Generative and Explainable AI for High-Dimensional Channel Estimation

Nghia Thinh Nguyen and Tri Nhu Do
Department of Electrical Engineering, Polytechnique Montréal, Montréal, Québec, Canada.
Emails: {nghia-thinh.nguyen, tri-nhu.do}@polymtl.ca

***Abstract*—In this paper, we propose a new adversarial training framework to address high-dimensional instantaneous channel estimation in wireless communications. Specifically, we train a generative adversarial network to predict a channel realization in the time-frequency-space domain, in which the generator exploits the third-order moment of the input in its loss function and applies a new reparameterization method for latent distribution learning to minimize the Wasserstein distance between the true and estimated channel distributions. Next, we propose an explainable artificial intelligence mechanism to examine how the critic discriminates the generated channel. We demonstrate that our proposed framework is superior to existing methods in terms of minimizing estimation errors. Additionally, we find that the critic's attention focuses on the high-power portion of the channel's time-frequency representation.**

***Index Terms*—5G NR, 3GPP, Channel estimation, MIMO, VAE, WGAN-GP, Third-order moment, Explainable AI**

I. Introduction

Channel state information (CSI) acquisition, particularly channel estimation in 5G New Radio (NR), presents unprecedented challenges, especially in handling high-dimensional data [1]. The increased degrees of freedom (DoF) related to the dimension of transmission pilots across the time, frequency, and space domains lead to higher computational complexity with traditional methods [2]. In this context, generative artificial intelligence (AI) emerges as a potential solution to these pressing challenges [3]. Recently, researchers have focused on enhancing realistic 5G NR pilot-based channel estimation techniques through advanced simulation frameworks like Sionna [4], following standards such as 3GPP 38.901. However, the shift from conventional methods like Least Square (LS) and Linear Minimum Mean Square Error (LMMSE) to AI-driven approaches in high-dimensional scenarios introduces its own set of research challenges [5].

Generative adversarial network (GAN)-based channel estimation has been proposed as an alternative solution. [6] employs a GAN-based approach as the primary method to estimate the channel for high degrees of freedom in terms of channel components. [7] uses a GAN-based method combined with pilot information to estimate the channel. [8] utilized a Wasserstein GAN with gradient penalty (WGAN-GP) as a component to estimate the channel based on the LMMSE method. However, a fundamental issue with the GAN model is the imbalance between the critic and the generator, as the critic converges too quickly, leading the generator to produce a limited variety of samples or experience vanishing gradients. This results in the generator failing to learn effectively from the critic's feedback, manifesting as inaccurate or incomplete reconstruction of channel state information [9].

In this paper, we propose a novel framework for adversarial training aimed at instantaneous channel estimation. Specifically, we focus on enhancing the generator's capability by leveraging additional probabilistic characteristics of the input features. To achieve this, we introduce an innovative loss function that integrates the third-order moment of the channel impulse response (CIR) with reconstruction loss, thereby capturing higher-order statistics of the channel characteristics. Additionally, we aim to strengthen the latent distribution learning capability of semi-supervised methods, such as Variational Autoencoder (VAE). Based on the proposed framework, we pose a *research question*: How can the AI model, particularly the critic in our framework, interpret the channel gain across the time, frequency, and space domains to effectively discriminate the generated channel?

The contributions of this work are twofold[1]. (i) We propose the use of the third-order moment of high DoF input in the generator's loss function and introduce a novel method for VAE's latent distribution learning, resulting in superior performance, as demonstrated by lower normalized mean square error (NMSE) in channel estimation. Notably, our estimation framework does not require pilot knowledge and, in some cases, achieves better performance than traditional methods. (ii) We introduce a new explainable AI method using activation mapping for the critic, termed AM4C. We observe a significant alignment between the critic's attention and the high-power gain regions in the time-frequency representation (TFR) of the CIR, providing insights into the design of the generator's loss, pilot patterns, and other loss functions for future research.

II. System Model

Notations. *Underlined notations* denote random variables, such as a single scalar $\underline{\alpha}$, a vector $\underline{\mathbf{x}}$, or a matrix $\underline{\mathbf{X}}$, while *non-underlined notations* represent realizations of these random variables, e.g., α, $\mathbf{x}$, and $\mathbf{X}$.

1) System Components and Network Topology: We consider a random network topology to analyze our proposed channel estimation framework comprehensively. Uplink (UL) estimation is performed by a single-antenna user terminal (UT) communicating with a base station (BS) equipped with K_{ant} antennas. The coordinates of the UT and BS in spherical

[1]The source code developed for this paper is available at tnd-lab.work

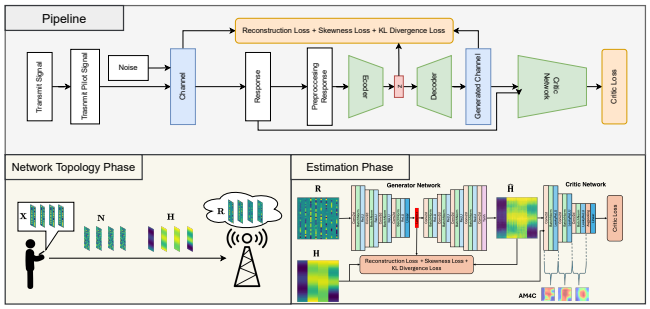


Fig. 1. Proposed WGAN-GP framework with explainable AI mechanism.

coordinates are denoted as $\mathbf{u}_{\mathrm{ut,bs}} = [d_{\mathrm{ut,bs}}, \varphi_{\mathrm{ut,bs}}, \phi_{\mathrm{ut,bs}}]$, where $d_{\mathrm{ut,bs}}$ is the radial distance, $\varphi_{\mathrm{ut,bs}}$ is the elevation angle of arrival along the z-axis, and $\phi_{\mathrm{ut,bs}}$ is the azimuth angle in the horizontal x-y plane. The UT antenna follows an omnidirectional pattern, so the antenna response vector for the UT is represented as $a_{\mathrm{ut}}(\varphi, \phi) = 1$. The BS is equipped with a uniform linear array (ULA) of K_{ant} cross-polarized antennas. The UL array response vector is: $\mathbf{a}_{\mathrm{bs}}(\varphi, \phi) = [a_1(\varphi, \phi)e^{\psi_1}, \ldots, a_{K_{\mathrm{ant}}}(\varphi, \phi)e^{\psi_{K_{\mathrm{ant}}}}]$, where $a_n(\varphi, \phi)$ represents the individual gain and phase response of the n-th antenna, λ_{cf} is the wavelength of the carrier frequency, and $\psi_n = -j\frac{2\pi}{\lambda_{cf}}(n-1)d_y \sin(\varphi)\sin(\phi)$ denotes the phase shift of the n-th antenna, d_y is the distance between two antennas.

2) Signal Modeling: Let K_{sym} be the number of OFDM symbols, i.e., time domain, K_{sc} be the umber of effective subcarriers, i.e., frequency domain. Recall that K_{ant} is the number of antennas, representing the space domain. A pilot mask, $\mathbf{M} \in \{0,1\}^{K_{\mathrm{sc}} \times K_{\mathrm{sym}}}$, is created using the orthogonal pilot pattern with Kronecker structure [4]. Let $\mathcal{T} \subseteq \{0, 1, \ldots, N_{\mathrm{sym}} - 1\}$ be the set of OFDM symbol indices reserved for pilot transmission, $\mathcal{K}_{i,j} \subseteq \{0, 1, \ldots, N_{\mathrm{sc}} - 1\}$ be the set of subcarrier indices allocated to transmitter i and stream j for pilot transmission. Considering *sparse* pilot pattern, the mask exhibits sparseness into distinct time and frequency components, i.e., $\mathbf{M} = \mathbf{P}_t \otimes \mathbf{P}_f^{i,j}$, where $\mathbf{P}_t(s)$ is a binary vector indicating pilot OFDM symbol indices, $\mathbf{P}_f^{i,j}(k)$ is a binary vector indicating the subcarrier indices allocated to transmitter i and stream j [4]. Here, $\mathbf{X}_p \in \mathbb{C}^{K_{\mathrm{sc}} \times K_{\mathrm{sym}}}$ can represented as $\mathbf{X}_p^{i,j}(k,s) = p_{i,j}$ if $\mathbf{M}^{i,j}(k,s) = 1$, and 0 otherwise, where $p_{i,j}$ are the QAM pilot symbols for transmitter i, stream j. $\mathbf{M}$ is designed as $\mathbf{M}^{i,j}(k,s) = 1$ if $s \in \mathcal{T}, k \equiv iK_{\mathrm{str}} + j \pmod{K_{\mathrm{seq}}}$, and 0 otherwise. Combining the mask and symbol assignment, the complete pilot pattern $\mathbf{X} \in \mathbb{C}^{K_{\mathrm{sc}} \times K_{\mathrm{sym}}}$ within the resource grid (RG) can be expressed as $\mathbf{X}^{i,j}(k,s) = \mathbf{M}^{i,j}(k,s) \times \mathbf{X}_p^{i,j}(k,s)$.

3) Channel Modeling: We consider communication between the UT and BS using the Urban Micro (UMi) channel model [4]. Let $h_{n,s,k}$ denote the channel for the n-th antenna and the k-th subcarrier at time step s. With the applied network topology, it can be represented as

$$h_{n,k,s} = \sum_{\ell=0}^{K_{\mathrm{path}}-1} h_\ell(s) e^{-j2\pi k \Delta_f \tau_\ell} a_n(\varphi_\ell, \phi_\ell) e^{\psi_{n,\ell}}, \quad (1)$$

where K_{path} denotes the number of multi-path components, and Δ_f represents the subcarrier spacing. Here, $h_\ell(s) \sim \mathcal{CN}(0,1)$ denotes the complex amplitude, τ_ℓ the delay, and $\psi_{n,\ell}$ the phase shift, all provided by the UMi model [4]. According to (1), the instantaneous TFR of the channel $\mathbf{H} \in \mathbb{C}^{K_{\mathrm{ant}} \times K_{\mathrm{sym}} \times K_{\mathrm{sc}}}$ represents a single realization of $\underline{\mathbf{H}}$.

4) Received Observation Modeling: Based on the channel model and signal modeling, assuming that the cyclic prefix (CP) is added to the transmit waveform and removed from the received waveform in the time domain, the TFR of the received signal, $\underline{\mathbf{R}}$, at the BS for the UT is expressed as

$$\underline{\mathbf{R}} = \underline{\mathbf{H}} \odot \underline{\mathbf{X}}_{\mathrm{exd}} + \underline{\mathbf{N}}, \quad (2)$$

where $\underline{\mathbf{R}} \in \mathbb{C}^{K_{\mathrm{ant}} \times K_{\mathrm{sym}} \times K_{\mathrm{sc}}}$ represents the received signal, and $\underline{\mathbf{N}} \in \mathbb{C}^{K_{\mathrm{ant}} \times K_{\mathrm{sym}} \times K_{\mathrm{sc}}}$ is an i.i.d. matrix of additive white Gaussian noise (AWGN) with entries $[\underline{\mathbf{N}}]_{ij} \sim \mathcal{CN}(0,1)$. The expanded version of $\underline{\mathbf{X}}$ is denoted by $\underline{\mathbf{X}}_{\mathrm{exd}} = \mathrm{repmat}(\underline{\mathbf{X}}, K_{\mathrm{ant}}, 1, 1)$, where repmat replicates $\underline{\mathbf{X}}$ along the antenna dimension.

III. Proposed Adversarial Training Framework

A. Adversarial Training for Channel Estimation

As aforementioned, we address the channel estimation problem using an adversarial training framework consisting of a generator G that generates synthetic channel estimates and a critic D that discriminates between the generated channel and the true channel. The generator G receives the observation $\underline{\mathbf{R}}$ as input, mapping it as $G(\underline{\mathbf{R}})$ such that $G(\underline{\mathbf{R}}) : \mathbb{R}^{K_{\mathrm{ant}} \times K_{\mathrm{sym}} \times K_{\mathrm{sc}}} \to \mathbb{R}^{K_{\mathrm{ant}} \times K_{\mathrm{sym}} \times K_{\mathrm{sc}}}$, where $\underline{\mathbf{R}} \mapsto \underline{\widehat{\mathbf{H}}}$. Thus, the instantaneous channel estimate is $\widehat{\mathbf{H}} = G(\mathbf{R})$. Given the true random instantaneous channel $\underline{\mathbf{H}}$, the input of the critic can be either $\underline{\mathbf{H}}$ or $\underline{\widehat{\mathbf{H}}}$. The critic can be represented as $D(\mathbf{H}^*) : \mathbb{R}^{K_{\mathrm{ant}} \times K_{\mathrm{sym}} \times K_{\mathrm{sc}}} \to [0,1]$ where $\underline{\mathbf{H}}^* \mapsto \{p(\mathbf{H}^*|H_0), p(\mathbf{H}^*|H_1)\}$. Here, $\underline{\mathbf{H}}^*$ can represent either $\underline{\mathbf{H}}$ or $\underline{\widehat{\mathbf{H}}}$ channel model. The likelihood probability $p(\mathbf{H}^*|H_0)$ and $p(\mathbf{H}^*|H_1)$ corresponding to the hypothesis H_0 for the generated channel and H_1 for the true channel.

The overall objective of the GAN framework is a *two-player minimax game*. The critic tries to maximize the value function $V(D,G)$ to distinguish between true and generated channels, while the generator tries to minimize this same value function to deceive the critic. Let $p_{\underline{\mathbf{H}}}(\mathbf{H})$, $p_{\underline{\widehat{\mathbf{H}}}}(\widehat{\mathbf{H}})$, $p_{\underline{\mathbf{R}}}(\mathbf{R})$ denote the distribution of the true channel, the generated channel, and the received observation, respectively. The minimax game of our channel estimation problem is expressed as

$$\begin{aligned} &\min_G \max_D V(D,G) = \\ &\mathbb{E}_{\mathbf{H}\sim p_{\underline{\mathbf{H}}}(\mathbf{H})}[\log D(\mathbf{H})] + \mathbb{E}_{\mathbf{R}\sim p_{\underline{\mathbf{R}}}(\mathbf{R})}[\log(1 - D(G(\mathbf{R})))] \\ &= \mathbb{E}_{\mathbf{H}\sim p_{\underline{\mathbf{H}}}(\mathbf{H})}[\log D(\mathbf{H})] + \mathbb{E}_{\widehat{\mathbf{H}}\sim p_{\underline{\widehat{\mathbf{H}}}}(\widehat{\mathbf{H}})}[\log(1 - D(\widehat{\mathbf{H}}))]. \end{aligned} \quad (3)$$

The technical challenge is the imbalanced optimization between the critic and generator networks during the early training phase. The critic D output has a more direct objective, discriminating tasks constrained to $[0,1]$, while the generator G learns to produce channel distributions over a wider and

more complex domain, i.e., $[\min \underline{\mathbf{H}}, \max \underline{\mathbf{H}}]$. In addition, $\underline{\mathbf{R}}$ and $\underline{\mathbf{H}}$ are high-dimensional data, which makes it challenging for G to learn the complex underlying distribution.

B. Problem Formulation for Optimizing the Generator

The training objective in our paper is to enable the generator to converge quickly from one distribution to another, i.e., from $p_{\underline{\widehat{\mathbf{H}}}}(\widehat{\mathbf{H}})$ to $p_{\underline{\mathbf{H}}}(\mathbf{H})$. To achieve this, the training process aims to minimize their Wasserstein distance, i.e., $W(p_{\underline{\mathbf{H}}}(\mathbf{H}), p_{\underline{\widehat{\mathbf{H}}}}(\widehat{\mathbf{H}}))$. Since the set of all joint distributions $\prod(p_{\underline{\mathbf{H}}}(\mathbf{H}), p_{\underline{\widehat{\mathbf{H}}}}(\widehat{\mathbf{H}}))$ is not available, Kantorovich-Rubinstein's dual formulation is used to simplify the Wasserstein distance [10].

$$\begin{aligned} &W(p_{\underline{\mathbf{H}}}(\mathbf{H}), p_{\underline{\widehat{\mathbf{H}}}}(\widehat{\mathbf{H}})) \\ &= \sup_{\|f\|_{\mathcal{L}} \leq 1} \left\{ \mathbb{E}_{\mathbf{H} \sim p_{\underline{\mathbf{H}}}(\mathbf{H})}[f(\mathbf{H})] - \mathbb{E}_{\widehat{\mathbf{H}} \sim p_{\underline{\widehat{\mathbf{H}}}}(\widehat{\mathbf{H}})}[f(\widehat{\mathbf{H}})] \right\}, \end{aligned} \tag{4}$$

where $f(\cdot)$ is a function parameterized by a neural network. Even if the probability density functions $p_{\underline{\widehat{\mathbf{H}}}}(\widehat{\mathbf{H}})$ and $p_{\underline{\mathbf{H}}}(\mathbf{H})$ do not perfectly align, the Wasserstein distance between them can remain small as long as their supports are close and the mass (probability density) is similarly distributed within those supports. In other words, if the generated channel estimates $\widehat{\mathbf{H}}$ have a distribution whose support largely overlaps with that of the true channel $\mathbf{H}$ and the mass is distributed in a similar fashion, then the Wasserstein distance will be small. Therefore, our goal is to ensure that the distribution $p_{\underline{\widehat{\mathbf{H}}}}(\widehat{\mathbf{H}})$ of the estimated channels closely matches the distribution $p_{\underline{\mathbf{H}}}(\mathbf{H})$ of the true channels, both in terms of support and mass distribution. To this end, first we let $\widehat{\mathbf{H}}$ and $\mathbf{H}$ be normalized to the same scale, i.e., the same range, such as $\mathbf{H}$ and $\widehat{\mathbf{H}}$ being in $[-1, 1]$. Considering pilot sparsity and adversarial training, the proposed channel estimation problem is formulated as

$$G^* = \arg \min_{(G|D)} W(p_{\underline{\mathbf{H}}}(\mathbf{H}), p_{\underline{\widehat{\mathbf{H}}}}(\hat{\mathbf{H}})) \tag{5a}$$

$$\text{subject to} \quad \mathfrak{Re}\{[\mathbf{H}]_{ij}\}, \mathfrak{Im}\{[\mathbf{H}]_{ij}\}, \mathfrak{Re}\{[\widehat{\mathbf{H}}]_{ij}\}, \mathfrak{Im}\{[\widehat{\mathbf{H}}]_{ij}\} \in [-1, 1], \forall i, \forall j, \tag{5b}$$

$$|p_{i,j}|^2 \leq 1, i \leq |\mathcal{T}|, j \leq K_{\text{sc}}, \tag{5c}$$

where $(G|D)$ indicates that we focus on optimizing G given a specific D. Constraint (5b) represents data normalization, while constraint (5c) enforces sparsity in the pilot pattern. We set $p_{i,j} = \pm\frac{1}{\sqrt{2}} \pm \mathrm{j}\frac{1}{\sqrt{2}}$, ensuring that $|p_{i,j}|^2 = 1$. Note that while the realization values are scaled to $[-1, 1]$, the distribution of the true channel $p_{\underline{\mathbf{H}}}(\mathbf{H})$ and the generated channel $p_{\underline{\widehat{\mathbf{H}}}}(\widehat{\mathbf{H}})$ differ. Achieving alignment between these distributions remains challenging.

IV. Third-Order Moment-Based VAE-WGAN-GP

A. Mathematical Description of the Proposed G and D

1) NN Architecture: Our goal is to upgrade the generator to a more refined version. Based on the work of [10], further investigated in [11], we define the generator function f_{g} with the input being the observation data $\underline{\mathbf{R}}$, and the set of parameters $\boldsymbol{\theta}$ is updated accordingly. Note that for a composite function, $f \circ g = f(g(x))$, and $\mathbf{H}^* = f_{\text{g}}(\mathbf{R}; \boldsymbol{\theta})$. Let $f_{\text{g}}^{[bl_i]}$ denote the i-th *feature extraction block*, which can be represented as $f_{\text{g}}^{[bl_i]} = f_{\text{g}}^{[a_i]} \circ f_{\text{g}}^{[b_i]} \circ f_{\text{g}}^{[c_i]}$, comprising activation, batch normalization, and convolution layers. Drawing from the VAE architecture [12], the G function can be divided into the encoder, latent space, and decoder parts. Let $\mathbf{a}_{\text{g,enc}} = f_{\text{g,enc}}(\mathbf{R}; \boldsymbol{\theta}_{\mathbf{a}_{\text{g,enc}}}) = f_{\text{g,enc}}^{[bl_1]} \circ f_{\text{g,enc}}^{[bl_2]} \circ f_{\text{g,enc}}^{[bl_3]} \circ f_{\text{g,enc}}^{[bl_4]} \circ f_{\text{g,enc}}^{[l]}$ represent the output vector of the encoder function. The latent vector $\mathbf{z}$ is the output of the transformation function $\mathbf{z} = f_{\text{g,trans}}(\mathbf{a}_{\text{g,enc}})$ and serves as the input to the decoder function. The decoder output is given by $\widehat{\mathbf{H}} = f_{\text{g,dec}}(\mathbf{z}; \boldsymbol{\theta}_{\text{g,dec}}) = f_{\text{g,dec}}^{[bl_1]} \circ f_{\text{g,dec}}^{[bl_2]} \circ f_{\text{g,dec}}^{[bl_3]} \circ f_{\text{g,dec}}^{[bl_4]} \circ f_{\text{g,dec}}^{[\tanh]}$. The function $f_{\text{g,dec}}^{[\tanh]}$ ensures that the output is normalized within $[-1, 1]$, aligning with the scaling of the true channel $\mathbf{H}$. Thus, G is mathematically described as

$$f_{\text{g}}(\mathbf{R}; \boldsymbol{\theta}) = f_{\text{g,dec}}(f_{\text{g,trans}}(f_{\text{g,enc}}(\mathbf{R}; \boldsymbol{\theta}_{\mathbf{a}_{\text{g,enc}}})); \boldsymbol{\theta}_{\text{g,dec}}). \tag{6}$$

In contrast, the critic network, denoted as f_{c}, has a set of parameters $\boldsymbol{\omega}$ and includes a feature extraction block defined as $f_{\text{c,ft}}^{[bl_i]} = f_{\text{c}}^{[a_i]} \circ f_{\text{c}}^{[b_i]} \circ f_{\text{c}}^{[c_i]}$ at the i-th block. Thus, the critic D can be mathematically expressed as

$$\begin{aligned} f_{\text{c}}(\mathbf{H}^*; \boldsymbol{\omega}) = f_{\text{c}}^{[l_4]}(f_{\text{c}}^{[avg_4]}(f_{\text{c,ft}}^{[bl_3]}(f_{\text{c,ft}}^{[bl_2]}(f_{\text{c,ft}}^{[bl_1]}(\mathbf{H}^*; \boldsymbol{\omega}_{\text{c,ft}}^{[bl_1]}); \\ \boldsymbol{\omega}_{\text{c,ft}}^{[bl_2]}); \boldsymbol{\omega}_{\text{c,ft}}^{[bl_3]}); \boldsymbol{\omega}_{\text{c}}^{[avg_4]}), \boldsymbol{\omega}_{\text{c}}^{[l_4]}). \end{aligned} \tag{7}$$

Before training the critic network, the function $f_{\text{c}}^{[\dim]}(\mathbf{H}) : \mathbb{R}^{K_{\text{ant}} \times K_{\text{sym}} \times K_{\text{sc}}} \mapsto \mathbb{R}^{(K_{\text{ant}} \times K_{\text{sym}}) \times K_{\text{sc}}}$ is defined to reduce the dimensionality of the channel input $\mathbf{H}$, making the model easier to train and reducing computational costs for high-dimensional channels. As the critic's target, the output of the linear function $f_{\text{c}}^{l_4}$ is scalar, with $f_{\text{c}}^{l_4}(X_{\underline{\mathbf{H}}}^{[l_4]}, \boldsymbol{\omega}_{\text{c}}^{[l_4]}) > f_{\text{c}}^{l_4}(X_{\underline{\widehat{\mathbf{H}}}}^{[l_4]}, \boldsymbol{\omega}_{\text{c}}^{[l_4]})$, where $X_{\underline{\mathbf{H}}}^{[l_4]}$ and $X_{\underline{\widehat{\mathbf{H}}}}^{[l_4]}$ are the inputs to the linear function for $D(\mathbf{H})$ and $D(\widehat{\mathbf{H}})$, respectively.

2) Proposed Latent Distribution Learning in VAE's Probabilistic Framework: $q(\mathbf{z}|\mathbf{H})$ represent the true posterior probability given the real channel $\mathbf{H}$. Since $\mathbf{H}$ cannot be observed directly, $q(\mathbf{z}|\mathbf{H})$ cannot be computed directly. We approximate it by sampling the approximate posterior probability of the encoder output vector, where $q(\mathbf{z}|\mathbf{H}) \approx p(\mathbf{z}|\mathbf{a}_{\text{g,enc}})$. Using the reparameterization method, the latent vector $\mathbf{z}$ is defined as $\mathbf{z} = f_{\text{g,trans}}(\mathbf{a}_{\text{g,enc}}) = \mu_{\mathbf{a}_{\text{g,enc}}} + \epsilon\sigma_{\mathbf{a}_{\text{g,enc}}}$, where $\mathbf{z} \sim p(\mathbf{z}|\mathbf{a}_{\text{g,enc}}) = \mathcal{N}(\mu_{\mathbf{z},\boldsymbol{\theta}}, \sigma_{\mathbf{z},\boldsymbol{\theta}})$. In most cases, the estimated latent vector $\mathbf{z} \sim \mathcal{N}(0, 1)$ is used for the inference phase. However, in our scenario, $\widehat{\mathbf{H}} \sim q(\widehat{\mathbf{H}}|z)$ cannot be generated randomly by the decoder using a Gaussian random vector as input. Moreover, $\mathbf{a}_{\text{g,enc}}$ is a compressed vector of the observation $\mathbf{R}$ rather than $\mathbf{H}$, making $q(\mathbf{z}|\mathbf{H}) \approx p(\mathbf{z}|\mathbf{a}_{\text{g,enc}})$ unreasonable, as $p_{\underline{\mathbf{R}}}(\mathbf{R})$ and $p_{\underline{\mathbf{H}}}(\mathbf{H})$ differ, as evidenced by the results. The proposed objective is to optimize the latent vector $\mathbf{z} \sim p(\mathbf{z}|\mathbf{a}_{\text{g,enc}})$ to closely approximate $p(\mathbf{z}|\mathbf{H})$, with $p_{\underline{\mathbf{H}}}(\mathbf{H}) \sim \mathcal{N}(\mu_{\underline{\mathbf{H}}}, \sigma_{\underline{\mathbf{H}}}^2)$. It is assumed that the number of training sets of real channel $|\mathcal{D}_{\text{train}}|$ is large enough to compute $\mu_{\underline{\mathbf{H}}}$, and $\sigma_{\underline{\mathbf{H}}}$. The latent vector $\mathbf{z} \sim \mathcal{N}(\mu_{\mathbf{z},\boldsymbol{\theta}}, \sigma_{\mathbf{z},\boldsymbol{\theta}}^2)$ is estimated directly from compressed vector $\mathbf{a}_{\text{g,enc}}$ and it is transformed close to $\mathcal{N}(\mu_{\underline{\mathbf{H}}}, \sigma_{\underline{\mathbf{H}}}^2)$. Applying trick parameterization, $\mathbf{z} = f_{\text{g,trans}}(\mathbf{a}_{\text{g,enc}}) = \mu_{\underline{\mathbf{H}}} + \mathbf{a}_{\text{g,enc}}\sigma_{\underline{\mathbf{H}}}$.

B. Proposed Loss Function

The target critic function D tries to maximize the distance of two distributions, while the generator model G is the opposite.

1) Loss of D: The loss function of critic network D is minimized by updating $\boldsymbol{\omega}$. Its formulation is represented as

$$L_D = \underbrace{\mathbb{E}_{\widehat{\mathbf{H}}\sim p_{\underline{\widehat{\mathbf{H}}}}(\widehat{\mathbf{H}})}[D_{\boldsymbol{\omega}}(\widehat{\mathbf{H}})]}_{\triangleq L_{\mathrm{f}}\text{: generative loss}} - \underbrace{\mathbb{E}_{\mathbf{H}\sim p_{\underline{\mathbf{H}}}(\mathbf{H})}[D_{\boldsymbol{\omega}}(\mathbf{H})]}_{\triangleq L_{\mathrm{r}}\text{: true loss}}. \quad (8)$$

Consider $p_{\underline{\tilde{\mathbf{H}}}}(\tilde{\mathbf{H}})$ as the mixture distribution of the $p_{\underline{\mathbf{H}}}(\mathbf{H})$ and $p_{\underline{\widehat{\mathbf{H}}}}(\widehat{\mathbf{H}})$, is given by: $p_{\underline{\tilde{\mathbf{H}}}}(\tilde{\mathbf{H}}) = \epsilon p_{\underline{\mathbf{H}}}(\mathbf{H}) + (1-\epsilon)p_{\underline{\widehat{\mathbf{H}}}}(\widehat{\mathbf{H}})$ where $\epsilon \sim \mathcal{U}(0,1)$ is a random scalar, drawn from a uniform distribution between 0 and 1. To constrain the critic network's update speed and preserve the 1-Lipschitz property. The gradient penalty loss can be defined as

$$L_{\mathrm{gp}} = \lambda_{\mathrm{gp}}\mathbb{E}_{\tilde{\mathbf{H}}\sim p_{\underline{\tilde{\mathbf{H}}}}(\tilde{\mathbf{H}})}[(\|\nabla D_{\boldsymbol{\omega}}(\tilde{\mathbf{H}})\|_2 - 1)^2]. \quad (9)$$

From (8) and (9), the adopted L_D is expressed as

$$\begin{aligned} L_D &= \text{generative loss} - \text{true loss} + \text{gradient penalty} \\ &= L_{\mathrm{f}} - L_{\mathrm{r}} + L_{\mathrm{gp}}. \end{aligned} \quad (10)$$

2) Proposed Loss of G: In the minimax game framework, the discriminator D function seeks to maximize the Lipschitz continuity of its output by controlling the gradients of $\widehat{\mathbf{H}}$ through its parameters $\boldsymbol{\omega}$. Concurrently, the generator G updates its parameters $\boldsymbol{\theta}$ to minimize the discrepancy between the generated data and real data. Initially, the loss of G can be designed based on the negative fake loss [10] as

$$L_G = -L_{\mathrm{f}} = -\mathbb{E}_{\mathbf{R}\sim p_{\underline{\mathbf{R}}}(\mathbf{R})}[D(G_{\boldsymbol{\theta}}(\mathbf{R}))]. \quad (11)$$

Taking into account the evidence lower bound of the VAE generator, we consider the KL divergence loss and the reconstruction loss, respectively, as

$$L_{\mathrm{KL}} = \lambda_{\mathrm{KL}}\frac{1}{2}\sum_{d=1}^{\mathbf{z}_{\mathrm{dim}}}\left(\log\frac{\sigma_{H_d}^2}{\sigma_{z_d,\boldsymbol{\theta}}^2} + \frac{\sigma_{z_d,\boldsymbol{\theta}}^2 + (\mu_{z_d,\boldsymbol{\theta}} - \mu_{H_d})^2}{\sigma_{H_d}^2} - 1\right), \quad (12)$$

$$L_{\mathrm{rec}} = \mathbb{E}_{\mathbf{H}\sim p_{\underline{\mathbf{H}}}(\mathbf{H}),\mathbf{R}\sim p_{\underline{\mathbf{R}}}(\mathbf{R})}[\|\mathbf{H} - G_{\boldsymbol{\theta}}(\mathbf{R})\|^2]. \quad (13)$$

Next, we aim to make the two distributions $p_{\underline{\mathbf{H}}}(\mathbf{H})$ and $p_{\underline{\tilde{\mathbf{H}}}}(\tilde{\mathbf{H}})$ as close as possible. Let γ_{r} and γ_{g} represent the *coefficients of skewness* or the third-order moments of the true and estimated channel distributions, respectively, which can be expressed as

$$\gamma_{\mathrm{r}} = \left(\mathbb{E}_{\mathbf{H}\sim p_{\underline{\mathbf{H}}}(\mathbf{H})}[(\mathbf{H} - \mu_{\mathrm{r}})^3]\right)/\sigma_{\mathrm{r}}^3, \quad (14)$$

$$\gamma_{\mathrm{g}} = \left(\mathbb{E}_{\widehat{\mathbf{H}}\sim p_{\underline{\widehat{\mathbf{H}}}}(\widehat{\mathbf{H}})}[(\widehat{\mathbf{H}} - \mu_{\mathrm{g}})^3]\right)/\sigma_{\mathrm{g}}^3, \quad (15)$$

where μ_{r}, μ_{g}, σ_{r}, and σ_{g} are the mean and the standard deviation of the true and generative distribution respectively. It is assumed that the distance of the skewness of true and generative distributions is close and continuous if the distance of two distributions is close. Thus, from (14) and (15), we can define our third order moment-based loss as

$$L_\gamma = |\gamma_{\mathrm{r}} - \gamma_{\mathrm{g}}|. \quad (16)$$

From (11)-(13), and (16), the proposed L_G is expressed as

$$\begin{aligned} L_G &= -\text{ fake loss} + \text{rec loss} + \text{kl loss} + \text{skewness loss} \\ &= -L_{\mathrm{f}} + L_{\mathrm{rec}} + L_{\mathrm{KL}} + L_\gamma. \end{aligned} \quad (17)$$

The proposed VAE-WGAN-GP-based channel estimation method, with enhanced G loss and latent distribution learning, is outlined in the pseudo-code of Algorithm 1.

Algorithm 1: Training Procedure of the Proposed VAE-WGAN-GP Generator and Critic Models

Input: Noisy received signal $\mathbf{R} \sim p_{\underline{\mathbf{R}}}(\mathbf{R})$. The true channel data $\mathbf{H} \sim p_{\underline{\mathbf{H}}}(\mathbf{H})$

Output: Trained Generator G for channel estimation $\widehat{\mathbf{H}} \sim p_{\underline{\widehat{\mathbf{H}}}}(\widehat{\mathbf{H}})$

Compute $\mu_{\underline{\mathbf{H}}}$, $\sigma_{\underline{\mathbf{H}}}$ from training set $\mathcal{D}_{\text{train}}$;

for $e = 0, \ldots, K_{epoch}$ **do**

  Sample $\{\mathbf{H}_i^{[e]}\}_{i=1}^b \sim p_{\underline{\mathbf{H}}}(\mathbf{H})$ a batch from true channel;

  Sample $\{\mathbf{R}_i^{[e]}\}_{i=1}^b \sim p_{\underline{\mathbf{R}}}(\mathbf{R})$ a batch from received signal;

  for $t = 0, \ldots, n_c$ **do**

    $\{\widehat{\mathbf{H}}_i^{[t],[e]}\}_{i=1}^b \leftarrow G(\{\mathbf{R}_i^{[e]}\}_{i=1}^b) \sim p_{\underline{\widehat{\mathbf{H}}}}(\widehat{\mathbf{H}})$ predict a batch from generated channel;

    $\{\tilde{\mathbf{H}}_i^{[t],[e]}\}_{i=1}^b \leftarrow \epsilon\{\mathbf{H}_i^{[e]}\}_{i=1}^b + (1-\epsilon)\{\widehat{\mathbf{H}}_i^{[t],[e]}\}_{i=1}^b$ a batch of mixture channel;

    Training $a_{\mathsf{c}}^{[t],[e]} \leftarrow D(\{\mathbf{H}^*{}_i^{[t],[e]}\}_{i=1}^b)$;

    Compute $f_{L_D}(a_{\mathsf{c}}^{[t],[e]})$;

    Compute $f_{L_{gp}}(\{\tilde{\mathbf{H}}_i^{[t],[e]}\}_{i=1}^b)$;

    $f_{L_D}^{[t],[e]} \leftarrow f_{L_D}(a_{\mathsf{c}}^{[t],[e]}) + f_{L_{gp}}(\{\tilde{\mathbf{H}}_i^{[t],[e]}\}_{i=1}^b)$;

    $\boldsymbol{\omega}^{[t+1],[e]} \leftarrow \mathrm{Opt}(f_{L_D}^{[t],[e]}, \boldsymbol{\omega}^{[t],[e]}, \eta, \beta_1, \beta_2)$;

  end

  $\mathbf{a}_{\mathrm{g,enc}}^{[e]} \leftarrow f_{\mathsf{g,enc}}(\{\mathbf{R}_i^{[e]}\}_{i=1}^b) \sim p(\mathbf{a}_{\mathrm{g,enc}})$;

  Reparameterize $\mathbf{z}^{[e]} \leftarrow \mu_{\underline{\mathbf{H}}} + \mathbf{a}_{\mathrm{g,enc}}^{[e]}\sigma_{\underline{\mathbf{H}}} \sim p(\mathbf{z}|\mathbf{a}_{\mathrm{g,enc}})$;

  $\{\widehat{\mathbf{H}}_i^{[e]}\}_{i=1}^b \leftarrow f_{\mathsf{g,dec}}(\mathbf{z}^{[e]})$;

  Compute $f_{L_f}(\{\widehat{\mathbf{H}}_i^{[e]}\}_{i=1}^b)$;

  Compute $f_{L_{rec}}(\{\widehat{\mathbf{H}}_i^{[e]}\}_{i=1}^b, \{\mathbf{H}_i^{[e]}\}_{i=1}^b)$;

  Compute $f_{L_{\mathrm{KL}}}(\mathbf{z}^{[e]}, \mu_{\underline{\mathbf{H}}}, \sigma_{\underline{\mathbf{H}}})$;

  $f_{L_G}^{[e]} \leftarrow f_{L_f}(\{\widehat{\mathbf{H}}_i^{[e]}\}_{i=1}^b) + f_{L_{rec}}(\{\widehat{\mathbf{H}}_i^{[e]}\}_{i=1}^b, \{\mathbf{H}_i^{[e]}\}_{i=1}^b) + f_{L_{\mathrm{KL}}}(\mathbf{z}^{[e]}, \mu_{\underline{\mathbf{H}}}, \sigma_{\underline{\mathbf{H}}})$;

  $\boldsymbol{\theta}^{[e+1]} \leftarrow \mathrm{Opt}(f_{L_G}^{[e]}, \boldsymbol{\theta}^{[e]}, \eta, \beta_1, \beta_2)$;

end

Initialization: Initialize parameters of G and D, set learning rates η, the batch size b, the number of iterations of the critic per generator iteration n_{c}, other Optimizer's parameters β_1 and β_2

Remark 1. *Numerical results demonstrate that all components of the Generator loss L_G, including the fake loss L_{f}, reconstruction loss L_{rec}, KL divergence loss L_{KL}, and the third-order moment-based skewness loss L_γ, successfully converge*

during the training process. This collective convergence of the individual loss components ensures the overall convergence of the proposed Generator loss L_G, as illustrated in Fig. 2.

C. Inference and Performance Analysis

To infer the model, our strategy involves simulating $|\mathcal{D}_{\text{test}}|$ samples from the Sionna framework, with each sample containing instantaneous channels with added noise as test sets. To adapt the model architecture, we reshape the actual channel based on $f_{\text{c}}^{[\text{dim}]}(\mathbf{H}^*)$. The evaluation metric is the Normalized Mean Square Error (NMSE), which measures the difference between the true and generated channels, defined as

$$\text{NMSE} = \frac{\sum_{i=1}^{|\mathcal{D}_{\text{test}}|}(\mathbf{H}_i - \widehat{\mathbf{H}}_i)^2}{\sum_{i=1}^{|\mathcal{D}_{\text{test}}|}(\mathbf{H}_i)^2}. \tag{18}$$

D. Explainable AI: Activation Mapping for the Critic (AM4C)

We propose an Explainable AI method to study the interpretability of the critic D during the training loop. Specifically, utilizing the activation mapping method, we propose the AM4C algorithm to identify and visualize the regions in the TRF of the channel that significantly influence the decision of D. The activation map of the k feature map at the spatial location (i, j) from the convolution function $f_{\text{c}}^{[c_n]}$, where $n \in \{1, 2, 3\}$ is expressed as

$$f_{\text{c}}^{[c_n]}(\mathbf{X}_{\text{c}}^{[c_n]}; \boldsymbol{\omega}_{\text{c}}^{[c_n]}) = \{\mathbf{A}_{k,i,j}^{c_n} \mid k \in \{1, \ldots, K\}\} \tag{19}$$

where K is the number of spatial locations, $\mathbf{X}_{\text{c}}^{[c_n]}$ is input of convolution layer. Next, we compute the gradient of the critic output of $f_{\text{c}}(\mathbf{H}^*; \boldsymbol{\omega}$ with respect to each feature map's activation $\frac{\partial f_{\text{c}}(\mathbf{H}^*; \boldsymbol{\omega}}{\partial f_{\text{c}}^{[c_n]}(\mathbf{X}_{\text{c}}^{[c_n]}; \boldsymbol{\omega}_{\text{c}}^{[c_n]})}$. Let a_{c}^k be the weight representing the overall influence of the critic decision value as follows

$$a_{\text{c}}^k = \frac{1}{K} \sum_{i,j} \frac{\partial f_{\text{c}}(\mathbf{H}^*; \boldsymbol{\omega})}{\partial f_{\text{c}}^{[c_n]}(\mathbf{X}_{\text{c}}^{[c_n]}; \boldsymbol{\omega}_{\text{c}}^{[c_n]})}. \tag{20}$$

Consider `ReLU` activation function to remove negative values. The activation map $M_D^{c_n}(i, j)$ can be defines as

$$M_D^{c_n}(i, j) = \texttt{ReLU}\Big((1/K) \sum_k \sum_{i,j} \frac{\partial f_{\text{c}}(\mathbf{H}^*; \boldsymbol{\omega})}{\partial f_{\text{c}}^{[c_n]}(\mathbf{X}_{\text{c}}^{[c_n]}; \boldsymbol{\omega}_{\text{c}}^{[c_n]})} \times f_{\text{c}}^{[c_n]}(\mathbf{X}_{\text{c}}^{[c_n]}; \boldsymbol{\omega}_{\text{c}}^{[c_n]})\Big). \tag{21}$$

The resulting activation map $M_D^{c_n}(i, j)$ provides a visual and quantitative representation of which spatial regions in the input channel data are most influential in the critic D. High values in the activation map indicate areas with substantial impact, thereby serving as the *attention regions* that the model focuses on during channel estimation.

Remark 2. *After training, the attention areas identified by the critic D consistently align with regions of high magnitude in the channel's TRF, as demonstrated in Fig. 6.*

V. Numerical Results and Visualization

Key simulation settings are presented in Table I; additional settings for simulation, training, and inference can be found in our published source code.

TABLE I
NETWORK, SIMULATION, AND HYPER-PARAMETERS SETTINGS

Parameter	Value	Parameter	Value
K_{ant}	4	λ_{gp}	10
K_{sym}	8	λ_{KL}	0.1
K_{sc}	32	$\mathcal{T}$	$\{2, 6\}$

1) Training Evaluation: To demonstrate the advantage of our proposed method, Fig. 2 shows that the generator loss and the citic loss tend to decrease shapely after K_{epoch}. It means that the proposed loss function convergence.

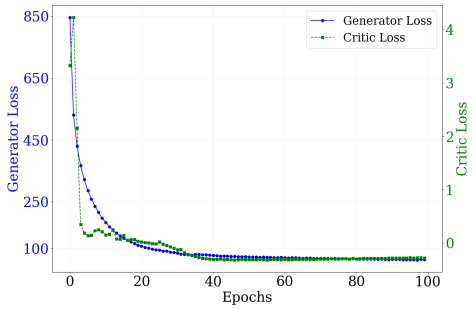


Fig. 2. Loss function of G and D

2) Inference: For testing, we use the same topology as the training set. To ensure fidelity, we also generate instantaneous channels with random noise as a test set and then evaluate the dataset. The comparison is shown in Fig. 3; the magnitude of the channel on each antenna is reconstructed similarly to the true channel. The results also indicate that the normalization process impacts the reconstruction of channel characteristics.

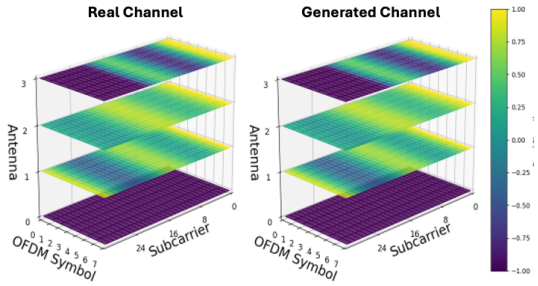


Fig. 3. True 3D channel realization and corresponding generative channel.

3) Performance Evaluation in terms of NMSE: We compare the NMSE of the proposed method with conventional methods using LS with interpolation techniques, i.e., nearest neighbor, linear, and LMMSE across different data domains. Fig. 4 shows that our methods obtain comparable outcomes with LMMSE methods in the lower SNRs. Besides it is a note that learning to generate samples without explicit distribution of WGAN-GP gives a poor result if compared with VAE methods. However, combining VAE and WGAN-GP can be a better solution, even with the proposed loss function which is the best result in our experiment.

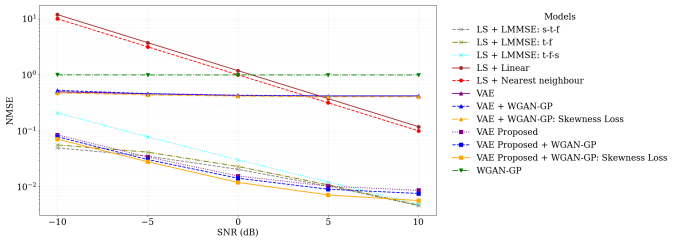


Fig. 4. NMSE of the proposed VAE-WGAN-GP and conventional methods.

It is true that the encoder input $\underline{\mathbf{R}}$ estimate from the function of Gaussian noise which is slightly shifted by the operation of $\underline{\mathbf{X}}$ containing mostly zero. It means that $\underline{\mathbf{R}} \sim \mathcal{N}(\mu_{\underline{\mathbf{R}}}, \sigma_{\underline{\mathbf{R}}})$, where $\mu_{\underline{\mathbf{R}}} \approx 0$, and $\sigma_{\underline{\mathbf{R}}} \approx 1$ as can be seen from the first image of Fig. 5. The distribution of the encoder output $\mathbf{a}_{\mathrm{g,enc}}$ is estimated as the same shape as the decoder input $\mathbf{z}$ and they are a right-skewed distribution. As a consequence, The distribution of the generated $\underline{\widehat{\mathbf{H}}}$ and the true $\underline{\mathbf{H}}$ is similar.

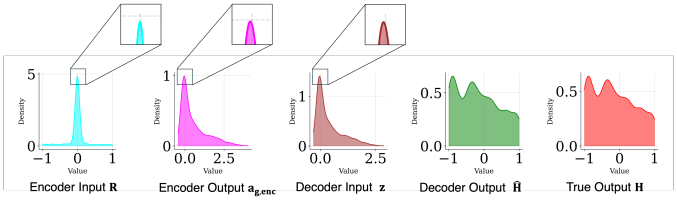
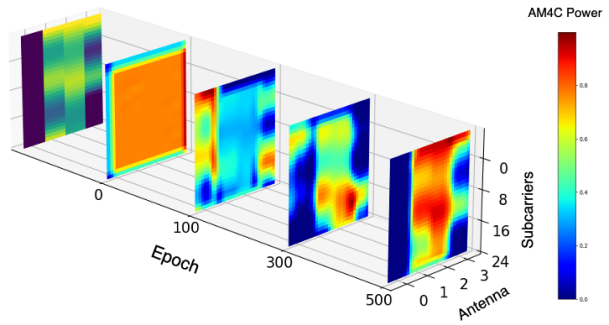


Fig. 5. The distribution of input and output of the model, $p_{\underline{\mathbf{R}}}(\mathbf{R})$, $p(\mathbf{a}_{\mathrm{g,enc}})$, approximated posterior $p(\mathbf{z}|\mathbf{a}_{\mathrm{g,enc}})$, $p_{\underline{\widehat{\mathbf{H}}}}(\widehat{\mathbf{H}})$, $p_{\underline{\mathbf{H}}}(\mathbf{H})$

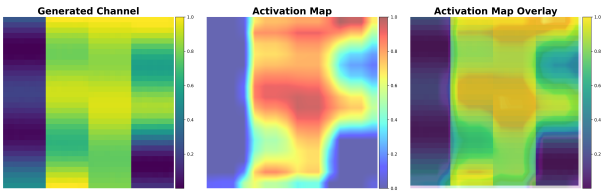
4) Explainable AI for Attention Area of the Channel Matrix: In Fig. 6a, we illustrate how the generator model updates features during training and how the critic model utilizes these features to determine whether they are generative. At epoch zero, it is observed that the critic focuses on the entire channel due to the initial noise distribution. By epoch 100, the critic tends to focus on lower magnitudes, as the generator easily produces these values and the critic seeks to avoid being misled by the generator. However, from epoch 300 to 500, the generator nearly surpasses the critic, prompting the critic to shift its focus to higher magnitudes, which contain more valuable information for classification. The activation map, in Fig. 6b, originates from the first convolutional layer of the critic, indicating that this layer is the closest to extracting information from the generator. This suggests that the critic model focuses on higher magnitudes of resource elements in the TFR to for decision-making.

VI. Conclusions

In this paper, we propose a new WGAN-GP model for channel estimation, in which the generator exploits the third-order moment of the data to improve its performance. We also propose AM4C as an explainable AI mechanism to obtain the attention area of the critic. We show that the proposed WGAN-GP performs superiorly, achieving lower NMSE than conventional methods, particularly in favorable training settings. Using the proposed AM4C, we demonstrate



(a)



(b)

Fig. 6. (a) The evolution of the attention area of the critic over training epochs. (b) Attention aera of the trained critic overlayed over generative channel.

that the critic relies on the high power gain regions of the time-frequency representation (TFR) of the channel to learn and discriminate the generated channel from the true channel.

References

[1] M. Hirzallah, M. Krunz, B. Kecicioglu, and B. Hamzeh, "5g new radio unlicensed: Challenges and evaluation," *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, no. 3, p. 689–701, Sep. 2021.

[2] X. Ma, Z. Gao, F. Gao, and M. Di Renzo, "Model-driven deep learning based channel estimation and feedback for millimeter-wave massive hybrid mimo systems," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 8, p. 2388–2406, Aug. 2021.

[3] H. Ye, G. Y. Li, and B.-H. Juang, "Power of deep learning for channel estimation and signal detection in ofdm systems," *IEEE Wireless Communications Letters*, vol. 7, no. 1, p. 114–117, Feb. 2018.

[4] Hoydis et al., "Sionna: An open-source library for next-generation physical layer research," Mar. 2022. [Online]. Available: https://nvlabs.github.io/sionna

[5] Jiao et al., "Advanced deep learning models for 6g: Overview, opportunities, and challenges," *IEEE Access*, vol. 12, p. 133245–133314, 2024.

[6] Haider et al., "Gan-based channel estimation for irs-aided communication systems," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 4, p. 6012–6017, Apr. 2024.

[7] Y. Du, Y. Li, M. Xu, J. Jiang, and W. Wang, "A joint channel estimation and compression method based on GAN in 6G communication systems," *Applied Sciences*, vol. 13, no. 4, p. 2319, Feb. 2023.

[8] E. Balevi and J. G. Andrews, "Wideband channel estimation with a generative adversarial network," *IEEE Transactions on Wireless Communications*, vol. 20, no. 5, p. 3049–3060, May 2021.

[9] D. Bau, J.-Y. Zhu, J. Wulff, W. Peebles, B. Zhou, H. Strobelt, and A. Torralba, "Seeing what a gan cannot generate," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, Oct. 2019.

[10] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of wasserstein gans," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, p. 5769–5779.

[11] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," in *Proceedings of the International Conference on Learning Representations*, Apr. 2017.

[12] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *CoRR*, vol. abs/1312.6114, 2013. [Online]. Available: https://api.semanticscholar.org/CorpusID:216078090