pp. 9159–9173, 2022.